

Deep Weakly Supervised Learning Networks for Object Localization

by

Shakeeb MURTAZA

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE
TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, JUNE 10, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Shakeeb Murtaza, 2025



This Creative Commons license allows readers to download this work and share it with others as long as the author is credited. The content of this work cannot be modified in any way or used commercially.

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Mr. Eric Granger, Thesis supervisor
Department of systems engineering, École de technologie supérieure

Mr. Marco Pedersoli, Thesis Co-Supervisor
Department of systems engineering, École de technologie supérieure

Mr. Christian Desrosiers, Chair, Board of Examiners
Department of software engineering and IT, École de technologie supérieure

Mr. Alessandro Lameiras Koerich, Member of the Jury
Department of software engineering and IT, École de technologie supérieure

Mr. Nicolas Thome, External Independent Examiner
Sorbonne Université, CNRS, ISIR, Paris, France

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON MAY 27, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my research advisors, Professor Eric Granger and Professor Marco Pedersoli, for their invaluable support, guidance, and encouragement during the doctoral studies. Their consistent mentorship, and unwavering dedication have been pivotal to the successful completion of this research. I am deeply thankful for the substantial time they invested in numerous discussions, which significantly shaped the direction of this thesis and facilitated the publication of several research articles. Furthermore, their financial support has been indispensable in bringing this research to fruition. I would also like to sincerely acknowledge my post-doctoral fellow, Soufiane Belharbi, for his collaboration throughout this research. His insightful discussions and technical support have been an essential part of this journey.

I would also like to extend my sincere gratitude to my master's supervisor, Professor Tehseen Zia, for nurturing my passion for research, and to my mentor, Mirza Ahsan Ullah, whose introduction to the field of AI played a crucial role in the development of my research skills.

I am immensely grateful to my parents and my sister for their unwavering sacrifice, love, and steadfast support throughout the years. Their encouragement has been a constant source of strength. I am especially indebted to my wife, whose support, understanding, and personal sacrifices enabled me to dedicate countless hours to this research, often late into the night. Her patience and unwavering belief in my work have been truly invaluable.

Also, I would like to express my thanks to my friends in Montreal, especially Muhammad KaleemUllah Khan and Alexis Guichemerre, who provided critical support during the more challenging moments of this journey.

Réseaux d'apprentissage profond faiblement supervisé pour la localisation d'objets

Shakeeb MURTAZA

RÉSUMÉ

La reconnaissance visuelle, qui englobe des tâches telles que la classification, la localisation et la segmentation d'objets, est cruciale pour différentes applications de vision par ordinateur allant de la conduite autonome à l'analyse d'images médicales. Les récentes avancées dans le domaine des réseaux neuronaux profonds ont permis d'améliorer considérablement ces tâches. Cependant, le succès de ces modèles est fortement attribué à la grande quantité de données annotées, dont l'acquisition est coûteuse et prend du temps. En outre, ce paradigme n'est pas extensible à différents domaines, car il n'est pas pratique de collecter des millions d'images étiquetées pour chaque domaine dans lequel nous voulons déployer notre modèle d'apprentissage profond. Pour éviter de dépendre d'annotations étendues, la communauté des chercheurs a exploré différents paradigmes d'apprentissage, y compris les méthodes d'apprentissage semi-supervisé, auto-supervisé et faiblement supervisé. Pour contribuer à cet effort, cette thèse se concentre sur l'apprentissage faiblement supervisé pour la localisation d'objets. Tout d'abord, nous avons identifié les défis associés aux méthodes existantes, suivis par la proposition de nouvelles directions visant à améliorer la performance de la localisation d'objets.

DiPS est la première contribution à cette thèse ; nous proposons d'exploiter les cartes agnostiques en termes de classes des transformateurs auto-supervisés pour la localisation d'objets faiblement supervisée, où seules les étiquettes de classes d'images sont disponibles. En utilisant différentes cartes d'attention provenant de transformateurs auto-supervisés, nous sélectionnons d'abord une carte qui contient des régions discriminantes correspondant à l'objet d'intérêt pour générer des propositions d'objets. Ces propositions sont ensuite utilisées pour construire des pseudo-étiquettes en sélectionnant quelques pixels d'arrière-plan et de premier plan qui correspondent au modèle, lequel est capable d'effectuer à la fois des tâches de classification et de localisation. Cet échantillonnage permet à notre modèle d'atténuer les suractivations et le bruit de fond. Des expériences approfondies sur les ensembles de données difficiles TelDrone, CUB, OpenImages et ILSVRC montrent que la méthode proposée permet d'obtenir de meilleures performances de localisation que les méthodes de pointe.

En deuxième lieu, nous introduisons la distillation de texte pour la localisation (TeD-Loc), une stratégie d'apprentissage pour distiller des informations de localisation à partir d'enchâssements de texte CLIP dans la colonne vertébrale visuelle de notre modèle afin d'améliorer les performances de localisation. Les méthodes WSOL conventionnelles extraient généralement des cartes de localisation à partir de classificateurs pré-entraînés, qui ont tendance à se concentrer sur les parties d'objet les plus discriminantes et ne parviennent pas à mettre en évidence les différents objets. Les méthodes WSOL récentes qui s'appuient sur des modèles vision-langage tels que CLIP dépendent également des étiquettes de classe de vérité au sol ou des prédictions de classe des classificateurs externes pour générer des cartes de localisation, ce qui limite leur déploiement dans différentes applications. Pour remédier à ces limitations, TeD-Loc distille

des informations à partir de l'intégration de textes CLIP dans l'épine dorsale de notre modèle de vision. En outre, en formulant le problème dans un cadre d'apprentissage multi-instances, TeD-Loc permet au modèle de converger pour les tâches de localisation et de classification d'objets. Les résultats expérimentaux montrent que TeD-Loc atteint des performances de pointe.

La troisième contribution de cette thèse est un protocole d'évaluation réaliste pour WSOL. Comme les modèles WSOL sont formés à partir d'étiquettes au niveau de la classe, leur évaluation, l'ajustement des hyperparamètres et la sélection reposent toujours sur un ensemble de validation qui comprend des annotations de boîtes de délimitation. L'utilisation d'annotations de boîtes de délimitation pour l'évaluation n'est pas conforme au paradigme WSOL, ce qui conduit à une surestimation des performances de localisation. Pour une évaluation réaliste, nous proposons d'utiliser des pseudo-boîtes pour un ensemble de validation qui peut être obtenu à partir de la méthode de proposition de région prête à l'emploi. Des expériences approfondies sur des ensembles de données d'images naturelles et médicales montrent que les performances du modèle évalué à l'aide de pseudo-boîtes sont comparables à celles obtenues à l'aide de boîtes GT annotées manuellement.

Mots-clés: Localisation d'objets, Localisation d'objets faiblement supervisée, Transformateurs de vision, Apprentissage auto-supervisé, Apprentissage profond

Deep Weakly Supervised Learning Networks for Object Localization

Shakeeb MURTAZA

ABSTRACT

Visual recognition, encompassing tasks such as object classification, localization, and segmentation, is crucial for different computer vision applications ranging from autonomous driving to medical image analysis. Recent advances in deep neural networks have yielded significant improvements in these tasks. However, the success of these models is heavily attributed to the large amount of annotated data, which are costly and time-consuming to acquire. Additionally, this paradigm is not scalable across different domains, as it is impractical to collect millions of labeled images for every domain in which we want to deploy our deep learning model. To avoid reliance on extensive annotations, the research community has explored different learning paradigms, including semi-supervised, self-supervised, and weakly supervised learning methods. To contribute to this effort, this thesis focuses on weakly supervised learning for object localization. First, we identified the challenges associated with existing methods, followed by the proposal of new directions aimed at improving object localization performance.

DiPS is the first contribution to this thesis; we propose to leverage class-agnostic maps from self-supervised transformers for weakly supervised object localization, where only image-class labels are available. Using different attention maps from self-supervised transformers, we first select a map that contains discriminative regions corresponding to the object of interest for generating object proposals. These proposals are then used to build pseudo-labels by selecting a few background and foreground pixels that fit the model, which is capable of performing both classification and localization tasks. This sampling enables our model to mitigate over-activations and background noise. Extensive experiments on the challenging TelDrone, CUB, OpenImages and ILSVRC datasets show that the proposed method can achieve better localization performance compared to the state-of-the-art methods.

As a second contribution, we introduce text distillation for localization (TeD-Loc), a learning strategy for distilling localization information from CLIP text embeddings into the visual backbone of our model to improve localization performance. Conventional WSOL methods typically extract localization maps from pre-trained classifiers, which tend to focus on the most discriminative object parts and fail to highlight different objects. Recent WSOL methods that leverage vision-language models like CLIP also depend on ground truth class labels or class predictions from external classifiers to generate localization maps, thereby limiting their deployment in different applications. To address these limitations, TeD-Loc distills information from the CLIP text embeddings into the backbone of our vision model. Furthermore, by formulating the problem within a multi-instance learning framework, TeD-Loc enable the model to converge for both object localization and classification tasks. Experimental results show that TeD-Loc achieves state-of-the-art performance.

The third contribution of this thesis is a realistic evaluation protocol for WSOL. As WSOL models are trained using class-level labels, their evaluation, hyperparameter tuning, and selection still rely on a validation set that includes bounding-box annotations. Employing bounding-box annotation for evaluation is not aligned with the WSOL paradigm, leading to overestimating localization performance. For realistic evaluation, we propose to employ pseudo-boxes for a validation set that can be obtained from the off-the-shelf region proposal method. Extensive experiments on both natural and medical image datasets show that the performance of the model evaluated using pseudo-boxes is comparable to the performance obtained using manually annotated GT boxes.

Keywords: Object Localization, Weakly Supervised Object Localization, Vision Transformers, Self-Supervised Learning, Deep Learning

TABLE OF CONTENTS

	Page
INTRODUCTION	1
0.1 Challenges in WSOL	2
0.2 Motivation	4
0.3 Research Objectives and Contributions	6
0.4 Thesis Organization	7
 CHAPTER 1 BACKGROUND AND LITERATURE REVIEW	 11
1.1 Machine Learning	11
1.1.1 Traditional ML	12
1.1.2 Deep Learning	13
1.1.2.1 Related DL models	14
1.2 Weakly Supervised Learning	16
1.3 Visual Recognition	18
1.4 A Taxonomy of Methods for Weakly Supervised Object Localization	19
1.4.1 Problem Formulation	20
1.4.2 Related Frameworks	21
1.4.3 Few-shot Methods (Contrastive-pretrained).	22
1.4.4 Bottom-up Techniques	23
1.4.5 Top-down Techniques	34
1.4.6 Critical Analysis	38
 CHAPTER 2 DIPS: DISCRIMINATIVE PSEUDO-LABEL SAMPLING WITH SELF-SUPERVISED TRANSFORMERS FOR WEAKLY SU- PERVISED OBJECT LOCALIZATION	 41
2.1 Introduction	42
2.2 Related Work	47
2.3 The Proposed Discriminative Pseudo-label Sampling	51
2.3.1 Notation	51
2.3.2 Discriminative Sampling of Proposals	52
2.3.3 Pixel-wise Pseudo-labels	55
2.3.4 Training DiPS Architecture	56
2.4 Results and Discussion	58
2.4.1 Experimental Methodology	58
2.4.2 Comparison with State-of-Art Methods	60
2.4.3 Error Analysis and Ablation Study	65
2.5 Conclusion	66
 CHAPTER 3 TED-LOC: TEXT DISTILLATION FOR WEAKLY SUPERVISED OBJECT LOCALIZATION	 71
3.1 Introduction	72

3.2	Related Work	75
3.3	The Proposed Method	78
3.3.1	Key Components	79
3.3.2	The TeD-Loc Training Method	80
3.4	Results and Discussion	83
3.4.1	Experimental Methodology	83
3.4.2	Comparison with State-of-the-Art	85
3.4.3	Complexity Analysis	88
3.5	Conclusion	89
3.6	Supplementary material	90
3.6.1	Sampling FG/BG Regions for Pseudo-label Generation	90
3.6.2	Extended Results	91
3.6.2.1	Patch-level Localization with Text Anchors	91
3.6.2.2	Impact of Orthogonalization on Performance	92
CHAPTER 4	A REALISTIC PROTOCOL FOR EVALUATION OF WEAKLY SUPERVISED OBJECT LOCALIZATION	95
4.1	Introduction	96
4.2	Related Work	100
4.3	A Realistic Protocol for Model Evaluation	103
4.3.1	Model Selection	103
4.3.2	Threshold Estimation	108
4.4	Experimental Validation	108
4.4.1	Methodology	108
4.4.2	Results and Discussions	110
4.5	Conclusion	114
4.6	Supplementary Material	115
4.6.1	Review of Evaluated Methods and their Search Spaces	116
4.6.1.1	Evaluated Methods	116
4.6.1.2	Hyperparamter Search Space	118
4.6.2	Generation of Noisy Ground Truth Bboxes	118
4.6.3	Pseudo-bboxes Performance Across Different Selection Steps	119
4.6.4	Localization Evaluation in WSOL (Thresholded-IoU vs IoU)	120
4.6.5	Localization Evaluation with different thresholds	122
4.6.6	Experiments with medical datasets	122
CONCLUSION AND RECOMMENDATIONS		127
5.1	Summary of Contributions	127
5.2	Limitations and Recommendations	128
APPENDIX I	DISCRIMINATIVE SAMPLING OF PROPOSALS IN SELF-SUPERVISED TRANSFORMERS FOR WEAKLY SUPERVISED OBJECT LOCALIZATION	131

APPENDIX II	LEVERAGING TRANSFORMERS FOR WEAKLY SUPER- VISED OBJECT LOCALIZATION IN UNCONSTRAINED VIDEOS	157
APPENDIX III	PUBLICATIONS DURING PH.D. STUDY	177
BIBLIOGRAPHY		178

LIST OF TABLES

	Page
Table 2.1	MaxBoxAccV2 and PxAP performance of proposed and state-of-the-art methods on the CUB and OpenImages datasets 60
Table 2.2	MaxBoxAccV2 performance of proposed and state-of-art methods on the ILSVRC dataset, along with IoU at different threshold values, denoted by δ . We adopted some of the results for baseline methods from (Kim, Kim, Lee & et. al, 2022) 61
Table 2.3	Quantitative comparison of DiPS and existing state-of-the-art methods on the TelDrone Dataset using MaxBoxAccV2 metric. Experimental results on this proprietary dataset indicate robust performance across various datasets 61
Table 2.4	A comparative analysis of our approach against self-supervised and prompt-based models; In self-supervised transformers, class tokens are utilized to compute localization performance. For this purpose, we harvest class tokens from the last attention block of the transformer. These class tokens are then binarized and used to perturb the original image by blurring the background regions. Subsequently, this perturb is processed by a classifier, and a map with the maximum classifier score is selected for computation localization scores. This selection strategy closely adheres to the pseudo-label generation process employed in our method. For CLIP-ES, class labels are obtained from pre-trained models that are then passed through the CLIP model along with the input image to obtain attention map Lin , Chen, Wang & et. al (2023). Moreover, our method is able to surpass self-supervised vision transformers and prompt-based models 64
Table 2.5	Error analysis of our method. Results of baseline methods (VGG16, InceptionV3, TS-CAM) are borrowed from (Gao <i>et al.</i> , 2021) 66
Table 2.6	Ablation study of our model for loss functions of localization head 66
Table 3.1	Performance comparison of TeD-Loc with state-of-the-art methods on the ILSVRC and CUB datasets. The metrics reported are MaxBoxAcc, top-1 loc., and top-5 loc.. The first two rows correspond to Grad-CAM for CLIP (?), where class labels are required to produce localization maps, making direct comparison unfair as it leverages privileged information. TeD-Loc outperforms

	existing methods without relying on ground-truth class labels or thresholds derived from ground-truth data.	85
Table 3.2	Ablation study on the CUB dataset showing the impact of different loss combinations on MaxBoxAcc performance.	88
Table 3.3	Computational complexity and localization performance of our proposed TeD-Loc against GenPrompt.	89
Table 3.4	Localization performance via (patch, class) embeddings dot product: $\langle z_p, t_y \rangle$ where $z_p, \forall p \in \Omega$ is the patch embeddings, and y is the true image class. We report localization performance (MaxBoxAcc, Top-1 loc., Top-5 loc.) using different variants of CLIP, and our method.	92
Table 3.5	Impact of class text embeddings (text anchors) orthogonalization over localization and classification performance in our method over CUB dataset.	92
Table 4.1	IoU performance of generated pseudo-bboxes w.r.t GT bboxes on the validation set	112
Table 4.2	Test-set MaxBoxAcc of WSOL models with different selection criteria on CUB and ILSVRC. The <i>select</i> column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the <i>cls.</i> performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT	113
Table 4.3	Analysis of MaxBoxAcc (IoU-50) versus IoU using oracle bboxes for model selection. Analysis using pseudo-bboxes is presented in <i>supplementary material</i> exhibiting same behaviour	114
Table 4.4	Hyperparameter search space for different methods	118
Table 4.5	Performance of pseudo-bboxes obtained using different off-the-shelf region proposal methods across multiple refinement stages over the validation set. The table highlights the incremental improvement in IoU through various selection steps	119

Table 4.6	Comparative Analysis of MaxBoxAcc (IoU-50) versus IoU on CUB and ILSVRC with different model selection methods120
Table 4.7	Test-set MaxBoxAccV2 that is average of three threshold IoU-30, IoU-50, IoU-70 (here MaxBoxAccV2 is average at three threshold) along with the it of WSOL models with different selection criteria on CUB. The <i>select</i> column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the <i>cls.</i> performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT123
Table 4.8	Test-set MaxBoxAccV2 that is average of three threshold IoU-30, IoU-50, IoU-70 (here MaxBoxAccV2 is average at three threshold) along with the it of WSOL models with different selection criteria on ILSVRC. The <i>select</i> column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the <i>cls.</i> performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT124
Table 4.9	Impact of varying levels of erosion in GT masks on LOC performance (PxAP) compared to GT-mask and pseudo-masks. It shows that despite substantial increases in mask error (1-AUC) across different noise levels, the LOC performance remains stable, particularly when using pseudo-masks, highlighting the robustness and generalizability of our evaluation protocol across different domains126

LIST OF FIGURES

	Page
Figure 0.1 Classification vs object localization Images obtained from Welinder, Branson, Mita & et. al (2010)	1
Figure 0.2 (a) Learning spectrum illustrating the WSL. The red area represents WSL, while the green area represents a conventional full-supervised learning setting. Moreover, the axis here represents human annotation. Specifically, the task of predicting bboxes or pixel-level labels from image-level labels is referred to as WSOL. (b) Zoom-in learning spectrum from the perspective of object localization	2
Figure 0.3 Thesis Organization. The arrows show the dependencies among the chapters and appendices	9
Figure 1.1 Illustration of self-distillation with <i>no</i> labels Adapted from Caron <i>et al.</i> (2021)	16
Figure 1.2 Types of weakly supervised settings Adapted from Zhou (2018)	17
Figure 1.3 Taxonomy of deep WSOL methodologies trained using image-level class labels. Methods are categorized and arranged chronologically within each group: (1) Bottom-up approaches, which extract activations for producing localization maps during forward-pass information exclusively; (2) Top-down approaches, which requires information from both forward and backward passes; and (3) Zero-shot approaches, based on CLIP model and was proposed within the Weakly Supervised Semantic Segmentation (WSSS) paradigm. Icons in front of method names denote modality support: ► indicates WSOL methods supporting video data, while ► denotes support for both image and video modalities in the WSOL context. the missing icon signifies that the method is restricted to single-image input	19
Figure 1.4 Illustration of CAM technique Adapted from Zhou, Khosla, Lapedriza & et. al (2016)	24
Figure 1.5 Illustration of multi-dilated network for improving CAM Adapted from Wei, Xiao, Shi & et. al (2018)	29
Figure 1.6 Illustration of TS-CAM: token semantic coupled attention map for WSOL Adapted from Gao <i>et al.</i> (2021)	31

Figure 2.2	<i>Our approach for generating discriminative proposals from class tokens of SSTs.</i> First, the localization map are automatically binarized (Otsu, 1979) to identify regions of interest that may contain a specific object. From each binarized map, all connected areas are extracted along with their tight bounding boxes. This creates a pool of bounding boxes from different attention maps. Each bounding box is scored using an external classifier f_x. We use the posterior probability of the true class y of the image to measure the likelihood of the bounding box containing the true object. Outside of the box is suppressed by blurring operations. The top-K boxes are selected. At each training step, we randomly select a proposal box. Inside the box, we randomly select foreground pixels. While background pixels are selected from outside the box. A detailed flow diagram is displayed in Fig.2.3	45
Figure 2.3	<i>DiPS: Our proposed method for training a transformer network for WSOL tasks using a combination of localization and classification networks.</i> The full model is composed of a vision transformer encoder, a classification head, and a localization head. Training: An image class label is required to train the classification head, and our generated pixel-wise pseudo labels are needed to train the localization head. Fig.2.2 illustrates how this pseudo-supervision is computed. Inference: In a single forward pass through our model over the input image, the classification head yields the predicted class probabilities, while the localization head localizes the object in the image	52
Figure 2.4	Samples from the test set on the CUB-200-2011 dataset generated by our proposed and state-of-the-art methods. In different examples, the results of baseline methods highlight different parts of the object, but the entire object is encompassed within the bounding box due to an extensive search of thresholds. In contrast, the map generated by our method covers the full object with relatively consistent activation values across the object. Additionally, our model effectively mitigates noise in the generated map, which is present in the class tokens used to produce pseudo-labels. Here, green denotes the ground truth bounding box, while red corresponds to the predicted bounding box	63
Figure 2.5	The MaxBoxAcc performance of DiPS and state-of-the-art methods is at different threshold values ($\delta = \{30, 40, 70\}$) calculated on a test set from the CUB-200-2011 dataset	65

Figure 2.6	Visual results for proposed and state-of-the-art methods on the ILSVRC dataset. Here, green denotes the ground truth bounding box, while red corresponds to the predicted bounding box	67
Figure 2.7	Visualization of results on the OpenImages dataset	68
Figure 2.8	Visualization of results on the TelDrone dataset. We added a few examples for this dataset, given the proprietary restrictions specified by Ericsson Incorporation. Here, green denotes the ground truth bounding box, while red corresponds to the predicted bounding box	69
Figure 2.9	Visual results of our methods compared to the self-supervised method. Here, green denotes the ground truth bounding box, while red corresponds to the predicted bounding box	69
Figure 3.1	Comparison of our TeD-Loc versus CLIP-ES Lin <i>et al.</i> (2023) methods for extracting localization maps from CLIP. (A) CLIP-ES utilizes Grad-CAM to extract localization maps from CLIP, requiring GT class labels during inference. (B) In contrast, our TeD-Loc model distills knowledge from CLIP text embeddings into the visual encoder during training, allowing it to produce both classification scores and localization maps without requiring class labels during inference.	72
Figure 3.2	Overview of the TeD-Loc method for distilling FG text embeddings into the patch embedding backbone. First, pseudo-labels are extracted to guide the identification of FG and BG patches. By leveraging these FG/BG regions, the model minimizes the similarity of E_V with the relevant text embedding for FG classes, while maximizing dissimilarity with embeddings of other classes. Through a binary FG/BG classifier, TeD-Loc generates localization maps by classifying patches as FG or BG, while generating class probabilities for image classification. This joint task enables the model to produce both accurate localization and classification outputs without explicit bounding box supervision.	77
Figure 3.3	t-SNE visualizations of CLIP text embeddings for ILSVRC Deng, Dong, Socher & et. al (2009) classes before and after orthogonalization. (Left) Prior to orthogonalization, embeddings of semantically similar classes (e.g., “airplane” and “aircraft”) cluster closely together, leading to potential confusion. (Right) After orthogonalization (QR decomposition), the embeddings are more uniformly distributed and orthogonal, reducing overlap.	81

- Figure 3.4 Qualitative comparison on the ILSVRC dataset. GenPrompt fails to localize objects in complex scenes, often due to its dependency on external classifiers for computing text embeddings, which can fail if the classifier makes mistakes. This dependence on class labels during inference highlights GenPrompt’s vulnerability to localization errors. In contrast, TeD-Loc can localize objects within complex scenes. Here, **green bboxes** denote GT localization, while **red bboxes** represent predicted localizations. 87
- Figure 3.5 Visualization of localization map defined via (patch, class) embeddings dot product: $\langle z_p, t_y \rangle$ where $z_p, \forall p \in \Omega$ is the patch embeddings, and y is the true image class over different variants of CLIP, and our method. TE@PE is the vanilla CLIP Radford, Kim, Hallacy & et. al (2021) where TE is the text embedding, and PE is the patch embedding. 93
- Figure 4.1 (A) Example of LOC (**MaxBoxAcc**) and **cls.** accuracy w.r.t the number of epochs on the CUB validation set in a WSOL setting, where only class-labels are used to train the model. These curves show that LOC and **cls.** tasks are loosely correlated where convergence is reached at different training epochs. Typically, high LOC performance is achieved early in the training, followed by degradation. However, the classifier takes a longer time to converge. Selecting the best model for LOC may need a LOC annotation in the validation set. (B) Significant bias arises when using GT bounding boxes and test set thresholds, leading to overestimated LOC accuracy, as bbox annotations are typically absent in real-world scenarios. *In a realistic WSOL scenario (CL-BV-OT), where model selection is based on **cls.** accuracy and OT over the validation set, performance declines considerably* 96
- Figure 4.2 Model selection with early stopping at the epoch indicated with a dot using LOC accuracy. Different approaches are compared for pseudo-bbox annotation versus GT annotations (oracle). Fig(B) is a zoom of Fig.(A) between epochs 0 and 5. Results are reported over CUB validation set using the CAM method (Zhou *et al.*, 2016) with IoU as a LOC measure. LOC curves with pseudo-bbox annotations typically have similar behaviour to the Oracle GT bbox annotations, making them suitable for WSOL model selection. They increase and reach their peak for a similar number of epochs, followed by a decline and stagnation in performance. In contrast, the **cls.** curve reaches its peak toward the end of training when LOC performance has already degraded. **cls.** accuracy is therefore inadequate as a WSOL selection criterion to achieve high LOC accuracy performance.

	Misalignment between LOC and cls. behaviour has been studied further in (Choe, Oh, Lee & et. al, 2020)	105
Figure 4.3	Our proposed LOC pseudo-bbox annotator $\hat{o}$. A set of bbox proposals is initially extracted using a region proposal model, and from them, discriminative ones are selected using pointing game analysis (Zhang, Bargal, Lin & et. al, 2018a). In the case where multiple bboxes are selected, the classifier confidence over the foreground region is used to select the most discriminative bbox	107
Figure 4.4	(A) The heatmap illustrates the inaccuracies in noisy bboxes at various noise levels generated by augmenting GT bboxes. (B) The blue line represents the average LOC accuracy over various configurations on the test set, incorporating maxima, minima, and standard deviation; the model is selected using a validation set comprising noisy bboxes with varying noise levels. In contrast, orange lines represent the average maximum and minimum performance of a model chosen during hyperparameter optimization over GT for configurations analogous to those employed with noisy labels. (C) Histogram illustrating the variance in model selection epochs when using bboxes at various noise levels w.r.t ground truth boxes for experiments with identical configurations, highlighting tendency around zero. (D) Model selection epoch frequency using GT and noisy boxes across different configurations and noise levels. (E) Illustration of pseudo-bboxes along with the GT over (E.1) CUB and (E.2) ILSVRC validation set	110
Figure 4.5	Illustration GT masks, noisy masks at various erosion levels	122

LIST OF ABBREVIATIONS

ACoL	Adversarial Complementary Learning
ADL	Attention-based Dropout Layer
BG	Background
BGC	Bridging the Gap between Classification and Localization
C ² AM	Contrastive Class-Agnostic Activation Map
CAA	Class-Aware Attention-based Affinity
CAM	Class Activation Mapping
CAM-IVR	Class Activation Mapping with Implicit Visual Relevance
CDM	Cue-Digging Module
CL	Classification
CLIP	Contrastive Language-Image Pretraining
CLS	Class Token
CNN	Convolutional Neural Network
CREAM	Class Re-Activation Mapping
CRF	Conditional Random Field
CUB	Caltech-UCSD Birds
CutMix	Cut and Mix Regularization
DINO	Self- <i>Distillation</i> with <i>no</i> labels
DL	Deep Learning

DRA	Discriminative Region Alignment
DeiT	Data-efficient Image Transformer
DeiT-S	Data-efficient Image Transformers Small
DiPS	Discriminative P roposals S ampling
EVA	Efficient Vision-Augmented Transformer
F-CAM	Full Resolution Class Activation Map
FAM	Feature Activation Map
FG	Foreground
FG	Foreground
FG/BG	Foreground/Background
GAP	Global Average Pooling
GAP	Global Average Pooling
GT	Ground Truth
HaS	Hide-and-Seek
HaS	Hide-and-Seek
I2C	Inter-Image Communication
ICL	In-sample Contrastive Learning
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IOP	Intersection over the Predicted Bounding Box
IS-CAM	Integrated Score-CAM

IoU	Intersection over Union
KD	Knowledge Distillation
LCTR	Local Continuity Transformer
LME	Localization More Error
LPE	Localization Part Error
LayerCAM	Layer-wise Class Activation Mapping
MIL	Multiple Instance Learning
MLP	Multi-Layer Perceptron
MaxBoxAcc	Max Box Accuracy
NL-CCAM	Non-local Combinational Class Activation Maps
OT	Otsu's Thresholding
PCM	Pre-trained CLIP Model
PDM	Part Discovery Module
PE	Patch Embedding
PG	Pointing Game
PxAP	Pixel-Wise Average Precision
QR	Quick Response (Decomposition)
RAM	Activation-Restricted Module
RLA	Region-to-Label Alignment
RPAM	Relational Patch-Attention Module

RPN	Region Proposal Network
ResNet	Residual Network
RoI	Region of Interest
SAT	Spatial-Aware Token
SCG	Self-Correlation Map
SCM	Spatial Calibration Module
SGD	Stochastic Gradient Descent
SGR	Self-Guided Region Rectification
SLT	Strengthen Learning Tolerance
SPA	Structure-Preserving Activation
SPG	Self-Produced Guidance
SPOL	Shallow Feature-Aware Pseudo Supervised Object Localization
SS	Selective Search
SSL	Semi-Supervised Learning
SST	Self-Supervised Transformer
T-SNE	t-Distributed Stochastic Neighbor Embedding
TCAM	Temporal Class Activation Mapping
TE	Text Embedding
TPSM	Token Priority Scoring Module
TRT	Re-attention Transformer

TRT	Token Refinement Transformer
TS-CAM	Token Semantic Coupled Attention Map
TelDrone	Telecommunication Drone Dataset
TrCAM-V	Transformer based CAM for Videos
U-Net	U-Net Convolutional Neural Network Architecture
UDA	Unsupervised Domain Adaptation
VAE	Variational Autoencoder
VGG	Visual Geometry Group
ViT	Vision Transformer
ViTOL	Vision Transformer for Object Localization
ViTOL-GAR	Vision Transformer for Object Localization with Global Attention Rollout
ViTOL-LRP	Vision Transformer for Object Localization with Layer-wise Relevance Propagation
WSOL	Weakly Supervised Object Localization
WSVOL	Weakly Supervised Video Object Localization
YTOv1	YouTube-Object v1.0
YTOv2.2	YouTube-Object v2.2

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\{\mathcal{M}_{ij} \geq \tau\}$	Binary mask obtained by thresholding at τ
A	Affinity measure
b	Binary map of connected regions
bx	Bounding box proposal
C	Number of classes in the dataset
CE	Cross-Entropy Loss Function
class	Class token
$\mathbb{C}^-$	Background pixels
$\mathbb{C}^+$	Foreground pixels
D	Fixed embedding size
$\mathbb{D}$	Training set
$\mathcal{D}_{train}$	Training dataset
δ	Threshold for Intersection over Union
$\cdot$	Missing bounding box supervision
$\mathcal{D}_{test}$	Test dataset
$\mathcal{D}_{val}$	Validation dataset
D_I	Decoder
e	Attention map from a class token
E_{cl}	Class token embeddings

$\ell(\cdot, \cdot)$	Localization similarity measure between two bounding boxes
E_T	Text Encoder
E_V	Visual Encoder
$f(\cdot; \theta)$	Model function with parameters θ
$f(x; \theta)$	Model's classification prediction for image x
$f(x; \theta)^b$	Predicted bounding box for image x
f_c	Classification head
f_e	Transformer-based encoder
f_i	Input video frame
f_l	Localization head
f_x	External pre-trained classifier
H	Pixel-wise alignment loss
$\mathbf{I}_{CL}$	Image Classification Loss
IoU	Intersection over Union
K	Number of classes
$\mathcal{L}$	Loss function
$\mathbf{L}$	Total Training Loss
λ	Weighting coefficient
λ_{ASL}	Weight for absolute size loss
λ_{CRF}	Weight for CRF loss

λ_{PAL}	Weight for pixel alignment loss
$\mathcal{L}_{ASL}$	Absolute size loss
$\mathcal{L}_{CLS}$	Classification loss
$\mathcal{L}_{CPA}$	Constrained Pixel Alignment loss
$\mathcal{L}_{CRF}$	Conditional Random Field loss
$\mathcal{L}_{PAL}$	Pixel alignment loss
$\mathcal{L}_{Total}$	Total loss
$\mathbf{L}_{KD}$	Knowledge Distillation Loss
M	Softmax activation maps
$\hat{M}$	Predicted localization map
$\mathbf{M}$	Localization Map
$\mathcal{M}_{ij}$	Pixel value at location (i, j) in the localization map
$\mathcal{M}^c$	Localization map for class c
$\mathcal{M}(\theta, \mathcal{D}_{val})_{\hat{o}}$	Average localization performance over validation set using pseudo-annotations
$\hat{M}_0$	Background channel of localization map
$\hat{M}_1$	Foreground channel of localization map
MaxBoxAcc	Localization accuracy metric used in WSOL
$\mathbf{M}^-$	Background map obtained through Otsu's threshold
$\mathbf{M}^+$	Foreground map obtained through Otsu's threshold
M'	Pseudo-label map

n^-	Number of background pixels
n^+	Number of foreground pixels
$o(x_i)$	Oracle manual bounding box annotation for image x_i
$\hat{o}$	Pseudo-annotator function
$\hat{o}(x_i)$	Pseudo-bounding box annotation for image x_i
$\mathcal{O}^-$	Operation to select bottom $n\%$ pixels in a map
$\mathcal{Q}$	Discrete image domain
$\mathcal{O}^+$	Operation to select top $n\%$ pixels in a map
p	Probability distribution
$\mathbb{P}$	Set of top-K discriminative proposals
ϕ	Sampling function
Pr	Probability function
$\mathbf{P}_{\text{CL}}$	Patch Classifier Loss
$\mathcal{R}$	CRF loss function
$\mathbb{R}^{H,W}$	Set of real numbers with dimensions $H \times W$
$\mathbf{S}$	Foreground and background maps
$\mathbf{S}^0$	Background map
$\mathbf{S}^1$	Foreground map
T	Number of encoder blocks
$\mathbb{T}$	Set of all bounding boxes

$\mathbb{T}$	Training set
τ	Softmax temperature
τ	Threshold for converting localization map into a binary mask
θ	Model parameters
θ	Model parameters
θ	Model parameters
θ_c	Parameters of the classification head
θ_e	Parameters of the transformer-based encoder
θ_l	Parameters of the localization head
T_{SST}	Transformer model
v_i	Video sequence of frames
W	Affinity matrix
W	Width of the image or localization map
$x/X/X_I$	Input image
x_j	Single frame in a video
Y	Pseudo-label mask
y_i	Image-class label for image x_i
a_p	Weight for Patch Embedding in Global Embedding
$f(h, t_k)$	Classification Score Using Global Image Embedding
$f(z_p, t_y)$	Similarity Between Patch Embedding and Text Embedding

$g(z_p)$	FG/BG Classification Score
$\mathbf{h}$	Global Image Embedding
$\lambda_1, \lambda_2, \dots, \lambda_n$	Weighting Factors for Loss Terms
ω	Set of Patch Locations
ω^+	Set of FG Patches
t_k	Text Embedding for Class k
y	Class Label
y'_p	Patch Pseudo-Label
$\hat{z}_p$	Similarity Score of Patch Embedding
z_p	Patch Embedding

INTRODUCTION

Weakly supervised learning (WSL) has recently emerged as a compelling paradigm in computer vision, driven by its ability to minimize the reliance on fine-grained annotations. Rather than requiring large-scale datasets annotated with pixel-level or bounding-box labels, WSL is able to train models with limited or imprecise supervision (e.g., image-level labels), thereby substantially reducing the time and cost required for data annotation. Recently, WSL has been employed for different tasks, such as object detection and segmentation, providing a learning paradigm that mitigates the significant bottleneck caused by the lack of extensive annotations. Object localization, however, focuses on predicting bounding boxes that precisely enclose objects within an image. Unlike classification, which only predicts class labels, localization aims to predict both the “what” (class) and the “where” (position of objects) as illustrated in Fig.0.1.

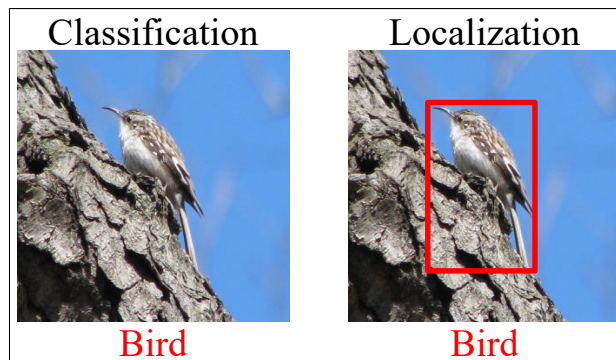


Figure 0.1 Classification vs object localization
Images obtained from Welinder *et al.* (2010)

Building on these ideas, weakly supervised object localization (WSOL) has garnered considerable attention. WSOL aims to identify and locate objects using only image-level labels indicating the presence of object classes, thus reducing annotation overhead compared to pixel-level labelling. However, WSOL presents unique challenges compared to fully supervised approaches. The core difficulty lies in the inherent ambiguity of weak supervision, as image-level labels provide no explicit information on the exact locations or boundaries of objects. Consequently, this

ambiguity complicates the task of accurately identifying the spatial extent of an object, leading to suboptimal localization performance. Moreover, as illustrated in Fig.0.2, WSOL operates under partial supervision, making it more practical for large-scale applications. Specifically, it enables the learning of models when fine-grained supervision is very difficult to obtain.

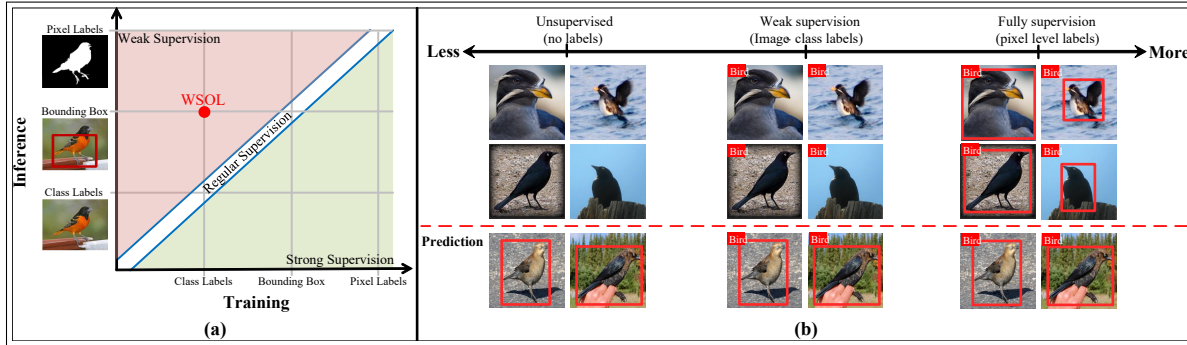


Figure 0.2 (a) Learning spectrum illustrating the WSL. The red area represents WSL, while the green area represents a conventional full-supervised learning setting. Moreover, the axis here represents human annotation. Specifically, the task of predicting bboxes or pixel-level labels from image-level labels is referred to as WSOL. (b) Zoom-in learning spectrum from the perspective of object localization

0.1 Challenges in WSOL

In the WSOL, the learning process must tackle challenges such as intra-class variations in appearance, transformation, scale, and aspect ratio, as well as learning under uncertainty due to the discrepancy between image-level annotations and the need for instance-level localization. The key is to propagate image-level supervisory signals to instance-level annotations for training, which involves dealing with ambiguous and noisy information. Specific challenges in WSOL include:

- **Learning with insufficient samples for different classes:** Similar to challenges in other paradigms, training robust object localizers in weakly supervised fashion conditions becomes challenging when class instances are limited. This causes the model to overfit in some particular instances of class. Due to this, data distribution across different object categories

tends to exhibit a long-tailed nature. These challenges are further exaggerated in WSOL, where limited supervision prevents the model from exploring different objects, impacting the generalization and localization performance of a model.

- **Learning with noisy labels:** Class labels referring to wrong or non-dominant objects prevent the model from confidently learning class-specific feature representations. This ambiguity often results in incorrect associations between underlying objects and labels. Label noise can also occur if the model confuses object parts of different instances having similar visual features (e.g., mistaking a wolf as a husky). This causes the model to learn incorrect representation.
- **Learning with domain shifts:** For different classes, the image training samples may lack diversity in object shape, appearance, scale, and viewpoint. This leads the model towards the poor performance on test samples that differ significantly from the training data. This occurs due to a large shift of samples in training and test time. as well as from differences in annotation granularity (class-labels in training vs bbox at test time).
- **Discriminative region bias:** In WSOL, models are typically trained with dataset with class labels. During the training, they try to minimize mutual information between different instances of the same class and class label. As a result, they often focus on the most discriminative parts, leading to the under-activation of object parts. Alleviating this discriminative region bias remains an open problem.
- **Resolution limitations:** CNNs generate downsampled feature maps, leading to low-resolution activation maps that compromise localization performance. Although transformers offer higher-resolution maps, they are computationally intensive and still produce maps that lack fine-grained details.
- **Cross-modal knowledge distillation:** Leveraging vision-language models like CLIP for WSOL requires distilling spatial localization cues from multimodal embeddings into a

vision-only model, a challenging task due to the inherent differences in how these models encode visual and textual information.

- **Unrealistic evaluation protocols:** Current evaluation protocols for WSOL rely on bbox annotations for model selection, which is not aligned with the weakly supervised paradigm. This leads to evaluations that do not reflect the true performance of WSOL methods in real-world applications.

0.2 Motivation

WSOL is a promising solution for object localization compared to its counterpart, which is a fully supervised approach requiring instance-level annotations. It significantly reduces annotation costs by learning to localize objects from image-level labels. Despite its advantages, WSOL remains a challenging problem. Training a model with only image-level labels often forces models to highlight the most discriminative object features rather than highlighting the full spatial extent of an object.

Recent advancements in self-supervised learning, transformer architectures, and vision-language models have opened promising avenues for WSOL tasks. In particular, vision transformers can capture long-range dependencies among different object parts, alleviating the under-activation of localization maps extracted via conventional methods, e.g. CAMs. Moreover, large-scale vision-language models have demonstrated an ability to localize an object using the prompt, enabling us to distill robust localization information to the vision model. Nevertheless, fundamental limitations persist:

- **Under-activations in localization maps:** WSOL model is often trained with class-level labels, and classification-driven training objectives often exploit discriminative regions across different instances of the same class. This leads towards under-activation in localization

maps, requiring us to train WSOL model in end-to-end fashion for producing localization maps.

- **Leveraging vision-language models for localization:** While CLIP-based methods yield competitive localization performance, they still require explicit class labels or external classifiers to obtain localization maps. This introduces errors, particularly when the classifiers are inaccurate or when classes are not known *a priori*. Distilling the vision-language model into a vision model could enable the joint optimization of the model for both tasks simultaneously, e.g. classification and localization.
- **Learning from pseudo-labels:** Generating localization maps using explainability methods (e.g. CAMs) leads towards under-activation within localization maps. To deal with this issue, we could employ pseudo-labels for training the localization model in end-to-end fashion. It enables the model to highlight different object parts for reliable localization performance.
- **Unrealistic evaluation protocol:** Current evaluation in WSOL relies on bbox annotations specifically for model selection and threshold estimation, which contradicts the WSOL paradigm. This leads to the overestimation of localization performance, limiting its deployments in real-world scenarios. This motivates us to build a realistic evaluation protocol for WSOL.

This thesis proposes to address the above-discussed issues for improving localization performance. For instance, we introduce *DiPS* to harvest robust pseudo-labels from self-supervised transformers using its attention maps from different layers. These pseudo-labels are then used to sample pseudo-pixels for training the model. This pipeline rectifies the usual pitfalls of CAM approaches by sampling from different candidate proposals—selected by an external classifier—to yield robust pseudo-labels. In the same relam, *TeD-Loc* proposes to distill patch-level localization information from text embeddings to our localization module guided by pseudo-labels. Aligning visual embeddings with text embeddings for foreground and background patches can highlight different object parts, improving localization performance. Lastly, *A realistic protocol for*

evaluation of WSOL proposes to align WSOL paradigm during training and evaluation by employing pseudo-boxes for evaluation. This evaluation protocol offers a realistic evaluation framework for WSOL aligning with WSOL paradigm. In summary, our work marks a significant progression in WSOL. It proposes frameworks for training WSOL model in end-to-end fashion, along with a realistic evaluation protocol.

0.3 Research Objectives and Contributions

This thesis aims to address these challenges by proposing novel methodologies for enhancing weakly supervised object localization. The contributions are structured into three main chapters:

1): Introduces **Discriminative Pseudo-Label Sampling (DiPS)**, a method that utilizes self-supervised Vision Transformers to generate high-quality pseudo-labels for training WSOL models. DiPS significantly improves localization coverage and accuracy compared to CAM-based methods:

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Aydin Sarraf, Eric Granger, “Discriminative Sampling of Proposals in Self-Supervised Transformers for Weakly Supervised Object Localization”, IEEE/CVF Conference on Applications of Computer Vision (WACV) Workshops, pp. 155-165, 2022.
- **Shakeeb Murtaza**, Marco Pedersoli, Aydin Sarraf, Eric Granger, “Leveraging Transformers for Weakly Supervised Object Localization” in IAPR Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR), 2024, pp. 195-207.
- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Aydin Sarraf, Eric Granger. “DiPS: Discriminative pseudo-label sampling with self-supervised transformers for weakly supervised object localization”, Image and Vision Computing, Volume 140, 2023.

2): Develops a novel framework for **distilling localization knowledge from CLIP** into a vision-only localizer, enabling precise object localization using only image-level labels. This

approach bridges the gap between vision-language models and vision-specific tasks, making WSOL more practical for diverse applications:

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Eric Granger, “TeD-Loc: Text Distillation for Weakly Supervised Object Localization”, (Submitted to IEEE/CVF Winter International Conference on Computer Vision (ICCV), 2025.

3): Proposes a **realistic evaluation protocol** for WSOL, which eliminates the dependency on ground-truth bboxes by using pseudo-bboxes generated through unsupervised region proposals. This protocol provides a fairer assessment of WSOL models in real-world scenarios:

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Eric Granger, “A Realistic Protocol for Evaluation of Weakly Supervised Object Localization”, IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2025.

These contributions advance the state of the art in weakly supervised object localization and establish a foundation for future research in this field. A full list of publications generated as a result of this research is provided in Appendix III.

0.4 Thesis Organization

An illustration of thesis organization is presented in Fig.0.3. In **Chapter 1**, we presented a background and literature review on object localization. It begins by introducing traditional ML and DL models. Then we explain WSL and shift the focus to WSOL. Next, we present a taxonomy of WSOL methods, starting with few-shot contrastive-pretrained models, bottom-up and top-down techniques, and problem formulation, then covering few-shot contrastive-pretrained models, bottom-up and top-down techniques. Also, a critical analysis of these approaches highlighting their issue is presented. **Chapter 2** present a new training paradigm that was proposed to train transformers for WSOL while harvesting pseudo labels from self-supervised transformers. This work corresponds to the paper “DiPS: Discriminative Pseudo-Label Sampling

with Self-Supervised Transformers for Weakly Supervised Object Localization”, published in image and vision computing. **Chapter 3** presented TeD-Loc, a training pipeline for distilling localization information from the vision-language model that enables our model to produce localization without requiring the language part of the models. The paper related to this method is “TeD-Loc: Text Distillation for Weakly Supervised Object Localization,” which was submitted to the IEEE/CVF Winter International Conference on Computer Vision. **Chapter 4** highlights the unrealistic evaluation proposed in WSOL, leading towards overestimation of localization performance not aligning with standard WSOL protocol. For this, a new evaluation is proposed for WOSL that provides LOC information for model selection without relying on manual bbox annotations. Our protocol consistently achieved competitive performance compared to models selected using ground-truth bbox annotation over the validation set. Furthermore, a paper related to this work, titled “A Realistic Protocol for Evaluation of Weakly Supervised Object Localization” was published in IEEE/CVF Winter Conference on Applications of Computer Vision. In Appendix I and II, papers are presented that have supporting material for the above-presented chapters. Finally, a list of publications during my Ph.D is presented in Appendix III.

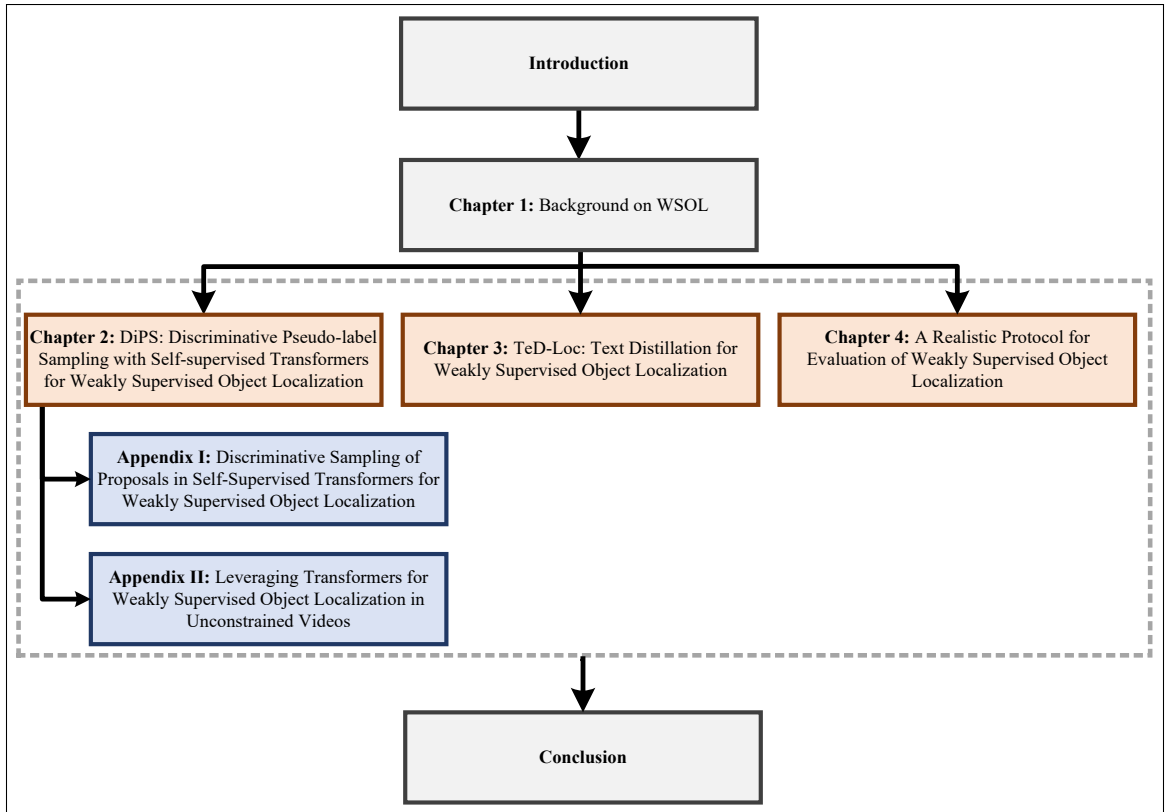


Figure 0.3 Thesis Organization. The arrows show the dependencies among the chapters and appendices

CHAPTER 1

BACKGROUND AND LITERATURE REVIEW

This chapter lays the groundwork for understanding weakly supervised object localization (WSOL). We start by introducing machine learning models, followed by an overview of both traditional and deep learning models. Then, we explain weakly supervised learning (WSL) and how it exploits weak labels to train the model. Building on this foundation, we provide an overview of visual recognition and the importance of WSL in visual recognition. We then present WSOL problem formulation for images and videos and its relation with other learning paradigms. Finally, we present a taxonomy of existing WSOL methods and a comprehensive review of related methods.

1.1 Machine Learning

Machine learning (ML) refers to learning from data and enabling machines to perform tasks without explicit intervention. Instead of relying on pre-defined rules/instructions, ML models are able to extract rules by identifying patterns in the data and are able to generalize on unseen data and tasks. The impact of ML is pervasive in daily tasks, often invisibly transforming domains from self-driving cars to healthcare diagnostics. However, ML is driven by the availability of data and the computational power to train ML models.

Based on the availability of data for training these models, we broadly categorized models based on these learning paradigms into three categories: *Supervised learning* relies on explicit pairs of inputs and their corresponding outputs (labels), aiming to minimize the mutual information between inputs and labels by minimizing task-specific loss, thereby ensuring generalizability to unseen data. *Unsupervised learning* involves training models that can derive insights solely from input data by exploiting underlying structures present in that data without requiring explicit labels. In contrast, *weakly supervised learning* enables training of models from data labelled with partial or noisy labels. Despite this weak supervision signal, models trained in this paradigm can learn useful representations surprisingly well.

1.1.1 Traditional ML

Traditional ML methodologies rely on the segregated features of input data that are used for training and prediction. These ML methods often require handcrafted features—selected by domain experts—are fed into different models, e.g., SVMs, adaboost, or random forests. These methods often thrive in constrained environments with relatively small training sets, where a well-curated set of input features can effectively capture the essential variations in the data.

A classic example of traditional ML is the support vector machine (SVM) (Cortes, 1995), which constructs a hyperplane by forming the boundaries between distinct class features for identifying class-specific attributes. This approach frequently grants SVMs strong generalization capabilities, especially in contexts with high-dimensional data. Ensemble methods, on the other hand, leverage features from multiple learners to produce one robust predictor. Take random forests (Breiman, 2001) as an example: they fuse different uncorrelated decision trees through bagging, where these trees are fitted using a bootstrap sample of the data. The individual trees' predictions are then aggregated to produce output that tends to have lower variance. Boosting algorithms, such as Gradient Boosting (Friedman, 2001) and Adaboost (Freund & Schapire, 1996) and, iteratively upweight training examples that are difficult to classify, thereby adapting the model in a progressive fashion.

Though these techniques form the bedrock of classical ML, they typically rely on the assumption that hand-crafted features extracted is able to capture all relevant variations in the input data. This assumption quickly becomes tenuous as ML algorithms are adopted to different domains, specifically into high-dimensional spaces, or seek to model the complexities of unconstrained real-world data. Indeed, the performance of these traditional algorithms may plateau once the representational power becomes saturated. The natural evolution—driven by both the surge of large-scale datasets and advances in computational hardware—has thus led to deep, multilayered approaches, which learn not only how to classify but also how to represent the data itself. This leads towards development of deep learning models which can be fitted in an end-to-end fashion on the input-output data without relying on hand-crafted features.

1.1.2 Deep Learning

Deep learning (DL) falls under the umbrella of ML, which uses deep neural networks to enable end-to-end learning without requiring hand-crafted features. Traditional programming requires a lot of effort, from detailed design to the development of the program. This approach is not feasible for solving real-world problems, e.g., detecting house numbers in an image. To develop a program for house numbers detection, we are provided with a dataset containing a large number of labelled images, i.e. each instance is tagged with a certain number that an image contains. In a conventional setting, we study each image along with its corresponding tag and come up with a set of rules. Developing such a program is quite challenging and may not be generalized to unseen images. DL algorithms can be used to infer patterns from underlying data without explicit program design to overcome this issue. These algorithms' performance relies on the quality of training data, and the goal of these algorithms is to learn rules from labelled data that can correctly predict unseen instances. The DL algorithm approximates a function $\hat{y}_i = f(x_i)$ by optimizing a loss function $\mathbb{L}$. These models are commonly known as artificial neural networks (ANNs) inspired by the biological neuron. Like biological neurons, it consists of multiple perceptrons (a fundamental building block of ANN). Furthermore, these neurons are stacked to make a layer and multiple layers are stacked together to make a DNN.

A key breakthrough in DL emerged with the introduction of AlexNet (Krizhevsky, Sutskever & Hinton, 2012) that substantially outperforms existing DNN. Since then, considerable efforts have been devoted to exploring the potential of DL models by introducing new network architectures, optimization strategies, losses and datasets. These developments led to the development of numerous convolutional neural network (CNN) variants, such as VGG (Simonyan & Zisserman, 2014), ResNet (He, Zhang, Ren & Sun, 2016), and Inception (Szegedy, Liu, Jia & et. al, 2015), which have collectively surpassed human-level performance on several computer vision benchmarks such as ILSVRC (Russakovsky, Deng, Su & et. al, 2015a).

While early CNN typically only accept (two-dimensional) 2D input, subsequent advances extended these networks to three-dimensional architectures (3D CNNs) (Dosovitskiy, Beyer,

Kolesnikov & et. al, 2020). Moreover, long short-term memory (LSTM) networks (Hochreiter & Schmidhuber, 1997), a variant of recurrent neural networks, have been introduced to capture longer-term temporal dependencies by employing recurrent connections. This allows for modelling of long-range dependencies within input data. However, LSTMs struggle to capture very long-range dependencies. To address this issue, transformer architecture (Vaswani, Shazeer, Parmar & et. al, 2017) was introduced to capture long-range dependencies for sequence modelling in natural language processing (NLP). Motivated by their success, vision transformers (ViTs) citep(Dosovitskiy *et al.*, 2020) were proposed and demonstrated state-of-the-art performance across various tasks.

Unfortunately, training a DL model for different tasks requires a large amount of data annotated by human experts. Thus, a natural question is: Can we leverage broad, image-level labels to learn where objects reside in the image? That is exactly where weakly supervised localization (WSL) fits in.

Unfortunately, training DL model for these tasks dataset requires a large dataset amount of annotated data requiring human experts to annotate data. Thus, a natural question is: Can we leverage broad, image-level labels to learn where objects reside in the image? That's exactly where weakly supervised localization (WSL) fits in.

1.1.2.1 Related DL models

Vision transformers. Transformers-based models employ a very special type of attention commonly known as self-attention, which computes the relevance between all inputs. Transformers are also recognized for their success in image classification (Dosovitskiy *et al.*, 2020; H. Touvron, Cord, Douze & et. al, 2021) and WSOL (Gao *et al.*, 2021). The transformer used in vision commonly known as ViTs and crops the image into small overlapping patches to compute the relevancy between them. It consists of n cascaded blocks, each containing multi-headed attention followed by multi-layer perceptron (MLP). First of all an input image x of size $W \times H$ is divided into N patches of resolution $(W/P) \times (H/P)$, here P denotes the patch size. These

patches are then linearly projected to a fixed embedding size d along with a learnable class token $[cls]$; $\{[cls]^{1 \times d}, t_1^{1 \times d}, t_2^{1 \times d}, \dots, t_3^{1 \times d}\}$. Furthermore, the feature of $[cls]$ token is fed to an MLP for the prediction of class scores. Mathematically,

$$p = \text{Softmax}(\text{MLP}([cls])) \quad (1.1)$$

Here, MLP denotes the classification block for predicting the probabilities of concerned classes with classification loss $-\log(p_y)$.

Hybrid ViTs (Transformers and CNNs). Transformers are exclusively based on self-attention operation and fully-connected layers. However, recently two variants of transforms have been proposed,

Convolution Vision Transformers (CvTs): In visual recognition, Convolution is the popular operation for calculating the feature maps by using a small number of parameters. Similarly, in ViTs, a fully connected layer has been replaced with convolution to reduce the parameters and computational complexity (Wu, Xiao, Codella & et. al, 2021).

Transformers based on CNN features: In recent papers, a CNN-transformer pipeline has been proposed in which CNN is used to feed feature maps to the transformer. The prominent method in this category is the object detection transformer (DETR) (Carion, Massa, Synnaeve & et. al, 2020).

Self-supervised transformer. DINO (Caron *et al.*, 2021) was proposed to harvest regions of interest without class-level labels. To achieve this goal author trained two networks possessing similar architecture and initialized with the same weights; student N_{θ_s} and teacher N_{θ_t} to match their probability distribution as shown in Fig.1.1. Furthermore, their probability P is calculated by normalizing the output of the concerned network by temperature softmax function as,

$$P_s(x)^{(i)} = \frac{\exp(N_{\theta_s}(x)^{(i)} / \tau_s)}{\sum_{k=1}^K \exp(N_{\theta_s}(x)^{(k)} / \tau_s)}, \quad (1.2)$$

For both student and teacher network x is augmented in different way, x_1 and x_2 . They represent two distorted views of input image x calculated using strategy defined in (Caron, Misra, Mairal & et. al, 2020). Specifically, author generated a set of views for each input image containing several local views and two global views as represented by x_1 and x_2 . Here, the local view contains less than 50% and the global view covers more than 50% of the area of the image. Furthermore, the loss is calculated between the output of two networks as

$$\min_{\theta_s} \mathcal{L}(P_s(x_1), P_t(x_2)) \quad \therefore \mathcal{L}(y, z) = -y \log z \quad (1.3)$$

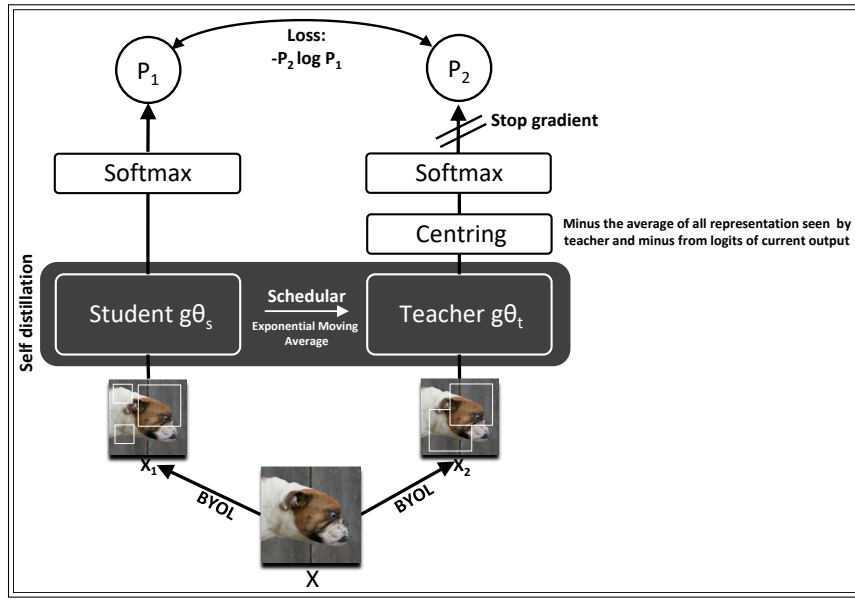


Figure 1.1 Illustration of self-distillation with *no* labels
Adapted from Caron *et al.* (2021)

1.2 Weakly Supervised Learning

Although supervised learning algorithms for visual recognition achieve superhuman performance, their accuracy relies on the labelling quality of training data. In critical domains, the data annotation process is very complex due to the large amount of data and the involvement of domain experts. To deal with this issue, we need algorithms that can learn from weakly supervised data.

ML algorithms that can learn from weakly annotated data may fall under the category of weakly supervised learning (WSL). WSL techniques may be adopted in the critical domain because it does not require fully annotated data, and based on the scope of annotations, we can classify these techniques into three categories: incomplete supervision, inexact supervision and inaccurate supervision, as depicted in Fig.1.2.

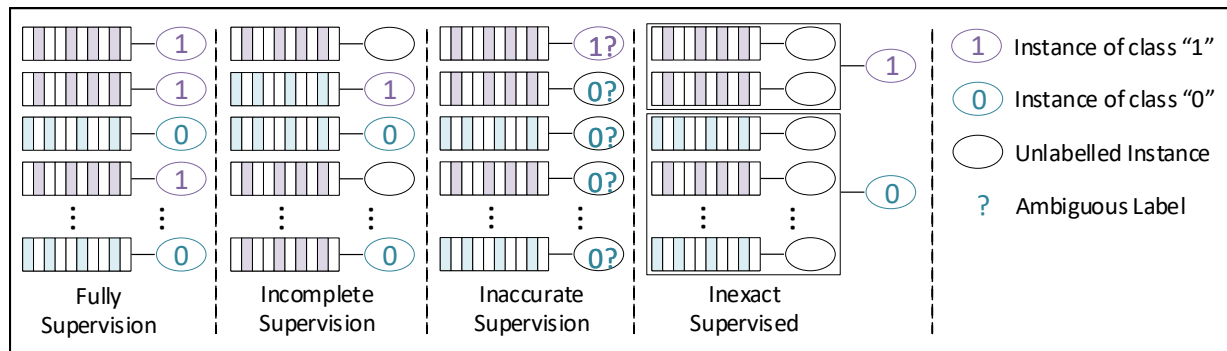


Figure 1.2 Types of weakly supervised settings
Adapted from Zhou (2018)

Incomplete Supervision: In incomplete supervision, learning algorithms extract patterns from a large amount of unlabeled data by utilizing partially annotated data. For example, in medical imaging, we can retrieve a large amount of data, but it is not easy to annotate the whole data as it requires the involvement of a domain expert. We can estimate a function from incomplete labelled data by semi-supervised learning (Van Engelen & Hoos, 2020) or active learning (Settles, 2009) algorithms. Semi-supervised learning works without human intervention by estimating unlabeled data labels with the help of minimal annotated data. In contrast to semi-supervised learning techniques, active learning algorithms require human assistance to learn from underlying data. These techniques assume that the annotations for the most valuable instances can be obtained from a third party.

Inaccurate Supervision: If annotations of training data are inaccurate or labels contain noisy information, inaccurate supervision techniques can extract useful information from noisy labels.

Inexact Supervision: If coarse-grained annotations are unavailable, we may end up with inexact supervision. For example, learning a function for semantic segmentation by utilizing image-level labels. Various approaches were proposed in the literature to achieve maximum performance by using a limited number of labels while keeping ambiguity to a minimum, and the most prominent approach from all of them is multi-instance learning (MIL). MIL is a type of machine learning in which, instead of feeding a single labelled instance to the learner, multiple instances as a set of labelled bags are given to the learner. For example, in the case of a multi-instance binary classification problem, a bag consisting of multiple negative instances is given to the learner. In the case of positive instances, the bag has positive labels. One key benefit of this approach is that only a single label is required for a bag of unlabeled instances. The learner’s task is to predict the labels of unseen bags based on the concept learned from training bags.

1.3 Visual Recognition

Visual object recognition broadly encompasses computer vision techniques that deal with identifying and localizing objects of various categories within an image. DNNs have been widely adopted for different visual recognition tasks, including but not limited to object detection and medical image diagnosis, due to their recent success in different tasks (Grauman & Leibe, 2011; Andreopoulos & Tsotsos, 2013). In general, visual object recognition can be loosely categorized into four groups: Image classification, single-object localization, object detection and segmentation.

In this work, our primary focus is on single-object localization within the weakly supervised paradigm, commonly referred to as “weakly supervised object localization” (WSOL). In this paradigm, only image-level class labels are provided during training—no explicit bounding boxes or segmentation masks are often described in the literature as incomplete supervision. The details and taxonomy of related methods for WSOL are in the following section.

1.4 A Taxonomy of Methods for Weakly Supervised Object Localization

In this section, we introduce a taxonomy (Fig.1.3) and provide a review of deep learning (DL) based WSOL methods capable of generating both image-level classifications and pixel-level localization maps. These methods require only image-level class labels during training and fall under weakly supervised object localization. Furthermore, we can categorize WSOL methods

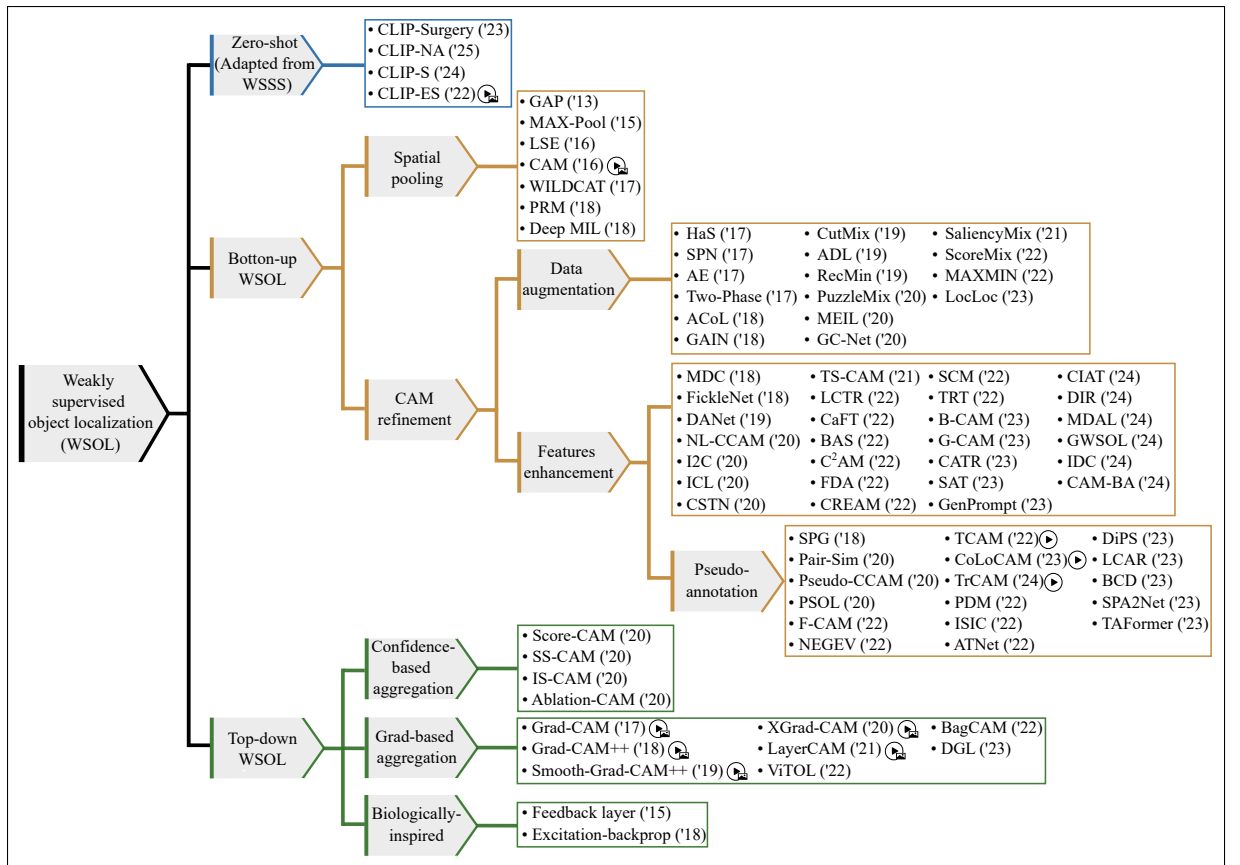


Figure 1.3 Taxonomy of deep WSOL methodologies trained using image-level class labels. Methods are categorized and arranged chronologically within each group: (1) Bottom-up approaches, which extract activations for producing localization maps during forward-pass information exclusively; (2) Top-down approaches, which requires information from both forward and backward passes; and (3) Zero-shot approaches, based on CLIP model and was proposed within the Weakly Supervised Semantic Segmentation (WSSS) paradigm. Icons in front of method names denote modality support: ▶ indicates WSOL methods supporting video data, while 📷 denotes support for both image and video modalities in the WSOL context. the missing icon signifies that the method is restricted to single-image input

into three categories. The first includes few-shot methods adapted from the weakly supervised semantic segmentation (WSSS) literature for application to WSOL tasks. The remaining two categories differ in how the localization maps are computed with respect to information flow through the network: (i) bottom-up methods rely on the forward pass to produce localization maps, and (ii) top-down methods incorporate backward information, in addition to the forward pass, to determine the specific image regions on which the network focuses.

Following this, we provide the formulation of WSOL task for both image and video modalities, establishing the framework used throughout this work. We then present a comprehensive review of methods belonging to each of these three categories.

1.4.1 Problem Formulation

Localization in images. In WSOL, the goal is to identify the spatial extent of objects using a model trained in a weakly supervised fashion. Specifically, given an image $x \in \mathbb{R}^{H \times W}$, the task is to determine which pixels belong to the object of interest by producing a dense binary mask $\mathbf{M} = (M_{11}, \dots, M_{HW})$, where each $M_{ij} \in \{0, 1\}$ indicates whether the pixel at position (i, j) is part of the object. This mask can represent either a bounding box or a pixel-wise object localization map. However, in a practical scenario, a model produces a continuous activation-alike map that is thresholded to produce a box or dense localization map. During training, the model only has access to image-level labels $y \in \{0, 1, \dots, C\}$, indicating the presence or absence of the object within each training image.

Formally, we define a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i, \cdot)\}$, where x_i is an input image, y_i is its class label, and $\cdot$ denotes missing localization supervision. The validation set $\mathcal{D}_{\text{val}} = \{(x_i, y_i, \cdot)\}_{i=1}^N$ and the test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i, o(x_i))\}$ include oracle bounding box annotations $o(x_i)$ for evaluation, though these are not to fit the model. WSOL models are trained with a classification objective to classify images while implicitly learning to localize objects through weak supervision.

Localization in videos. Instead of just localizing the spatial extent in image, the objective of WSVOL is to identify the spatial and temporal extent of objects with a model trained without

localization supervision. Models in WSVOL produces a sequence of dense binary masks $\mathbf{T} = \{\mathbf{T}_t\}_{t=1}^T$, where each $\mathbf{T}_t$ denotes the pixel-wise binary labels for the t -th frame. During training, the model has access only to video-level labels without explicit localization information, and the task is to infer pixel-level object presence from these global labels.

Moreover in videos, we define a training set $\mathcal{D}_{\text{train}} = \{(V_i, y_i, \cdot)\}$, where V_i is an sequence of images, y_i is its class label, and $\cdot$ denotes missing localization supervision for all frames. However, the validation set $\mathcal{D}_{\text{val}} = \{(V_i, y_i, \cdot)\}_{i=1}^N$ and the test set $\mathcal{D}_{\text{test}} = \{(V_i, y_i, o(V_i))\}$ include oracle bounding box annotations $o(x_i)$ for each frame separately to select and evaluate the model performance, though these are not to fit the model.

Each frame X_t is viewed as a bag of patches, with the object localization task involving predicting object presence for each patch. The label y serves as a collective annotation indicating the presence of the object within the video. WSOL for videos aims to learn a scoring function that assigns scores to patches, approximating the binary object labels through thresholding, with patches, video-level labels, and pixel-wise labels drawn i.i.d. from a joint distribution.

The training set $\mathcal{D}_{\text{train}} = \{(V_i, y_i, \cdot)\}_{i=1}^N$ consists of videos with video-class labels and missing localization supervision. The validation and test sets $\mathcal{D}_{\text{val}} = \{(V_i, y_i, \cdot)\}_{i=1}^M$ and $\mathcal{D}_{\text{test}} = \{(V_i, y_i, o(V_i))\}_{i=1}^P$ include oracle annotations for evaluation, which are not used in training. WSOL models are trained with a classification objective, learning to localize objects implicitly through weak supervision.

1.4.2 Related Frameworks

Inexact supervision or multiple-instance perspective. Additionally, WSOL can be framed from an inexact supervision or multiple-instance learning (MIL) perspective for both images and videos, extending the concept to a multiple-instance multiple-label (MMIL) setting. In this framework, each input—whether an image or a frame within a video—is treated as a bag of patches $(x_{11}, \dots, x_{HW})$ for images and $(X_{t,11}, \dots, X_{t,HW})$ for video frames, where each patch represents a local region sampled using a stride-1 sliding window. The localization task then

involves predicting the object presence T_{ij} for each patch. Under inexact supervision, the label y serves as a collective annotation, indicating whether at least one of the patches across the image or video sequence contains the object, thus aligning with the MIL paradigm. WSOL aims to learn a scoring function $s(x)$ or $s(V)$ that assigns scores to patches, approximating the binary object labels through thresholding. The MMIL approach extends this formulation by considering the temporal dependencies between frames in video data, capturing the presence of objects across both spatial and temporal dimensions.

Multi-task learning perspective. Typically, in WSOL, the model converges early for the localization task during training, subsequently declining localization performance as the training progresses. Conversely, classification accuracy tends to improve gradually, requiring more epochs to converge. So, selecting a model at a particular epoch that simultaneously maximizes performance for both tasks presents a significant challenge in WOSL. An alternative perspective on WSOL is through multi-task learning (MTL). In MTL, WSOL aims to optimize a model for two distinct but interrelated tasks: classification and localization. It highlights the inherent discrepancies between these. For this, we need to propose multi-task loss functions and learning to weight each task over different epochs. Leveraging such an approach not only aligns with the core objectives of WSOL but also reduces the computation complexity during inference, as only one forward pass is required for both tasks.

1.4.3 Few-shot Methods (Contrastive-pretrained).

CLIP (Radford *et al.*, 2021) is a foundational model designed to align visual and language representations, trained on 400 million image-text pairs collected from web data. By learning from paired data, CLIP produces a robust model capable of zero-shot adaptation to diverse tasks by computing similarities between images and textual descriptions. In WSOL, GenPrompt (Zhao, Ye, Wu & et. al, 2023) leverages CLIP to identify discriminative regions and employs VQGAN for embedding generation within a denoising process to localize objects; however, its computational complexity hinders real-time applicability.

Moreover, CLIP has been widely adapted for weakly supervised semantic segmentation (WSSS), prompting multiple approaches to leverage its capabilities for generating class activation maps (CAMs) without extensive labelled data. For instance, CLIMS (Xie, Hou, Ye & Shen, 2022a) utilizes CLIP to enhance the completeness of object regions within CAMs while suppressing BG regions. CLIP-ES (Lin *et al.*, 2023) employs GradCAM to extract CAMs directly from CLIP, demonstrating that activations can be generated without fine-tuning. SCLIP (Wang, Mei & Yuille, 2023) and NACLIP (Hajimiri, Ayed & Dolz, 2024) propose modifying the last attention block to produce score maps for each patch embedding, enabling segmentation map generation without requiring a backward pass. Similarly, (Yang & Gong, 2024) introduces learnable prompts in CLIP and uses pseudo-labels from SAM (Kirillov, Mintun, Ravi & et. al, 2023) to fine-tune for WSSS.

1.4.4 Bottom-up Techniques

Bottom-up methods produce localization maps by harvesting pixel-level activation from different network layers during forward pass in the network. Seminal methods proposed aggregating activation from different layers of the network weighted by class-specific features. However, these methods focus on the small representation part of a particular object common between different instances of the same class across the dataset. To deal with these issues, different bottom-up techniques have been proposed to improve the quality of activation maps. Below, we provided a brief description of activations-based methods, seminal methods, and other methods to deal with the issue that is inherited to these methods, which are categorized into spatial pooling and CAM refinement methods.

Spatial pooling based methods. In a typical WSOL setting, a classification network is trained, and this network can be employed to obtain activation highlighting the concerned object responsible for classification. These activation are typically known as features map $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$, where C defines the number of activation channels with spatial dimension of $H \times W$. Different channels in a features map normally encompass different object parts corresponding to different classes, and averaging them leads to a noisy map. To deal with

this issue, we need weights for each channel C to highlight relevant object parts and suppress background noise. To obtain these weights, a polling method is required to consolidate the information within an activation map, which could be utilized to map them to a weight that is then used to classify an object; Global average pooling (GAP), is a seminal pooling method that computes a global representation of spatial responses by averaging the activations across each feature map. For a feature map $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$, where C is the number of channels and H and W are the spatial dimensions, the GAP operation produces a channel-wise average $\mathbf{m} \in \mathbb{R}^C$,

$$m_c = \frac{\sum_{j=1}^H \sum_{k=1}^W M_{c,j,k}}{H \cdot W},$$

where $M_{c,j,k}$ denotes the activation at spatial location (j, k) within the c -th channel. These m_c contain a compact global representation of spatial information for each channel that represents the weight of every channel (class) and its contributions towards the prediction of each class. It also helps us to produce a heat map that weighs each pixel according to their contribution towards the decision. As shown in Fig.1.4, the class-specific weights of a layer associated with GAP are multiplied with the last convolution layer's feature map to generate an activation map. Mathematically,

$$CAM = \sum_k m_k^c \cdot M_k^{i,j} \quad (1.4)$$

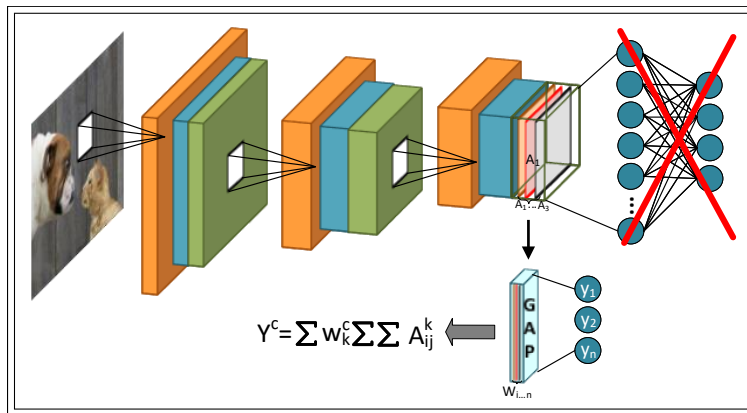


Figure 1.4 Illustration of CAM technique
Adapted from Zhou *et al.* (2016)

However, localization maps produced by CAM always focus on the most discriminative areas as they are trained with classification loss that always tries to minimize the mutual information between different instances of the same class and the underlying class. Different methods have been proposed to improve localization accuracy, including the development of new pooling, some of them specially designed for WSOL and others adapted from other tasks. Although this strategy allows weak supervision signals to “spread” across the input, it also blends background features into the final class score, sometimes fragmenting the true object region and ignoring certain parts of the target object (Zhou *et al.*, 2016).

MAX-Pool can potentially replace GAP. However, it activates attention only over a few pixels because of its reliance on the maximum response that drives the receptive field of the network towards the small discriminative object parts, as only a single pixel on the response map contributes to the per-class score. To deal with this issue, (Pinheiro & Collobert, 2015; Sun, Paluri, Collobert, Nevatia & Bourdev, 2016) proposes to employ a smoothed approximation of the maximum function that encourages the discovery of more extensive regions of interest. In particular, these methods employ the *log-sum-exp* (LSE) function. Similarly (Oquab, Bottou, Laptev & Sivic, 2015) improve this further instead of selecting a single maximum element in the response map, *local maxima* pooling has been proposed in (Zhou, Zhu, Ye, Qiu & Jiao, 2018) where multiple spatial maxima are considered to build localization map to encompasses different object parts.

While above discussed pooling-based techniques rely on high activation values to determine foreground regions, they often fail to capture the full spatial extent of the target object, particularly when visually similar object parts appear across multiple classes. To address this limitation, WILDCAT (Durand, Mordan, Thome & Cord, 2017) proposed to includes *negative evidence* while computing localization maps, low-activation areas—when computing per-class scores. The inclusion of low-activation responses serves a regularizing function, mitigating overfitting to more discriminative areas and covering different object parts. To include relevant negative evidence, WILDCAT leverages multiple modality maps for each class, yielding robust localization maps.

Inspiring from the pooling-based approaches, Deep MIL (Ilse, Tomczak & Welling, 2018) adopts an explicit MIL paradigm (Carbonneau, Cheplygina, Granger & Gagnon, 2018; Wang, Yan, Tang, Bai & Liu, 2018b; Zhou, 2004), wherein each image is divided into “instances” and their representations are aggregated into a “bag”. This aggregation employs attention for computing weights, thereby highlighting regions of interest. High attention weights signify potential foreground instances, whereas lower weights denote background regions.

In the following section, we provide a brief description of methods focused on refining CAMs, offering to improve CAMs performance.

CAM refinement methods. Despite significant progress in pooling methods for WSOL, one common issue with these methods is that localization maps tend to underestimate the spatial extent of the concerned object. Instead of just developing new pooling methods, the researchers shifted their focus away from tinkering with pooling methods. For instance, different methods are proposed specifically to rectify under-activation by recovering the full object extent, which is categorized into data augmentation methods to harvest cues beyond discriminative ones and methods to improve the internal activation of the network. Further details regarding data augmentation methods to improve localization performance are presented below.

Data augmentation methods. Augmenting input images is a common practice in machine learning to mitigate overfitting and generalization of networks beyond training examples (Goodfellow, Bengio, Courville & Bengio, 2016). As data augmentation improves network performance without requiring new training samples, it is widely adopted in WSOL to prevent models from anchoring their focus on a discriminative region, producing an activation map covering different object parts. The predominant data augmentation is the “erasing” strategy, wherein different parts of an input image or intermediate features map are deliberately removed. Theoretically, such perturbation forces the network to look beyond discriminative regions to highlight all object parts for an object of interest. For instance, hide-and-seek (HaS) (Singh & Lee, 2017), whereby the input image is divided into different patches. During learning, a subset of patches are randomly erased at each training step. Hiding patches forces the network to rely on less

discriminative regions for prediction and improve object discovery to produce regions of interest (ROI).

Another category of augmentation-based methods focuses on blending the image with external information. For instance, MixUp (Zhang, 2017) blends two images and assigns labels that are linear combinations of underlying images for regularized models by decreasing sensitivity to adversarial examples. CutMix (Yun *et al.*, 2019), built on top of MixUp, proposed randomly removing patches from an image, replacing them with corresponding patches from another image, and using global labels proportional to the size of the replaced region. HaS and CutMix fill the erased areas with image content from different sources, leading to mismatched label information. To address this issue, PuzzleMix (Kim, Choo & Song, 2020) proposed to optimize the input image jointly and its corresponding mask for mixing the most salient image parts. Similarly, SaliencyMix (Uddin, Monira, Shin, Chung & Bae, 2021) leverages image saliency to determine optimal region remixing.

A parallel line of work focuses on iterative erasure and mining object parts to optimize the learning process. For instance, AE (Wei, Feng, Liang & et. al, 2017) erase image parts whose corresponding pixels are highly activated in the activation maps. However, this becomes very computationally expensive. To deal with this issue, ACoL (Zhang, Wei, Feng, Yang & Huang, 2018c) employs two classifier branches to predict discriminative and complementary regions concurrently to improve localization performance. Similarly, MEIL (Mai, Yang & Luo, 2020) employs multiple output branches, integrating the erasure process directly into learning. GAIN (Li, Wu, Peng *et al.*, 2018) employs two parallel networks that share their backbone. The first network highlights discriminative ROIs, which are then erased, and the second network ensures that no significant discriminative area is left. This ensures that even less discriminative areas are included in the localization maps.

The above-discussed methods alter the input image. However, an attention-based dropout layer (ADL) (Choe, Lee & Shim, 2021b) was proposed to build an attention map in different layers to spot potential ROIs. For this, ADL employs self-attention to derive a more robust localization

map. ADL aims to hide the most discriminative parts of the target object from the classifier head. Also, simultaneously enlarging the network’s receptive field beyond these highly discriminative areas enables the model to utilize different object parts for decision. The ADL constructs a self-attention map by performing channel-wise pooling on the feature maps and subsequently derives a drop mask and an importance map via thresholding and sigmoid activation. These two maps are then fused to produce a robust localization map.

Similarly, MAXMIN (Belharbi *et al.*, 2022b) proposes to employ a localizer-classifier pipeline where the output of the localizer is used to mask the input image before feeding it to the classifier. This enforces the inclusion of object parts in the localization map by penalizing or rewarding specific activations. By using the entropy of the classifier output to be high in the foreground and low in the background, MAXMIN ensures that fewer ROIs are erroneously omitted.

Recently, LocLoc (Cao, Zheng, Shen & et. al, 2023) proposed to improve classification and localization performance by incorporating low-level image cues. It consists of a GrabCut-enhanced generator (GEG) and a local feature digging module (LFDM). The GEG leverages the GrabCut algorithm to refine object localization by highlighting the foreground and background based on colour and texture information. GrabCut enables the model to produce a refined mask with sharp boundaries. Furthermore, LFDM improves classification performance by focusing on low-level cues.

Features enhancement methods. Feature enhancement methods aim to learn robust features essential for localization tasks. To enable robust feature learning, different authors propose new architectures or the fusion of multi-level features to improve localization performance. Additionally, utilizing pseudo-labels for model training that are often obtained from model activations plays an important role in network training. A prominent method in this domain, revisiting dilated convolution (Wei *et al.*, 2018), proposes to employ different classification heads containing CNN layers with varying dilation rates. Employing different heads for classification increases the network’s receptive field, enabling it to explore object parts, even if they are less discriminative, producing a robust activation map (see Fig.1.5).

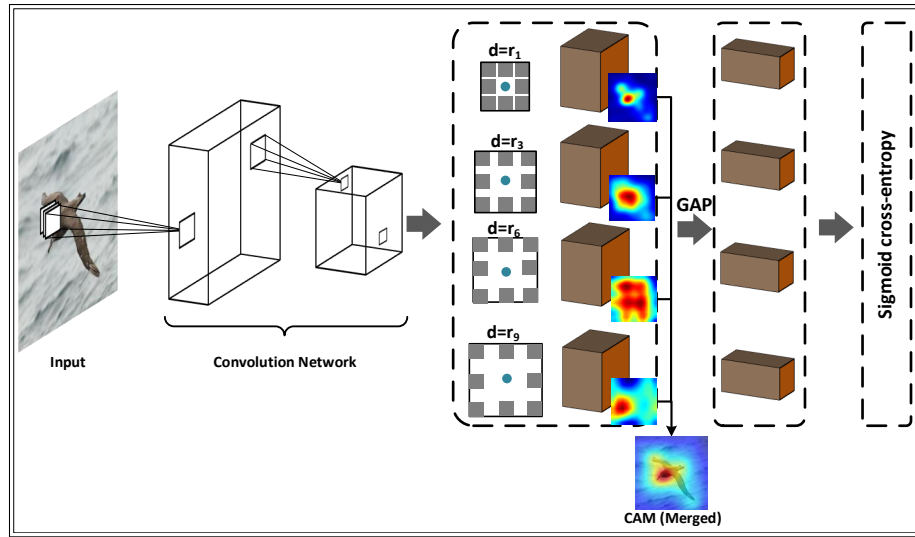


Figure 1.5 Illustration of multi-dilated network for improving CAM
Adapted from Wei *et al.* (2018)

Similarly, NL-CCAM (Yang, Kim, Kim & Kim, 2020) fuses class activation maps obtained from the few highest-performing and few lower-ranked classes to highlight salient foreground regions while suppressing background activations. Fusing maps from different classes enables highlighting different object parts and suppresses background noise, thereby producing robust localization maps. FickleNet (Lee, Kim, Lee & et. al, 2019) proposes to improve localization performance by including less discriminative regions in the localization maps. It randomly zeroes out subsets of feature map locations to generate multiple CAMs for each input.

To further improve WSOL performance and enable the network to discover all object parts, irrespective of their discriminative nature, DANet (Xue, Liu, Wan & et. al, 2019) introduces multiple network branches that produce localization maps at varying resolutions. DANet also introduces discrepant divergent activation loss, which enables the network to focus on all object parts, even when overlapping with other classes, without affecting classification accuracy. Instead of employing multiple branches to discover less discriminative areas, I²C (Zhang, Wei & Yang, 2020e) improves localization performance by leveraging inter-image features and ensuring consistency between local and global discriminative cues. This method employs local and

global constraints to enhance the consistency among discriminative pixels while maintaining a per-class global feature. Moreover, to suppress background, ICL (Minsong, Youngjung, Wonyoung & Hyeran, 2020) introduces attention-based contrastive loss that identifies foreground and background regions based on high and low activation values, while values in between remain uncertain. Using this information, the model is able to push background and foreground features far away to produce robust localization maps.

Vision transformers (ViTs) and their variants, originally proposed for classification tasks, have also been adapted for WSOL tasks. In this context, TS-CAM emerged as a pioneering method that adapted ViTs for WSOL, leveraging their ability to capture long-range dependencies among different parts of an image. This enables the model to connect different object parts even if they are not discriminative. For this, TS-CAM (Gao *et al.*, 2021) reallocates the dependency of classification head from $[cls]$ token to patch tokens $t_* = \{t_1^{1 \times d}, t_2^{1 \times d}, \dots, t_3^{1 \times d}\}$. These patch tokens, denoted as $t_*^{N,d}$ are then concatenated and reshaped to $t'_* = t_*^{N,(W/P),(H/P)}$. Finally, convolution filters k are applied to t'_* to yield final map. To further improve the robustness of maps, the average of all $[cls]$ tokens $[cls]'$ is multiplied with class-aware map t'_* as $\sum_{i=1}^d t'_* * k_{c,d}$.

As discussed above, it first divides the images into patches to capture long-range dependency information, subsequently registering its effects in the $[cls]$ token. The attention of $[cls]$ token is then fused with the semantic aware map to produce the final attention/activation map. A diagram illustrating the TS-CAM is presented in Fig.1.6.

Building on this transformer-based paradigm, different methods (Chen, Wang, Wang & et. al, 2022; Li, 2022; Bai, Zhang, Wang & Wan, 2022; Su, Ye, Chen, Song & Cheng, 2022) have been proposed to improve the localization capability of transformers. For instance, (Chen *et al.*, 2022) proposes to tackle the lack of inherent local inductive bias by improving local continuity for discovering the full object. Li proposed CaFT for clustering and filtering of tokens to eliminate the need for the hand-crafted threshold to separate out foreground and background features. (Bai *et al.*, 2022) employs a spatial calibration module (SCM) to adjust attention scores for improving localization maps by incorporating semantic similarity and spatial context. After training, SCM

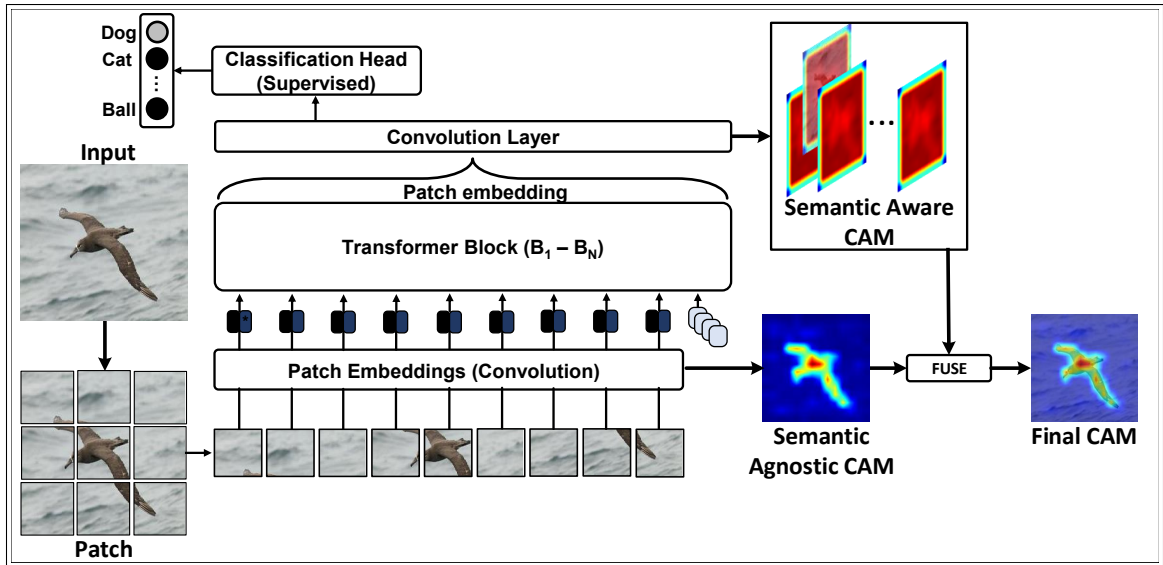


Figure 1.6 Illustration of TS-CAM: token semantic coupled attention map for WSOL
Adapted from Gao *et al.* (2021)

can be removed without impacting the localization performance. (Su *et al.*, 2022) extended this by emphasizing on iterative refinement strategy: they proposed iteratively re-weighting attention maps through auxiliary supervision to expand the activation propagation towards all object parts. More recently, (Wu, Zhai, Cao, Luo & Zha, 2023) aims to minimize optimization conflict between localization and classification tasks by introducing spatial-aware tokens (SAT) by aggregating representations.

Recently, different methods have been proposed to improve the localization capability of deep neural networks by highlighting different object parts. For instance, B-CAM (Wu, Zhai & Cao, 2022) and BAS (Zhu *et al.*, 2021) suppress over-activation of object-regions by learning background cues and activation-based penalty. In contrast, CREAM (Xu, Hou, Zhang & et. al, 2022) and FDA (Kim *et al.*, 2022) tackle the challenge of partial activation over object parts by clustering and aligning feature vectors to include all foreground regions in the localization map. Generative prompt model (GenPrompt) (Zhao *et al.*, 2023) introduces a conditional image-denoising pipeline, recovering the entire object extent via representative embeddings guided by discriminative queries.

Beyond these methods, a series of approaches based on feature enhancement of convolution networks have drawn significant attention. For instance, IDC (Wang, Xu, Xu & et. al, 2024) exploits intrinsic discrimination by learning foreground–background regions using metric learning by enforcing intrinsic consistency. It enforces pixel and object-wise consistency to improve localization accuracy. Likewise, MDAL (Zhang, Ling & Wang, 2023) proposes a multi-layer decoupling attention mechanism to deal with the issue of incomplete activation and inaccurate boundaries. It fuses shallow and deep features by isolating (i.e., decoupling) their inter-dependencies, thus mitigating background noise or over-activations. Alternatively, DIR (Lee, Mun, Uh & et. al, 2024) highlights the importance of refining the inference pipeline: it binarizes the penultimate feature map and its corresponding classification weights by introducing two hyperparameters. Substantial improvements in localization performance across different backbones are yielded without re-training the network, highlighting the need for a robust inference process. Similarly, CAM-BA (Yasuki & Taki, 2024) improves the localization performance of conventional CAMs by utilizing CNNs with large kernels for extracting activation maps. Finally, CIAT (Su & Yang, 2024) proposes to exploit semantic relationships between shallow and deep features paired with an online adaptive threshold (OAT). OAT separately estimates a segmentation threshold for each input image to produce accurate bboxes.

Pseudo-annotation: A recent trend in WSOL is to harvest pseudo-labels to train localization models in an end-to-end fashion. These pseudo-labels are often extracted from a pre-trained classifier. While this approach allows us to train the localization head directly, it inherently presents the challenge of training the model with noisy labels as they are obtained from another model pre-trained for the classification task. In Pair-Sim framework (Rahimi, Shaban, Ajanthan & et. al, 2020), a fully supervised Faster-RCNN (Ren, He, Girshick & Sun, 2015) was trained on a source dataset and then used to generate proposals on a target weakly supervised dataset.

In WSOL, optimizing a network for both classification and localization tasks proves to be challenging. The author in (Choe *et al.*, 2020) proposes to train separate networks for these tasks. Following this, (Wei, Wang, Li & et. al, 2021) proposes using low-level features to

generate pseudo labels, which are used to guide the training of another network. Similarly, F-CAM (Belharbi *et al.*, 2022c) proposed a method to train U-Net-like architecture (Ronneberger, Fischer & Brox, 2015) for producing high-resolution activation maps, and to train this model, pseudo-labels are obtained from a pre-trained classifier. They further introduce a *stochastic* sampling strategy for selecting local evidence during training, in contrast to the fixed pseudo-labels. In another line of work, the self-produced guidance (SPG) (Zhang, Wei, Kang & et. al, 2018d) utilizes top- and low-level attention maps to produce fine-grained cues for more accurate localization. DiPS (Murtaza, Belharbi, Pedersoli, Sarraf & Granger, 2023b) utilizes a self-supervisory signal from DINO to train a transformer for robust localization that enables the model to produce localization maps, maintaining consistent performance across a wide range of thresholds. Similarly, ATNet (Meng, Zhang, Zhang, Zhang & Wu, 2022c) train transformer for object localization in an adversarial fashion. ATNet has two main components: an object transformer for producing pseudo-labels and localization maps. a part transformer for discrimination between different maps. Wei, Wang, Zhou & et. al proposes ISIC to address the tendency of WSOL to highlight only the most discriminative object parts. For this, they proposed an inter-class to highlight less-discriminative features shared among different categories, thereby improving the completeness of predicted object regions. Also, instead of relying on class-activation maps (CAM), the authors use a non-negative matrix factorization to utilize features of different images belonging to the same class to yield more robust localization. Similarly, part-based discovery model (PDM) (Meng, Zhang, Yang & et. al, 2022a) proposed to learn different discriminative object parts in one unified model. It incorporates three learning mechanisms: *diversity*, *compactness*, and *importance*, enabling the models to harvest features that are relevant for localization and classification tasks. This approach produces more robust localization and classification by integrating object parts within a unified network.

More recently, TAFormer (Meng, Zhang, Zhang & et. al, 2022b) proposes a transformer-based encoder-decoder for producing class-agnostic localization maps and classification scores in a task-aware manner. TAFormer introduces an optimal transport algorithm for producing pseudo-labels for refining foreground maps, improving both object localization accuracy and robustness

against adversarial attacks. SPA²Net (Chen, Pan, Tang & et. al, 2023a) proposes high-order self-correlation for structure-preserving in localization maps to improve the performance of the localization network. The authors proposed to employ separate branches for localization tasks and classification as they argue that these tasks are irrelevant. Xu, Luo, Hu & et. al proposes a binary-class detector (BCD) approach that is able to clearly separate foreground regions from background ones by employing a weighted entropy loss. Meanwhile, Local consistency aware re-prediction (LCAR) (Pan, Yao, Cao & et. al, 2023) refines object masks by aggregating visual cues from spatially incoherent activation, thereby improving predicted bounding boxes.

1.4.5 Top-down Techniques

Top-down WSOL methods rely on the backward pass information within a neural network to produce localization maps. These methods are further categorized into three subcategories, each differing in how the backward pass is computed: the first group of methods is biologically inspired, the second employs gradient information to aggregate spatial features from the backbone, and the third exploits posterior class scores for aggregation.

Biologically inspired methods. Approaches classified in this category are built upon cognitive science principles. Inspiring from cognitive science, Cao, Liu, Yang & et. al (2015) propose a feedback network attached to standard feedforward layers, with binary variables that are activated by top-down signals. To maximize the posterior score over the target, the model employs a forward pass followed by a backward pass that enforces neuron selectivity via an L_1 term. This optimization step is performed over multiple passes and jointly optimizes the target class. However, the iterative nature of the process introduces challenges for WSOL. A similar paradigm was adopted by excitation-backprop (Zhang *et al.*, 2018a) that employs winner-take-all (WTA) processes that utilize top-down and bottom-up activation by using Neuraon’s winning path. Moreover, a contrastive top-down method was proposed to improve localization by focusing on one target at a time (Zhang *et al.*, 2018a; Cao *et al.*, 2015).

Grad-based aggregation. Grad-based methods compute the gradient of the posterior class scores (for the true label) with respect to the target layer’s feature maps to compute channel-wise weights for the activation maps of a particular layer. These methods were originally proposed to explain a network’s decision. In (Selvaraju *et al.*, 2017), the authors introduce the *Grad-CAM* method, which—unlike traditional CAM—is capable of harvesting feature maps from any convolutional layer rather than being limited to the last block. By employing the global average of gradients, Grad-CAM highlights significant image regions, where the magnitude of these gradients reflects the rate of change most relevant to the target class.

For producing localization maps, Grad-CAM first computes a set of coefficients (weights) by taking the gradient of the network output, y^c , associated with the selected class c and averaging these gradients spatially. The resulting activation map for class c is then computed by aggregating the feature maps A_k (from layer k) using the gradient-derived coefficients. Mathematically, replacing the weights in the standard CAM equation (Eq. 1.4) with the gradient-based weights leads to:

$$\sum_k A_k \cdot \overbrace{\frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}}^{\text{global average}} \quad (1.5)$$

gradients

Finally, a rectified linear unit (ReLU) is employed to filter the positively correlated pixel with the targeted class.

Localization maps generated by CAM often fail to capture all object parts. To overcome these issues, Grad-CAM++ (Chattopadhyay, Sarkar, Howlader & Balasubramanian, 2018) is proposed for producing coarse-grained localization maps. Unlike Grad-CAM, it weights all pixels of a gradient feature map differently. The weights can be recalculated by taking partial derivative

w.r.t. A_k and rearranging the Equation 1.5:

$$\underbrace{\sum_k A_k}_{\text{feature map}} \cdot \underbrace{\sum_i \sum_j \left[\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2 \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}} \right]}_{\text{Recalculating weights}} \cdot \text{relu}\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right) \quad (1.6)$$

Furthermore, to enhance the fidelity of activation maps, two key axioms—*sensitivity* and *conservation* (Fu *et al.*, 2020). Based on these axioms, XGrad-CAM was proposed to compute robust coefficients for each activation map to produce localization maps. Most existing CAM-based methods extract feature maps from the last convolutional layer to produce localization maps. By contrast, LayerCAM (Jiang, Zhang, Hou, Cheng & Wei, 2021) employs both high and lower-level features to refine the localization map. Higher layers typically encode coarser spatial cues, whereas lower layers provide finer-grained yet noisier localization signals. LayerCAM extracts Grad-CAM from each layer and fuses these layer-wise CAMs into a localization map.

ViTOL (Gupta, Lakhotia, Rawat & Tallamraju, 2022) employed gradient attention rollout (GAR) to compute activation maps covering different object parts while mitigating the background noise. Moreover, BagCAM (Zhu, Chen, Jin & et. al, 2022a) and DGL (Tan, Gu, Ruan & et. al, 2022) proposed to exploit network gradients for improving localization performance. BagCAM introduces regional localizers that enable the generation of activation maps by computing gradients, allowing us to combine regional cues to enhance object localization beyond discriminative regions without retraining the base model. For instance, DGL employs gradients from different convolution layers, proposing pixel-level class selection and class-aware enhancement strategies. It also incorporates skip-layer connections to produce localization amps covering different object parts.

Confidence-based aggregation. CAM relies on network activations to compute the coefficients for feature maps, resulting in an underestimation of object parts. Different methods aim to compute the coefficient using the confidence of the classifier. One prominent example is

Score-CAM (Wang *et al.*, 2020a), which element-wise multiplies each activation map A_k from layer l with the input image, then feeds the masked image into the classifier C_θ to measure the target-class score. This score serves as the weight $w_k^c = C(A_k)$, reflecting the Channel-wise Increase of Confidence (CIC). The final CAM is computed as,

$$L_{\text{Score-CAM}}^c = \text{ReLU}\left(\sum_k w_k^c A_k\right), \quad (1.7)$$

where ReLU retains only positive contributions.

Building upon Score-CAM, the SS-CAM (Wang, Naidu, Michael & Kundu, 2020b) introduces smoothing to produce sharper activation maps. In particular, it averages the classifier's response over N noisy versions of the masked image (or masked activation maps), thus yielding a more stable and robust weight:

$$w_k = \frac{1}{N} \sum_{i=1}^N C(A_k + \mathcal{N}(0, \sigma)) \quad \text{or} \quad w_k = \frac{1}{N} \sum_{i=1}^N C((\chi \times A_k) + \mathcal{N}(0, \sigma)), \quad (1.8)$$

depending on whether perturbations are applied to the activation map or the input itself.

Subsequently, IS-CAM (Naidu, Ghosh, Maurya & et. al, 2020) combines SS-CAM with IntegratedGrad (Sundararajan, Taly & Yan, 2017) to incrementally update a mask M via:

$$M_{i+1} \leftarrow M_i + \left((\chi \times A_k) \frac{i}{N}\right), \quad M_0 = 0, \quad (1.9)$$

thus accumulating contributions from zero to the full activation map in small steps.

In all the above methods, each activation map A_k undergoes a min-max normalization before being multiplied with the input:

$$\text{Normalize}(A_k) = \frac{A_k - \min(A_k)}{\max(A_k) - \min(A_k)}. \quad (1.10)$$

While Score-CAM, SS-CAM, and IS-CAM provide good localization and mitigate some limitations of gradient-based approaches, they are computationally expensive, which can hinder their scalability (Belharbi *et al.*, 2022c).

1.4.6 Critical Analysis

WSOL remains a challenging task due to the difficulty of jointly optimizing models for both classification and localization, as well as biased evaluation protocol. Furthermore, a review of different WSOL methods highlights these challenges,

Localization bias: Different WSOL methods, such as CAM, tend to focus on the discriminative parts of an object rather than highlighting all object parts. This issue is prevalent in WSOL methods as they are trained with image-class labels that enforce the model to minimize the mutual information between different instances of the same class. This can lead to degraded localization as the model continues to train for longer epochs.

Challenges of employing transformers in WSOL: Transformer-based models achieve promising results for different vision tasks as they can capture long-range dependencies among different object parts. However, WSOL faces challenges due to the small resolution of generated attention maps and its inability to produce class-aware attention/localization maps. Furthermore, token selection and localization consistency across different attention maps are ongoing challenges, particularly in complex scenes.

Unrealistic evaluation protocol: Utilizing ground truth (GT) bounding box (bboxes) annotations for model selection, hyperparameter tuning, and threshold estimation leads toward biased model selection. Utilizing GT boxes for model selection overestimates localization performance and hinders applicability in real-world scenarios.

Discrepancy in training and evaluation: Training models for WSOL tasks employ image-class label labels, while evaluation requires GT bboxes annotation. This discrepancy in training and evaluation can lead to suboptimal localization performance.

Complexity of recent methods: Recent methods in WSOL, like GenPrompt, aim to improve localization performance by highlighting all object parts but require complex models. These include intricate prompt learning and iterative conditional denoising processes, increasing computation demands.

Dependence on image-class labels during inference: Recent WSOL methods built atop of vision-language model, e.g., CLIP, require class-specific prompts during inference. Nonetheless, these methods require external classifiers to generate localization maps, introducing challenges in real-world scenarios.

CHAPTER 2

DIPS: DISCRIMINATIVE PSEUDO-LABEL SAMPLING WITH SELF-SUPERVISED TRANSFORMERS FOR WEAKLY SUPERVISED OBJECT LOCALIZATION

Shakeeb Murtaza^a , Soufiane Belharbi^a , Marco Pedersoli^a , Aydin Sarraf^b , Eric Granger^a

^aDepartment of Systems Engineering, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

^bEricsson, Global AI Accelerator,
8275 Trans Canada Route, Saint-Laurent, Québec H4S 0B6

Paper published in *Image and Vision Computing*, December 2023

Abstract Self-supervised vision transformers (SSTs) have shown great potential to yield rich localization maps that highlight different objects in an image. However, these maps remain class-agnostic since the model is unsupervised. They often tend to decompose the image into multiple maps containing different objects while being unable to distinguish the object of interest from background noise objects. In this paper, **Discriminative Pseudo-label Sampling (DiPS)** is introduced to leverage these class-agnostic maps for weakly-supervised object localization (WSOL), where only image-class labels are available. Given multiple attention maps, DiPS relies on a pre-trained classifier to identify the most discriminative regions of each attention map. This ensures that the selected ROIs cover the correct image object while discarding the background ones, and, as such, provides a rich pool of diverse and discriminative proposals to cover different parts of the object. Subsequently, these proposals are used as pseudo-labels to train our new transformer-based WSOL model designed to perform classification and localization tasks. Unlike standard WSOL methods, DiPS optimizes performance in both tasks by using a transformer encoder and a dedicated output head for each task, each trained using dedicated loss functions. To avoid overfitting a single proposal and promote better object coverage, a single proposal is randomly selected among the top ones for a training image at each training step. Experimental results on the challenging CUB, ILSVRC, OpenImages, and TelDrone datasets

indicate that our architecture, in combination with our transformer-based proposals, can yield better localization performance than state-of-the-art methods.

2.1 Introduction

The recent success of deep learning models (DL) in different visual recognition tasks, such as image classification (He *et al.*, 2016), object localization (Choe *et al.*, 2020), detection (Redmon, Divvala, Girshick *et al.*, 2016), and segmentation (Chen, Papandreou, Kokkinos & et. al, 2018b) requires big models, and most importantly, large, annotated datasets. The high cost of dense localization and of segmentation supervision makes it difficult to train these models and annotate large-scale datasets. Weakly supervised object localization (WSOL) has recently emerged as a surrogate training strategy to alleviate the need for bounding box supervision (Zhou *et al.*, 2016). This allows a cost-effective and fast collection of large datasets. Using only a global image label, *i.e.*, image class, a DL model can be trained to classify an image and localize objects of interest.

Class activation Mapping (CAM) methods are Convolution Neural Network (CNN)-based approaches that have been dominating the WSOL field (Belharbi, Murtaza, Pedersoli & et. al, 2023c; Belharbi *et al.*, 2022c; Choe & Shim, 2019; Lee *et al.*, 2019; Rahimi *et al.*, 2020; Singh & Lee, 2017; Wei *et al.*, 2021, 2017; Xue *et al.*, 2019; Yang *et al.*, 2020; Yun *et al.*, 2019; Zhang, Cao & Wu, 2020c; Zhang *et al.*, 2018d). They require only image-class supervision, and leverage spatial information in a CNN. Guided only by a discriminative loss, CAM methods can yield a spatial map per class to localize an object of interest while classifying the image. However, this allows the emergence of only the most discriminative part of an object (Choe *et al.*, 2020; Rony, Belharbi, Dolz & et. al, 2023). Often, it is limited to small repetitive patterns across samples of a class. Such limited coverage leads to poor localization since large parts of an object are missing (Fig.2.1c). Different techniques have been proposed to improve CAM-based localization (Rony *et al.*, 2023), including different spatial poolings, data augmentation, feature enhancement, and feedback methods. A recent prominent line of research aims to leverage pseudo-labels to fine-tune models (Zhang *et al.*, 2020c; Wei *et al.*, 2021; Belharbi, Pedersoli, Ayed, McCaffrey & Granger, 2022a; Belharbi *et al.*, 2022c; Belharbi,

Ayed, McCaffrey & Granger, 2023a; Belharbi *et al.*, 2023c). Various works have shown that classification and localization tasks are antagonistic in a WSOL setup (Zhang *et al.*, 2020c; Choe *et al.*, 2020; Rony *et al.*, 2023), where localization converges very early on, while classification converges late in the training. Typically, a classifier is trained until convergence and used to generate pseudo-labels. Then, it is frozen and equipped with a localization branch to be fine-tuned for localization. This allows to build a single model that yields the best performance over both tasks. Despite the success of pseudo-labeling methods, the performance obtained is strongly tied to the emerging localization, *i.e.*, ROIs, in CAMs, which tend to be *local*, and limited to a small part of an object. This makes the pseudo-labels less efficient.

Recently, self-supervised transformers (SSTs) (Caron *et al.*, 2021) have come to constitute a strong competitor to CNN-based models. Without any supervision, they can produce good attention maps for different objects in an image (Fig.2.1a). With their long-range dependency and self-supervised training, SST models can decompose different objects into multiple maps. However, they are unable to discriminate between different objects linked to a particular class. Most importantly, though, they can accumulate the *localization* information of different objects into multiple attention maps (tokens). These maps highlight all objects in a scene as they are trained without any class information. Each token focuses on different objects that are semantically distinct from one another. Although this provides a rich source for object localization, these attention maps are not associated with any particular class, making them less useful for WSOL tasks.

Various works have attempted to leverage transformers for WSOL (Bai *et al.*, 2022; Chen *et al.*, 2022; Gao *et al.*, 2021; Gupta *et al.*, 2022; Su *et al.*, 2022) by using the image class as supervision. The TS-CAM method (Gao *et al.*, 2021) is one of the prominent transformer-based WSOL approaches. It performs a simple fusion of all attention maps across all layers to build a single class-agnostic map. This map is then aggregated with a CAM for final localization. Such cross-layer aggregation of maps introduces localization noise (Fig.2.1b), which hampers performance. In (Bai *et al.*, 2022; Gupta *et al.*, 2022), the authors propose to suppress background noise and calibrate foreground activations. Despite the success of these methods, they still

need complex changes to deal with background noise. Building class-aware attention maps in transformer-based models is still an ongoing challenge.

In this work, we aim to design a method to exploit the rich localization information built into self-supervised transformers. Unlike TS-CAM (Gao *et al.*, 2021), which fuses attention maps, we propose to use a pre-trained classifier to identify the most *discriminative ROIs*, with respect to the true image class, across different attention maps (Fig.2.2). This allows us to gather more

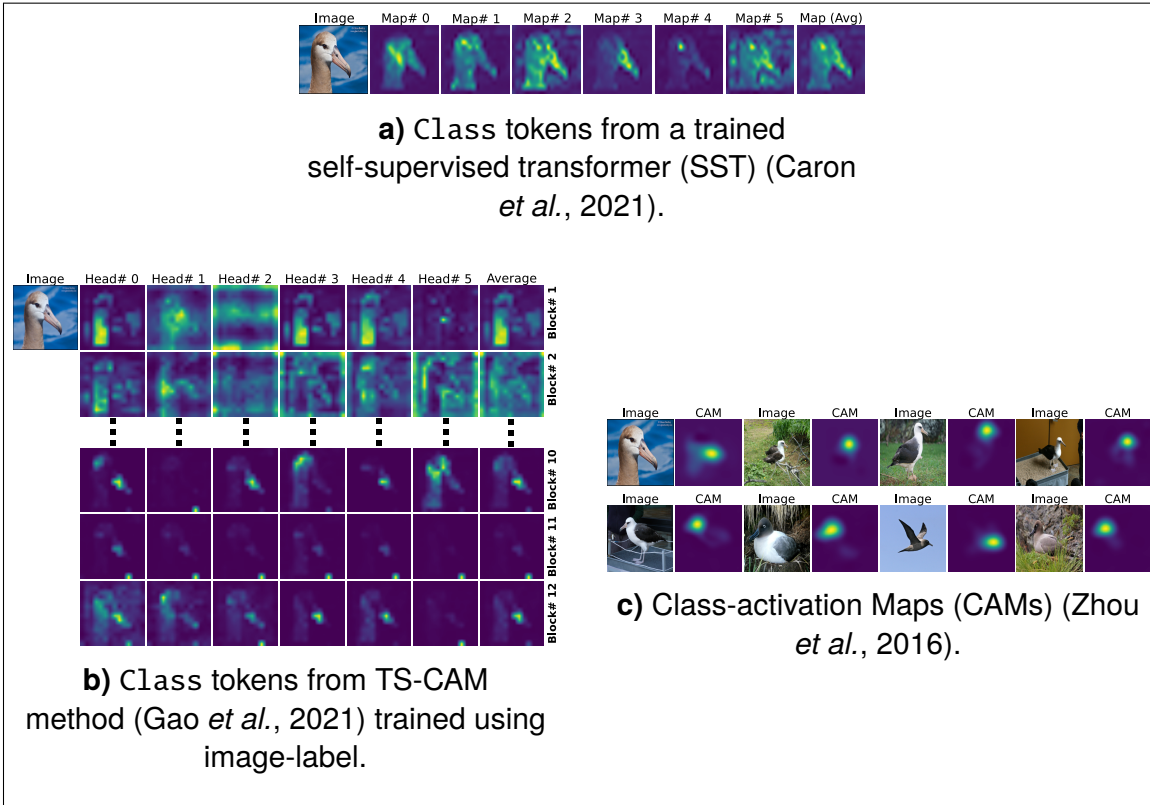


Figure 2.1 Comparison of activation maps of three different models: **(a)**: The last layer of the SST attends to all parts of foreground regions. **(b)**: Visualizations of class tokens collected from different attention layers of TS-CAM (Gao *et al.*, 2021). Earlier attention layers attend to background regions instead of focusing exclusively on foreground regions. Fusing all these maps, as in TS-CAM, introduces noisy localization. **(c)**: CAMs collected from CNN-based model (Oquab *et al.*, 2015) show that they are very local and only focus on a small discriminative area since correctly classifying the image is deemed to be enough.

Note that large parts of the object are missing, compared to SST maps. Naturally, this positions SST maps as a better choice for sampling good-quality pseudo-labels

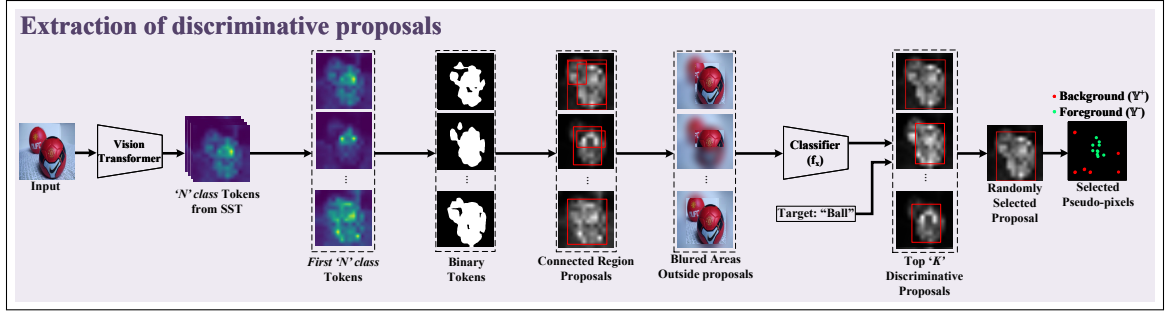


Figure 2.2 Our approach for generating *discriminative* proposals from class tokens of SSTs. First, the localization map are automatically binarized (Otsu, 1979) to identify regions of interest that may contain a specific object. From each binarized map, all connected areas are extracted along with their tight bounding boxes. This creates a pool of bounding boxes from different attention maps. Each bounding box is scored using an external classifier f_x . We use the posterior probability of the true class y of the image to measure the likelihood of the bounding box containing the true object. Outside of the box is suppressed by blurring operations. The top- K boxes are selected. At each training step, we randomly select a proposal box. Inside the box, we randomly select foreground pixels. While background pixels are selected from outside the box. A detailed flow diagram is displayed in Fig.2.3

reliable and diverse proposals that better cover the object class in the image. Furthermore, it ensures that background noise objects, which typically emerge in such maps, are discarded. We adopt a fine-tuning approach for WSOL to build a model that achieves the best results in both the classification and localization tasks. More specifically, we leverage self-supervised transformers by using them as a backbone and employ the aforementioned proposals to train them. The localized discriminative proposals from an SST are used to sample pixel-wise pseudo-labels, which are then used to train our model for localization. These pseudo-labels, as elaborated in our preliminary study (Murtaza, Belharbi, Pedersoli, Sarraf & Granger, 2023a), are important in the training of the localization model. Moreover, our final model is composed of a transformer-based encoder equipped with two output heads, namely, a classification head and a localization head. Both are trained separately for better performance.

Our main contributions are summarized as follows:

(1) A novel method called Discriminative Pseudo-Label Sampling (DiPS) is introduced to leverage the rich localization information contained in a set of *class-agnostic* attention maps of SSTs (see Fig.2.2). Using a pre-trained classifier, DiPS collects the most discriminative ROIs with respect to the image class, while discarding the background noise objects. Given the multiple attention maps produced by transformers and different ROIs, our method provides a rich pool of diverse and discriminative proposals to cover different parts of the object. DiPS allows for the production of reliable pseudo-labels with better object coverage during localization. This differs from standard CAMs, which are constrained to highlighting smaller object regions by minimizing the mutual information between class instances.

(2) Our new DiPS method is trained to simultaneously provide a high level of classification and localization accuracy. It is composed of a transformer-based encoder, a classification head, and a localization head. The encoder is pre-trained under an SST model, which is frozen. The classification head is trained to yield the best classification accuracy and then is frozen as well. Moreover, the localization head is trained using the discriminative pseudo-labels harvested from SST. To avoid overfitting a single proposal and promote better object coverage, DiPS randomly selects a single proposal among the top ones for a training image at each training step. Additionally, for a better object-boundary delimitation, it uses a CRF loss (Tang, Perazzi, Djelouah & et. al, 2018). Moreover, *DiPS* only requires a single forward pass through the encoder and respective heads to concurrently accomplish classification and localization tasks.

(3) Our extensive results compare DiPS with state-of-the-art WSOL methods on four datasets – three public common benchmarks for the WSOL task (ILSVRC, CUB-200-2011, OpenImages), and a fourth proprietary dataset, TelDrone, designed for cell tower inspection using drones. Our DiPS method outperforms recent WSOL methods and achieves a new state-of-the-art localization performance. Further analysis and ablations are provided for our method, along with our public code.

2.2 Related Work

This section provides a brief summary of WSOL methods built on top of CNNs and vision transformers (ViTs).

Convolutional Neural Network: State-of-the-art WSOL methods are designed to collect the localization map from the last convolution layer of CNNs (Zhou *et al.*, 2016). These maps can be generated by aggregating the activation maps from the penultimate layer based on the contribution of each map toward the final prediction. Different methods have been proposed to improve the map extraction mechanism for WSOL (Chattopadhyay *et al.*, 2018; Fu *et al.*, 2020; Ramaswamy *et al.*, 2020; Selvaraju *et al.*, 2017). These methods always focus on discriminative regions as they are optimized using class-level labels. This allows the network to find the common object parts between instances of the same class. To address this limitation, different methods have been proposed to expand the receptive field beyond discriminative regions. Among them, the most common one used for enlarging the activation map beyond discriminative regions proceeds by removing them either by an adversarial perturbation (Singh & Lee, 2017; Yun *et al.*, 2019) or by employing an adversarial loss (Choe & Shim, 2019; Zhang *et al.*, 2018c). On the other hand, instead of carrying out weighted averaging of all the activation maps of a specific layer, the proposed fusion-based methods combine the activation maps according to their importance as determined by the classifier’s score (Naidu *et al.*, 2020; Wang *et al.*, 2020b,a).

Instead of using post-hoc techniques for collecting the localization maps from pre-trained networks, different architectures have been designed for WSOL that are able to generate localization maps directly instead of fusing activation maps (Lee *et al.*, 2019; Rahimi *et al.*, 2020; Xue *et al.*, 2019; Yang *et al.*, 2020; Zhang *et al.*, 2020c). For instance, (Wei *et al.*, 2018) replaces the penultimate convolution layer with a layer having multiple parallel convolution filters with different dilation rates. To generate the localization maps, standard CAMs are added to the average of different CAMs generated from dilated convolution layers. The optimization of this network is extremely difficult as the classification loss must be minimized by using a separate classification for each convolution layer in the penultimate layers. For instance, (Yang

et al., 2020) proposes combining CAMs of different classes from highest to lowest, based on class probability scores. This method is also cost-intensive as CAMs must be averaged from different classes. To overcome this limitation, (Wu *et al.*, 2022; Zhu *et al.*, 2021) proposes a method to suppress the background regions to help the network identify foreground regions with high confidence.

In (Meng, Zhang, Tian & *et. al*, 2021), the author introduces object-aware and part-aware attention modules to jointly optimize localization and classification accuracy. This model first obtains the whole object and then decomposes it into parts for classification. It is capable of producing efficient localization maps while retaining the same classification performance. Additionally, (Xie, Luo, Zhu & *et. al*, 2021) added a sub-network between different layers of the network to regularize its internal features. The sub-network is an encoder-decoder with a classification head serving to preserve the details of objects at different levels in the network. In (Zhang *et al.*, 2018d), the author proposes self-produced guidance (SPG) for expanding the foreground map beyond the discriminative regions. SPG first collects seeds from the attention map, which are then expanded to areas having high activation values to produce the foreground map. Inter-image communication (I^2C) (Zhang *et al.*, 2020e) was introduced to expand the activation maps to cover the whole object. Nonetheless, this approach adversely affects classification performance. To address this issue, (Pan, Gao, Lin & *et. al*, 2021) proposed a model with two modules, namely, an activation-restricted module (RAM) to discover the parts of the objects by using a classification network and a self-correlation map-generating (SCG) module for producing the final map. Shallow feature-aware pseudo-supervised object localization (SPOL) produces pseudo-labels with high confidence for background and foreground regions (Wei *et al.*, 2021). However, these pseudo-labels remain fixed for different iterations, which could generate noisy maps. Hence, (Belharbi *et al.*, 2022c; Murtaza *et al.*, 2023a) propose using effective pseudo-labels that will be sampled probabilistically at each step, allowing the network to discover related parts of the image to improve the localization accuracy.

The aforementioned methods rely on the activation maps of CNNs trained using classification loss, which limits their receptive field. Inductive bias forces the CNNs to decompose the object

into local semantic parts (Bau, Zhou, Khosla & et. al, 2017; Zeiler & Fergus, 2014), preventing them from forming global relationships between object parts presented in different receptive fields. This hinders the ability of CNNs (trained using a classification loss) to detect all the object parts, which thus leads to blobby maps. Furthermore, several approaches have been proposed to extend the localization map beyond merely discriminative regions. These methods harness pixel similarities to allow the network to identify various parts of an object (Wang, You, Li & Ma, 2018c; Wang, Zhang, Kan & et. al, 2020c; Zhang *et al.*, 2020c,e). Concurrently, (Minsong *et al.*, 2020) proposed to leverage long-range dependencies in CNNs to capture spatial similarities, thereby enhancing object coverage.

Vision Transformer (ViT) Methods: In contrast to CNNs, transformer networks have the ability to capture long-range dependencies due to their intrinsic properties, which help generate efficient localization maps. These models rely heavily on attention mechanisms by computing the dot product between key-value pairs at different levels. Recently, the transformers were effectively employed for computer vision tasks, achieving great success. Specifically, various transformer-based methods for WSOL have been proposed (Chen *et al.*, 2022; Li, 2022; Gupta *et al.*, 2022; Bai *et al.*, 2022; Su *et al.*, 2022; Meng *et al.*, 2022c). TS-CAM is another emerging work for WSOL using a transformer (Gao *et al.*, 2021). It employs a classification head on top of the `class` tokens to train a transformer using class labels and then extracts a semantic-aware map from it. The map is then multiplied by the average of all `class` tokens from different layers. This accumulation introduces background noise, which hinders the performance of this method. Moreover, TS-CAM focuses solely on capturing long-range dependencies while ignoring the inductive locality bias, leading to unreliable maps. To deal with this, (Bai *et al.*, 2022) proposes a calibration mechanism for calibrating the network to produce relatively smooth activation values for different objects. This mechanism limits the receptive field of the transformer, producing more stable activation values across different parts of an object of interest. It also introduces the Spatial Calibration Module (SCM) to align the object boundaries with the edges of the localization map. Similarly, (Chen *et al.*, 2022) introduces a local continuity transformer (LCTR) for fusing local and global features to improve the perception quality of activation maps for

an object of interest. These features are fused by incorporating two modules into the network, namely, the relational patch-attention module (RPAM) and the cue-digging module (CDM). RPAM and CDM help in retaining the global features and highlighting the less discriminative object parts, respectively. Furthermore, to reduce the background noise in the localization map generated by transformers, (Gupta *et al.*, 2022) presents a novel dropout mechanism within a transformer block to limit the receptive field of the transformers. This method utilizes the attention roll-out method (Abnar & Zuidema, 2020) to produce the attention maps for a particular object, significantly mitigating the background noise. Similarly, (Su *et al.*, 2022) proposes a token refinement transformer (TRT) for WSOL to produce high-confidence localization maps. Here, the TRT employs a token priority scoring module (TPSM) to capture the precise object semantics by suppressing the background regions.

The aforementioned methods require an optimal threshold to draw bounding boxes around the object of interest, making them sensitive to threshold values. Also, most baseline methods do not minimize the loss over generated maps, which results in blobby and unreliable localization maps. Although some methods use pseudo-labels to train the localization module, they fix pseudo-labels and force the underlying model to generate maps close to the pseudo-label (Wei *et al.*, 2021). Similarly, (Belharbi *et al.*, 2022c) employed a probabilistic sampling for building pseudo-labels from standard CAMs, restricting their performance because CAMs always highlight mutual information common to different instances of a particular class. Moreover, transformer-based methods accumulate attention maps from different layers and fuse them with activation maps, introducing a background noise in the final map (Gao *et al.*, 2021). Comparatively, the proposed method collects a `class` token from the last layer of SST, which has been shown to hold rich localization information (Fig.2.1). Discriminative proposals using an external scoring classifier are collected. The top reliable proposals are used as pseudo-labels for the localization task. Such discriminative selection allows picking potential ROIs while discarding background noise. Random selection of these proposals for training furthermore prevents overfitting to a single region and promotes exploring different parts of an object.

2.3 The Proposed Discriminative Pseudo-label Sampling

2.3.1 Notation

We denote by $\mathbb{D} = \{(X, y)_i\}_{i=1}^N$ a training set, where $X_i : \Omega \subset \mathbb{R}^2$ is an image, and Ω is a discrete image domain. The image class label is denoted as $y_i \in \{1, \dots, C\}$, with C being the total number of classes. Our model performs both classification and localization tasks. It is composed of three main parts (Fig.2.3):

1) A transformer-based encoder (Dosovitskiy *et al.*, 2020) f_e , which takes the input image and produces token embedding E , and N class tokens $E_{cl} = \{e_0, \dots, e_{N-1}\}$. Equipped with its own prediction head, this model is trained in a self-supervised manner (Caron *et al.*, 2021). We keep the encoding part for our model and freeze it (no more training) for further use. Its classifier part is discarded. The parameters set of this encoder is denoted as θ_e .

2) A classification head f_c performs classification using the class tokens E_{cl} as input. Its parameters are referred to as θ_c . This head produces per-class probabilities $f_c(E_{cl}) \in [0, 1]^C$, where $f_c(E_{cl})_k = \Pr(k|X)$. It is trained to perform classification using standard cross-entropy: $\min_{\theta_c} -\log(\Pr(y|X))$. Once trained, its parameters, θ_c , are frozen.

3) A CNN localization head f_l , which is a decoder that performs object localization using the token embedding E as input. This encoder produces two full-resolution activation maps which are normalized via a softmax: $S = f_l(E) \in [0, 1]^{|\Omega| \times 2}$, where S^0, S^1 represent the background and foreground maps, respectively. We denote by $S(p) \in [0, 1]^2$ a row of matrix S , with the index $p \in \Omega$ indicating a point location within Ω . Using its own parameters θ_l , this module is trained to localize foreground regions over the input image X associated with its class y . To this end, we use our collected pseudo-labels presented in the next section.

The aforementioned three modules are part of our final model. They are used during training and inference. For our discriminative proposals sampling, we use an additional external pre-trained classifier f_x . It is used to assess the likelihood of a region of an image covering the object

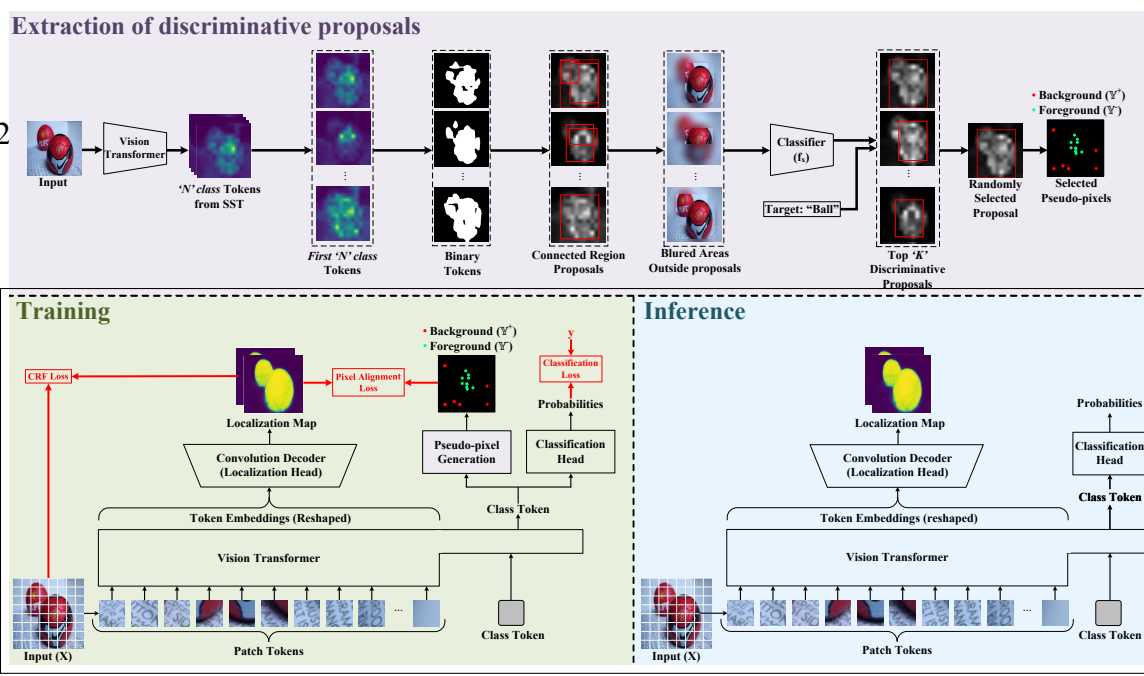


Figure 2.3 *DiPS*: Our proposed method for training a transformer network for WSOL tasks using a combination of localization and classification networks. The full model is composed of a vision transformer encoder, a classification head, and a localization head.

Training: An image class label is required to train the classification head, and our generated pixel-wise pseudo labels are needed to train the localization head. Fig.2.2 illustrates how this pseudo-supervision is computed. **Inference:** In a single forward pass through our model over the input image, the classification head yields the predicted class probabilities, while the localization head localizes the object in the image

class associated with the image. This helps us collect reliable discriminative ROIs and discard non-discriminative background regions. It is trained over the trainset $\mathbb{D}$ to correctly classify each sample using the image class as supervision. In this work, we use a simple the (He *et al.*, 2016) network as f_x .

2.3.2 Discriminative Sampling of Proposals

In this section, we present our proposed strategy to sample discriminative proposals from a set of attention maps, *i.e.*, class token, $\mathbf{E}_{cl} = \{e_0, \dots, e_{N-1}\}$, extracted from a pre-trained SST.

As illustrated in Fig.2.1a, self-supervised transformers produce multiple attention maps that are rich in localization information. These models tend to decompose the scene into multiple objects or parts of objects through attention maps (Caron *et al.*, 2021). While this is clearly beneficial for the localization task, attention maps and localized objects are not associated with semantic meaning, unlike CAM-based methods (Oquab *et al.*, 2015). Therefore, it is

extremely challenging to use these maps directly for localization. To leverage these rich maps, we propose introducing a discriminative strategy to associate regions with semantic meaning, *i.e.*, the image class label. This allows for obtaining reliable discriminative ROIs that can be used as pseudo-labels for training a WSOL model.

Alg.2.1 shows a step-by-step approach to building a pool of discriminative proposals from a labeled image. Fig.2.2 also presents an overview of the approach. First, an external classifier, f_x is trained using the training dataset using image class labels. This model is later used as a scoring function to measure the likelihood of a part of an image being discriminative. The process consists of iterating through all attention maps in the set E_{cl} . Following common assumptions in CAM (Zhou *et al.*, 2016), strong activations in maps are considered as potential foregrounds, while low activations are considered to be backgrounds. Therefore, the attention map is automatically thresholded (Otsu, 1979), and a tight bounding box is computed around all connected regions.

After obtaining the set of all bounding boxes in the image from all attention maps, E_{cl} , we proceed to score each one. We refer to this set as $\mathbb{T}$. To show only the contents of the box to the scoring model f_x , we perturb the image to suppress information outside the box. Particularly, blurring the outside is considered as it has proven to be more efficient in information suppression in deep models (Fong, Patrick & Vedaldi, 2019). The posterior probability of the true image class is used as a score. A higher value indicates a greater likelihood that the box will contain a discriminative part related to the image class label. All boxes, $\mathbb{T}$, are scored and sorted. Only the boxes of the set $\mathbb{P} = \text{top-K}(\mathbb{T})$ are kept for further processing.

This process of generating discriminative proposals allows the building of a rich and diverse pool of potential ROIs, which makes it more advantageous than CAM-based ROIs (Fig.2.1c). The latter are very local and limited to a small discriminative region, which limits their use in generating good localization pseudo-labels. In the next section, we describe how the generated bounding box proposals, $\mathbb{P}$, are leveraged to create pixel-wise pseudo-labels to train a WSOL model.

Algorithm 2.1 Our discriminative proposals generation (Fig.2.2) from attention maps of a self-supervised transformers (SST).

```

Input :Input image:  $X$ ,
        Its class label:  $y$ ,
        Its set of  $N$  class tokens:  $E_{cl} = \{e_0, \dots, e_{N-1}\}$ ,
        External pretrained classifier:  $f_x$ ,
        Maximum number of proposals:  $K$  (top- $K$ ).

Output Set of top- $K$  discriminative proposals:  $\mathbb{P}$ .

:
1  $\mathbb{P} = \emptyset$ .
2 Temporary proposals holder:  $\mathbb{T} = \emptyset$ .
3 for class token  $e \in E_{cl}$  do
4     Threshold the map  $e$  using Otsu (Otsu, 1979) to obtain a binary map  $b$ .
5     Find all connected region proposals in  $b$ , and their tight bounding boxes. Set
         $\mathbb{BBOX}$  as the set of all the bounding boxes obtained.
6     for bounding box  $bx \in \mathbb{BBOX}$  do
7         Perturb the image: Blur the content of the image outside the bounding box  $bx$ .
            Leave the content of the image inside the box. Denote the perturbed image as
             $X_{bx}$ .
8         Compute the probability of the true class:  $f_x(X_{bx})_y$ .
9         Store the bounding box proposal and its probability score:
             $\mathbb{T} \leftarrow \mathbb{T} \cup \{(bx, f_x(X_{bx})_y)\}$ 
10    end for
11 end for
12 Sort  $\mathbb{T}$  using probability scores in descending order.
13 Take the top- $K$  most discriminative proposals:  $\mathbb{P} = \text{top-K}(\mathbb{T})$ .

```


2.3.3 Pixel-wise Pseudo-labels

Using a collected set of region proposals, $\mathbb{P}$, we proceed to create pixel-wise pseudo-labels. In particular, we use these regions to indicate foreground and background pixels. This pseudo-supervision will later be used to train our decoder, f_l , to perform object localization following recent successful approaches over CAMs (Murtaza *et al.*, 2023a; Belharbi *et al.*, 2023a,c, 2022a,c).

The sampling of pixel supervision is tied to a bounding box. Here, we present the sampling of pixel-wise pseudo-labels for a single bounding box, assuming that e is the attention map from the set of class tokens E_{cl} from where a bounding box was generated. Instead of considering the entire content of the box as foreground and keeping it fixed (Kolesnikov & Lampert, 2016), a stochastic sampling of locations is considered. Such a random approach has been shown to be more efficient as it helps avoid overfitting the bounding box (Belharbi *et al.*, 2022c) and promotes exploring object parts. Assuming that an object is contiguous, we use the magnitude of activations inside the box to guide the sampling using a multinomial distribution. In particular, we consider the top- n^+ pixels inside the box for sampling. This helps in exploring different parts of the region inside the box while focusing on the potential pixels. Among these pixels, few locations are sampled as foreground.

Sampling of background pixels is done with respect to all bounding boxes. In particular, we ensure that the sampled background pixel lands outside all the bounding boxes. Similarly to foreground pixels, we guide the sampling of the background pixels by the low magnitude activations of the map e . We sort all activations of e and take the low- n^- pixels from which to sample. Assuming that background regions are uniformly distributed over the image, we use uniform, instead of multinomial, sampling to pick a pixel. As such, few pixels are selected to be the background.

Foreground and background pixel locations are sampled randomly. It is done at every training step for each image. This allows exploring different regions and prevents overfitting to a specific part. Additionally, it gives the decoder enough time to allow the emergence of consistent

foreground regions. Since the training is done using only a few pixels at a time, the decoder learns to consistently *fill in the gap* across the rest of the image by transferring the knowledge learned from the pseudo-labels to other similar regions.

The sampled pixel locations are encoded in the image domain $\mathcal{Q}'$. They are gathered in a partial pseudo-label mask $Y(p) \in \{0, 1\}^2$ with labels 0 for background and 1 for foreground. Locations with unknown labels are encoded as unknown. Training the decoder is done by performing a *pixel-wise alignment* between the output maps S and the pseudo-supervision Y . At location p , we use partial cross-entropy as follows:

$$H_p(Y, S) = -(1 - Y(p)) \log(S^0(p)) - Y(p) \log(S^1(p)), \text{ for } p \in \mathcal{Q}'. \quad (2.1)$$

In practice, at each training step, and in each image, we sample several pixels from the foreground and background and ensure that they are balanced. Additionally, since the proposals are selected to be aligned with the true image class, *i.e.*, y , the selected foreground pixels follow as well. They indicate the same object annotated in the image. Therefore, the final foreground map S^1 points similarly to the image class. This allows to build a single model for all classes.

2.3.4 Training DiPS Architecture

Our model is composed of three main parts that we train separately. The full model is illustrated in Fig.2.3. The first module is a transformer-based encoder (Dosovitskiy *et al.*, 2020), $f_e(\cdot; \theta_e)$. Initially, it is equipped with a classification output module. It is trained in a self-supervised fashion, similarly as in (Caron *et al.*, 2021). Once trained, the classification module is discarded, and only the encoder part is kept for our model. Its pre-trained parameters are frozen and are not trained further. In practice, we typically use pre-trained models on ImageNet dataset (Russakovsky *et al.*, 2015a) or fine-tune it on the corresponding dataset.

In addition, we have a classification module, $f_c(\cdot; \theta_c)$, which relies on the encoder output features, E_{cl} , to classify the image: $f_c(E_{cl})_k = \Pr(k|X)$. It is trained using standard cross-entropy:

$$\min_{\theta_c} -\log(\Pr(y|\mathbf{X})) . \quad (2.2)$$

Once this module is trained, its weights are frozen and are no longer modified.

The last module, $f_l(\cdot; \theta_l)$, is for the localization task. It is a CNN-based decoder that outputs two full-size maps for foreground and background regions. Its training loss combines two elements: pixel-wise pseudo-labels and a Conditional Random Field (CRF) loss (Tang *et al.*, 2018). To ensure that the activations of the output map $\mathbf{S}$ are well aligned with the object boundaries, a CRF loss (Tang *et al.*, 2018) is employed. This loss considers both the pixels' proximity and color similarity:

$$\mathcal{R}(\mathbf{S}, \mathbf{X}) = \sum_{r \in \{0,1\}} \mathbf{S}^r{}^\top \mathbf{W} (\mathbf{1} - \mathbf{S}^r) , \quad (2.3)$$

where $\mathbf{W}$ denotes an affinity matrix in which $\mathbf{W}[i, j]$ captures the color similarity and proximity between pixels i, j in the image $\mathbf{X}$. We employ a Gaussian kernel (Krahenbuhl & Koltun, 2011) to compute $\mathbf{W}$.

The sampling of pixel-wise pseudo-labels from a single bounding box proposal was presented in a previous section (Sec.2.3.3). However, we collected a pool of top- K bounding boxes, *i.e.*, $\mathbb{P}$. To avoid overfitting over a single bounding box and promote exploring the different object parts, we randomly select a bounding box from $\mathbb{P}$, at each training step and for each sample. Then, we proceed to sample pixel-wise pseudo-labels from the selected box. The following is the full training loss for the localization decoder for a single sample:

$$\min_{\theta_l} \lambda_1 \sum_{p \in \Omega'} \mathbf{H}_p(\mathbf{Y}, \mathbf{S}) + \lambda_2 \mathcal{R}(\mathbf{S}, \mathbf{X}) , \quad (2.4)$$

where λ_1, λ_2 are weighing coefficients. Following (Tang *et al.*, 2018), we set $\lambda_2 = 2e^{-9}$. A validation set is used to search for λ_1 .

2.4 Results and Discussion

2.4.1 Experimental Methodology

Datasets: To validate our proposed method, we employed four challenging WSOL datasets, namely, OpenImages, ILSVRC, CUB-200-2011, and TelDrone. **(i) OpenImages** (Benenson, Popov & Ferrari, 2019; Choe *et al.*, 2020) is a dataset comprising 37,319 images divided into 100 classes. We allocated 29,819 images for training and 5,000 images for testing. The remaining 2,500 images served as the validation set. **(ii) ILSVRC** includes approximately 1.2 million images spanning over 1,000 classes. Following the recommendations in (Choe *et al.*, 2020), we extracted 50,000 images for training and testing and 10,000 images for validation. **(iii) CUB200-2011** consists of 11,788 images distributed across 200 categories, with 5,994 designated for training and 5,794 for testing (Welinder *et al.*, 2010). To perform a validation and hyperparameter search, we used an independent validation set of 1,000 images compiled by (Choe *et al.*, 2020). **(iv) TelDrone** is a private dataset, which is owned by Ericsson Inc¹. It includes 915 high-resolution 4K images captured via a drone orbiting around a tower site. These images are bifurcated into two classes: one class containing images with an inspection site, and the other, those devoid of it. We divided these images into a training set of 797 images, a validation set containing 13 images, and a test set with 105 images.

Evaluation measures: For performance evaluation, four measures and three error metrics were considered. They are commonly used in WSOL task evaluation (Choe *et al.*, 2020; Gao *et al.*, 2021) – **(i) PxAP** denotes pixel-wise precision and recall at a particular threshold. **(ii) MaxBoxAccV2** represents the average proportion of predicted bounding boxes with IoU values exceeding a specified threshold relative to the ground truth map. It is computed by averaging the results across three different IoU thresholds $\delta = \{30\%, 50\%, 70\%\}$. **(iii) Top-1** localization, defined as the fraction of images, where the predicted class label correctly matches the ground truth and its corresponding IoU exceeds $\delta = 50$ **(iv) Top-5** localization is the fraction of images in which the ground truth class label is among the top-5 predicted class labels and the IoU is

¹ <https://www.ericsson.com/>

greater than $\delta = 50$. (v) **Localization part error (LPE)** detects instances where the localization map partially captures an object along with an intersection over the predicted bounding box (IoP) value greater than 0.5. (vi) **Localization more error (LME)** indicates that the predicted bounding box is larger than the actual bounding box, potentially including adjacent objects or background regions. This detects the instances whose intersection over the annotated-bounding box (IoA) value is greater than 0.7. (vii) **Multi-instance Error (MIns-Error)** indicates the ratio of the predicted bounding box intersecting with more than one bounding box, with an intersection-over-ground-truth-box (IoG) value exceeding 0.3.

Implementation details: In all experiments, we follow the protocol proposed in (Choe *et al.*, 2020). For all datasets, we employ a batch size of 32, resize images to 256×256 , and then randomly crop to 224×224 , followed by random horizontal flipping, as described in (Choe *et al.*, 2020). We then employ the Stochastic Gradient Descent (SGD) optimizer and search the learning rate between $\{0.1, 0.0001\}$. For training, we employ 50 epochs for the CUB and TelDrone datasets and reduce the epochs to 10 for the OpenImages and ILSVRC datasets.

Baseline Models: To validate the performance of our proposed method, we compare our results with various state-of-the-art methods, as presented in Table 2.1. We obtain the quantitative results of CAM (Zhou *et al.*, 2016), ADL (Choe & Shim, 2019), HaS (Singh & Lee, 2017), ACoL (Zhang *et al.*, 2018c), SPG (Zhang *et al.*, 2018d), and CutMix (Yun *et al.*, 2019) from (Choe *et al.*, 2020). For other methods, we present the quantitative results as reported in their respective publications. Additionally, we reproduce the qualitative results of CAM (Zhou *et al.*, 2016), HaS (Singh & Lee, 2017), ADL (Choe & Shim, 2019), ACoL (Zhang *et al.*, 2018c), SPG (Zhang *et al.*, 2018d), CutMix (Yun *et al.*, 2019), and TS-CAM (Gao *et al.*, 2021) by following the protocols outlined in (Choe *et al.*, 2020). Furthermore, we compare the visual results of all datasets with the `class` tokens of the last layer of SST, which are used to collect pseudo-labels.

2.4.2 Comparison with State-of-Art Methods

Quantitative Comparison. Evaluation of different localization datasets demonstrates the performance of DiPS. Table 2.1 shows that DiPS surpasses the baseline model and other related methods on OpenImages when evaluated using PxAP metric. Specifically, DiPS achieves a PxAP of 74.9%, surpassing other recent methods. In particular, our method yields better results than the recent method, F-CAM (Belharbi *et al.*, 2022c), with a PxAP of 72.2%, which relies on pseudo-labels from CAMs. Similarly, on the ILSVRC, CUB, and TelDrone datasets, DiPS achieves competitive performance in terms of MaxBoxAccV2, top-1 and top-5 localization accuracy metrics, as shown in Table 2.1, 2.2 and 2.3. Additionally, we also present the localization accuracy at different thresholds δ on the ILSVRC dataset to show that our approach outperforms the other methods with a high IoU value. This indicates that our method produces a more accurate localization, as demonstrated in the following qualitative evaluation. Qualitative results also show that our activation maps have sharp boundaries.

Table 2.1 MaxBoxAccV2 and PxAP performance of proposed and state-of-the-art methods on the CUB and OpenImages datasets

Method	Backbone	OpenImages	CUB			
		PxAP	MaxBoxAccV2	Top-1 loc.	Top-5 loc.	
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr,2016</i>)	ResNet50	63.2	63.7	56.1	—	
HaS (Singh & Lee, 2017) (<i>iccv,2017</i>)	ResNet50	58.1	64.7	60.7	—	
ACoL (Zhang <i>et al.</i> , 2018c) (<i>cvpr,2018</i>)	ResNet50	57.3	66.5	57.8	—	
SPG (Zhang <i>et al.</i> , 2018d) (<i>eccv,2018</i>)	ResNet50	62.3	60.4	51.5	—	
ADL (Choe & Shim, 2019) (<i>cvpr,2019</i>)	ResNet50	58.7	66.3	41.1	—	
CutMix (Yun <i>et al.</i> , 2019) (<i>eccv,2019</i>)	ResNet50	62.5	62.8	54.5	—	
PAS (Bae, Noh & Kim, 2020) (<i>eccv,2020</i>)	GoogleNet	63.3	—	—	—	
PAS (Bae <i>et al.</i> , 2020) (<i>eccv,2020</i>)	ResNet50	60.9	—	—	—	
ICL (Minsong <i>et al.</i> , 2020) (<i>accv,2020</i>)	ResNet50	—	63.1	56.1	—	
CAM-IVR (Kim, Choe, Yun & Kwak, 2021) (<i>iccv,2021</i>)	InceptionNet	63.6	66.9	—	—	
CAM-IVR (Kim <i>et al.</i> , 2021) (<i>iccv,2021</i>)	ResNet50	58.9	60.9	—	—	
TS-CAM (Gao <i>et al.</i> , 2021) (<i>iccv,2021</i>)	DeiT-S	—	76.7	71.3	83.8	
VITOL-GAR (Gupta <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	DeiT-S	—	72.4	—	—	
VITOL-LRP (Gupta <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	DeiT-S	—	73.1	—	—	
PDM (Meng <i>et al.</i> , 2022a) (<i>tip,2022</i>)	ResNet50	—	72.4	—	—	
C ² AM (Xie, Xiang, Chen & et. al, 2022b) (<i>cvpr,2022</i>)	ResNet50	—	83.8	76.6	89.15	
SCM (Bai <i>et al.</i> , 2022) (<i>eccv,2022</i>)	DeiT-S	—	89.9	—	—	
TRT (Su <i>et al.</i> , 2022) (<i>corr,2022</i>)	DeiT-B	—	82.0	76.5	88.0	
BGC (Kim <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	ResNet50	—	75.9	73.2	86.7	
BGC (Kim <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	VGG16	—	80.1	70.8	88.1	
BR-CAM (Zhu <i>et al.</i> , 2022a) (<i>eccv,2022</i>)	ResNet50	67.6	—	—	—	
CREAM (Xu <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	ResNet50	64.7	73.5	76.0	—	
F-CAM+XGradCAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv,2022</i>)	VGG16	69.0	80.1	22.0	49.6	
F-CAM+LayerCAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv,2022</i>)	ResNet50	72.2	82.7	47.7	76.1	
DiPS (ours)	DeiT-S	74.9	91.5	79.2	92.2	

Qualitative Comparison. Visual results of DiPS and related baselines are presented in Fig.2.4-2.8, indicating that our method yields good localization. Baseline methods focus

Table 2.2 MaxBoxAccV2 performance of proposed and state-of-art methods on the ILSVRC dataset, along with IoU at different threshold values, denoted by δ . We adopted some of the results for baseline methods from (Kim *et al.*, 2022)

Method	Backbone	MaxBoxAccV2				loc.	
		$\delta = 0.3$	$\delta = 0.5$	$\delta = 0.7$	Mean	Top-1	Top-5
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr,2016</i>)	ResNet50	83.7	65.7	41.6	63.7	51.8	—
HaS (Singh & Lee, 2017) (<i>iccv,2017</i>)	ResNet50	83.7	65.2	41.3	63.4	49.9	—
SPG (Zhang <i>et al.</i> , 2018d) (<i>eccv,2018</i>)	ResNet50	83.9	65.4	40.6	63.7	47.4	—
ADL (Choe & Shim, 2019) (<i>cvpr,2019</i>)	ResNet50	83.6	65.6	41.8	63.7	48.5	—
CutMix (Yun <i>et al.</i> , 2019) (<i>eccv,2019</i>)	ResNet50	83.6	65.6	41.8	63.7	51.5	—
ICL (Minsong <i>et al.</i> , 2020) (<i>accv,2020</i>)	ResNet50	84.3	67.6	43.6	65.2	48.4	—
BGC (Kim <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	ResNet50	86.7	71.1	48.3	68.7	53.8	65.7
DiPS (ours)	DeiT-S	83.2	69.7	51.7	68.2	56.4	66.7

Table 2.3 Quantitative comparison of DiPS and existing state-of-the-art methods on the TelDrone Dataset using MaxBoxAccV2 metric. Experimental results on this proprietary dataset indicate robust performance across various datasets

Methods	MaxBoxAccV2	
	VGG16	ResNet50
CAM Zhou <i>et al.</i> (2016) (<i>cvpr,2016</i>)	55.9	45.2
HaS Singh & Lee (2017) (<i>iccv,2017</i>)	60.3	49.1
ACoL Zhang <i>et al.</i> (2018c) (<i>cvpr,2018</i>)	43.4	59.1
SPG Zhang <i>et al.</i> (2018d) (<i>eccv,2018</i>)	62.8	67.3
ADL Choe & Shim (2019) (<i>cvpr,2019</i>)	66.0	63.5
CutMix Yun <i>et al.</i> (2019) (<i>eccv,2019</i>)	57.2	50.3
TS-CAM Gao <i>et al.</i> (2021) (<i>iccv,2021</i>)	DeiT-S: 72.2	
DiPS (ours)	DeiT-S: 92.4	

on common areas of discriminative regions that are shared among different objects of the same class, as they highlight these areas. The blobby nature of these maps necessitates the selection of optimal threshold values to accurately identify a particular object. However, this thresholding mechanism renders localization maps unreliable and may include regions with concealed activations for an object of interest. In contrast to the baseline models, DiPS is able to highlight foreground objects, maintaining a uniform activation across different object parts. For instance, on the CUB-200-2011 dataset, our method produces robust activation maps with sharper boundaries, outperforming current state-of-the-art techniques. Additionally, our method maintains a consistent localization performance across different datasets, including ILSVRC, OpenImages, and TelDrone. For instance, DiPS is able to distinguish the foreground object

even in scenarios where the object’s texture closely resembles the background regions(Fig.2.7). Additionally, we also compared the output of our method with the `class` tokens used to collect pseudo-labels (Fig.2.4-2.7). These visualizations indicate that our method is able to identify objects efficiently while mitigating the noise that is present in the `class` token of the last attention blocks.

All in all, state-of-the-art methods generate localization maps for objects of interest with varying intensities. They are also able to properly draw a bounding box around the object due to an extensive threshold search, which can include low-scoring areas in the localization map. In contrast, DiPS produces localization maps that identify a particular object with sharper boundaries, thus eliminating the need for a precise threshold value.

Comparison with Self-Supervised Methods. In addition to the above results, we also report the `MaxBoxAccV2` performance of different self-supervised vision transformers, including SST (Caron *et al.*, 2021), used to collect pseudo-labels to train our model (Table 2.4). To extract localization maps from these models, we gather different attention maps and overlay them to the original image to produce different perturbed images. One of the perturbed images is selected based on the classifier’s score and its corresponding attention map to compute the localization performance. Moreover, we also report the `MaxBoxAccV2` performance of maps extracted from the Contrastive Language-Image Pretraining (CLIP-ES) model (Lin *et al.*, 2023). CLIP-ES is a prompt-based model trained on paired text and image datasets. The CLIP-ES model requires a class label alongside the input image to localize the ROI for the specified class. This distinguishes CLIP from traditional WSOL models, which only rely on the image for inference CAMs. The design of CLIP inherently confers upon it a significant advantage, as it is explicitly aware of the target class during its computations. In our experiments, we input a predicted class label from a pretrained classifier, along with the image, to the CLIP model.

Empirical results in Table 2.4 indicate that our model is capable of achieving high performance compared to the self-supervised methods, using only class-level labels in terms of `MaxBoxAccV2`. Visual results show that, in contrast to our method, self-supervised methods hotspot a different



Figure 2.4 Samples from the test set on the CUB-200-2011 dataset generated by our proposed and state-of-the-art methods. In different examples, the results of baseline methods highlight different parts of the object, but the entire object is encompassed within the bounding box due to an extensive search of thresholds. In contrast, the map generated by our method covers the full object with relatively consistent activation values across the object. Additionally, our model effectively mitigates noise in the generated map, which is present in the class tokens used to produce pseudo-labels. Here, **green** denotes the ground truth bounding box, while **red** corresponds to the predicted bounding box

region, which requires an optimal threshold to draw an optimal bounding box, as shown in Fig.2.9. This analysis not only enriches the comparative analytical landscape but also fortifies the benefits of DiPS as a robust and generalizable methodology for object localization.

Table 2.4 A comparative analysis of our approach against self-supervised and prompt-based models; In self-supervised transformers, class tokens are utilized to compute localization performance. For this purpose, we harvest class tokens from the last attention block of the transformer. These class tokens are then binarized and used to perturb the original image by blurring the background regions. Subsequently, this perturb is processed by a classifier, and a map with the maximum classifier score is selected for computation localization scores. This selection strategy closely adheres to the pseudo-label generation process employed in our method. For CLIP-ES, class labels are obtained from pre-trained models that are then passed through the CLIP model along with the input image to obtain attention map Lin *et al.* (2023). Moreover, our method is able to surpass self-supervised vision transformers and prompt-based models

Method	MaxBoxAccV2		PxAP
	CUB-200-2011	ILSVRC	OpenImages
DINO (DeiT-S) Caron <i>et al.</i> (2021)	65.1	59.9	52.5
ESVIT (SWIN-ViT) Li, Yang, Zhang & et. al (2022)	51.5	54.5	39.1
MoCoV3 (ViT-B) Chen, Xie & He (2021c)	47.3	56.1	44.4
SimMIM (ViT-B) Xie, Zhang, Cao & et. al (2022c)	31.2	44.6	22.1
CLIP-ES Lin <i>et al.</i> (2023)	79.9	61.7	54.2
DiPS (DeiT-S) - ours	91.5	68.2	74.3

Complexity Analysis. DiPS achieves competitive performance in comparison to the baseline model introduced in our preliminary study (Murtaza *et al.*, 2023a). The model presented in this paper requires a single forward pass to generate both classification scores and localization maps. Moreover, the forward pass of our model heads (localization and classification) requires merely 2.396G Multiply-Accumulate (MACs) operations with 1.616M parameters, as opposed to our previous model (Murtaza *et al.*, 2023a), which called for 31.895G MACs with 58.078M parameters for inference.

Analysis of Distribution Shift. In this section, we analyze the impact of varying thresholds on the localization performance, along with the distribution of activation values corresponding to the object of interest. The shift in MaxBoxAcc at different threshold values for our model selected during the hyperparameter search through MaxBoxAcc is presented in Fig.2.5. For the

baseline methods, we observe that the **MaxBoxAcc** rapidly declines to zero as the threshold increases. This makes it challenging to search for the optimal threshold values for each image during inference. In contrast, the output generated by our method exhibits a lower susceptibility to threshold variations.

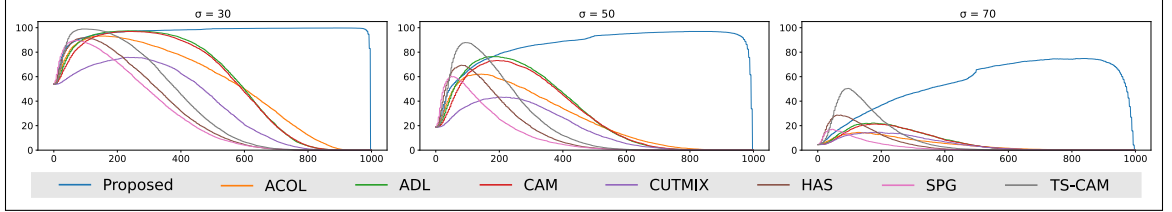


Figure 2.5 The **MaxBoxAcc** performance of DiPS and state-of-the-art methods is at different threshold values ($\delta = \{30, 40, 70\}$) calculated on a test set from the CUB-200-2011 dataset

2.4.3 Error Analysis and Ablation Study

For a fair evaluation of our presented approach, different error metrics adopted from (Gao *et al.*, 2021) are employed (Section 2.4.1). Using these metrics, we analyze the performance of our model on the CUB-200-2011 and ILSVRC datasets (Table 2.6). These results show that the localization maps produced by our method are able to accurately localize a particular object, avoiding any overestimation or underestimation of the object of interest. In contrast to the related methods, **MIns-Error** on the ILSVRC dataset demonstrates the ability of our method to localize a specific object rather than multiple objects. This analysis also reveals that the maps generated by our method are highly robust and exhibit considerably fewer errors as compared to the baseline methods.

Furthermore, the performance of our model, selected during the hyperparameter search without a CRF loss, is presented in Table 2.6. This indicates that CRF loss significantly contributes to the model’s performance. Without the CRF loss, our pixel-alignment loss exhibits a slightly lower performance as compared to the results reported in the previous section. Thus, the inclusion of the CRF loss notably enhances our model’s performance.

Table 2.5 Error analysis of our method. Results of baseline methods (VGG16, InceptionV3, TS-CAM) are borrowed from (Gao *et al.*, 2021)

Methods	CUB		ILSVRC		
	LPE ↓	LME ↓	MIns-Error ↓	LPE ↓	LME ↓
VGG16 (CAM)	21.91	10.53	10.65	3.85	9.58
InceptionV3 (CAM)	23.09	5.52	10.36	3.22	9.49
TS-CAM (Gao <i>et al.</i> , 2021)	6.30	2.85	9.13	3.78	7.65
DiPS (our)	.002	.001	0.03	0.01	0.04

Table 2.6 Ablation study of our model for loss functions of localization head

Losses	CUB (MaxBoxAccV2)	ILSVRC (MaxBoxAccV2)	OpenImages (MaxBoxAccV2)
Pixel pseudo-labels	89.2	67.2	70.9
Pixel pseudo-labels + CRF	91.5	68.2	74.9

2.5 Conclusion

In this paper, we proposed a novel transformer-based method for the WSOL task. In particular, we designed a discriminative approach to sample reliable proposals from the attention maps of a self-supervised transformer. Such maps have proven to be rich with localization information but lack semantic meaning. Using a pre-trained classifier, we score region proposals and measure their likelihood of containing the true image class. Only top-scoring proposals are retained. Our sampling strategy allows us to build a diverse and rich pool of region proposals to train a WSOL model. Additionally, we design a transformer-based model for WSOL that aims to achieve the best performance for both classification and localization tasks simultaneously. Experimental results demonstrate that our method is capable of producing reliable localization maps, outperforming the class tokens used to generate pseudo-labels. Our model generates robust localization maps that exhibit less sensitivity to threshold values. Results of our method over four challenging datasets show its benefits compared to state-of-the-art methods. Furthermore, our model can produce localization maps with consistent intensities across all object parts, unlike its counterpart methods.



Figure 2.6 Visual results for proposed and state-of-the-art methods on the ILSVRC dataset. Here, **green** denotes the ground truth bounding box, while **red** corresponds to the predicted bounding box

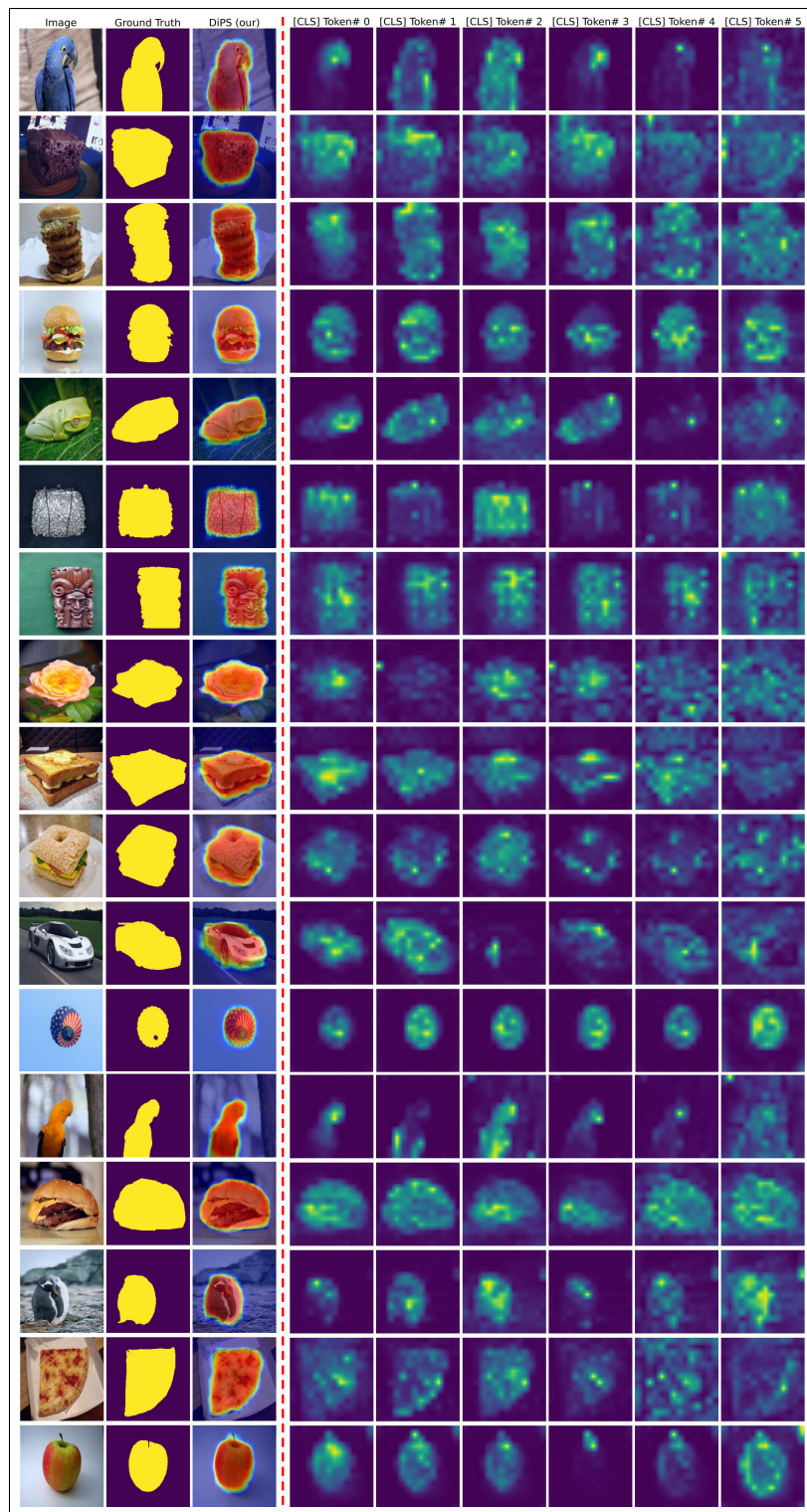


Figure 2.7 Visualization of results on the OpenImages dataset

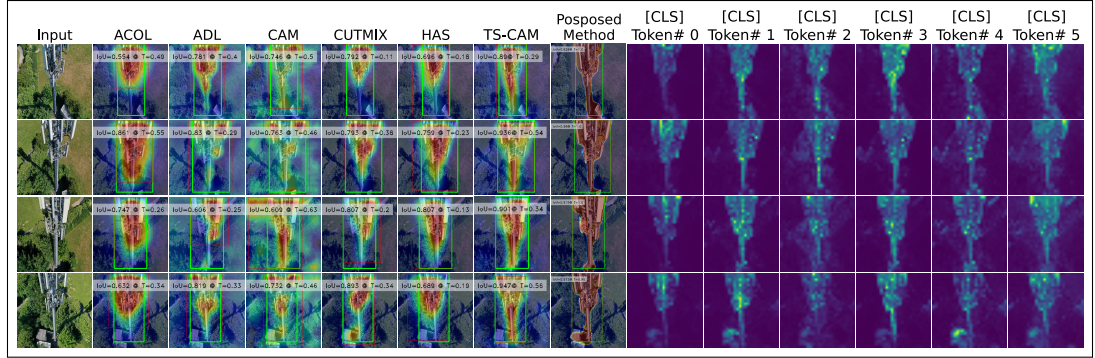


Figure 2.8 Visualization of results on the TelDrone dataset. We added a few examples for this dataset, given the proprietary restrictions specified by Ericsson Incorporation. Here, **green** denotes the ground truth bounding box, while **red** corresponds to the predicted bounding box



Figure 2.9 Visual results of our methods compared to the self-supervised method. Here, **green** denotes the ground truth bounding box, while **red** corresponds to the predicted bounding box

CHAPTER 3

TED-LOC: TEXT DISTILLATION FOR WEAKLY SUPERVISED OBJECT LOCALIZATION

Shakeeb Murtaza^a , Soufiane Belharbi^a , Marco Pedersoli^a , Eric Granger^a

^aDepartment of Systems Engineering, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper Submitted in *IEEE/CVF International Conference on Computer Vision (ICCV)*,
October 2025

Abstract Weakly supervised object localization (WSOL) using classification models trained with only image-class labels remains an important challenge in computer vision. Given their reliance on classification objectives, traditional WSOL methods like class activation mapping focus on the most discriminative object parts, often missing the full spatial extent. In contrast, recent WSOL methods based on vision-language models like CLIP require ground truth classes or external classifiers to produce a localization map, limiting their deployment in downstream tasks. Moreover, methods like GenPrompt attempt to address these issues but introduce considerable complexity due to their reliance on conditional denoising processes and intricate prompt learning. This paper introduces Text Distillation for Localization (TeD-Loc), an approach that directly distills knowledge from CLIP text embeddings into the model backbone and produces patch-level localization. Multiple instance learning of these image patches allows for accurate localization and classification using one model without requiring external classifiers. Such integration of textual and visual modalities addresses the longstanding challenge of achieving accurate localization and classification concurrently, as WSOL methods in the literature typically converge at different epochs. Extensive experiments show that leveraging text embeddings and localization cues provides a cost-effective WSOL model. TeD-Loc improves top-1 loc. accuracy over state-of-the-art models by about 5% on both CUB and ILSVRC datasets, while significantly reducing computational complexity compared to GenPrompt.

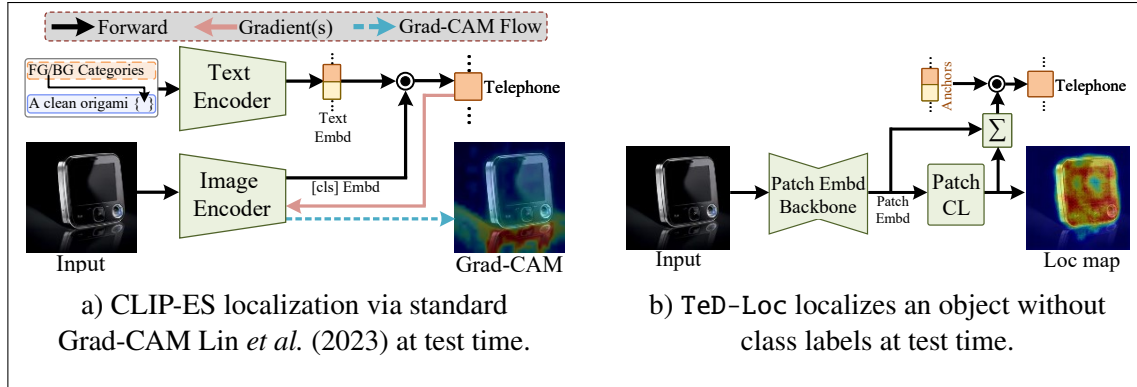


Figure 3.1 Comparison of our TeD-Loc versus CLIP-ES Lin *et al.* (2023) methods for extracting localization maps from CLIP. **(A)** CLIP-ES utilizes Grad-CAM to extract localization maps from CLIP, requiring GT class labels during inference. **(B)** In contrast, our TeD-Loc model distills knowledge from CLIP text embeddings into the visual encoder during training, allowing it to produce both classification scores and localization maps without requiring class labels during inference.

3.1 Introduction

WSOL is a critical yet challenging task in computer vision, aiming to localize objects within images using a model trained using only image-class labels rather than instance-level annotations. Popular class activation mapping (CAM) method Zhou *et al.* (2016) leverage classification models for generating localization maps, however, they inherently focus on the most salient regions of an object and often fail to capture the full spatial extent Belharbi *et al.* (2022c). This limitation arises because discriminative models are optimized to minimize mutual information between different instances of the same class. Various strategies have been proposed to mitigate this issue, including spatial regularization Lu, Jia, Xie & et. al (2020a); Wu *et al.* (2022); Xue *et al.* (2019); Yun *et al.* (2019); Zhang *et al.* (2018d), adversarial erasing Zhang *et al.* (2018c); Choe & Shim (2019); Choe *et al.* (2021b), and leveraging pseudo-labeling Belharbi *et al.* (2022c); Zhang *et al.* (2018d); Murtaza *et al.* (2023a). However, these approaches are constrained by the local receptive fields of convolutional neural networks (CNNs), which limit their ability to capture global dependencies essential for complete object localization.

Vision transformers (ViTs) Gao *et al.* (2021) have recently shown potential in modeling long-range dependencies through self-attention, thereby mitigating some limitations of CNNs for WSOL. However, they lack the local inductive biases inherent to CNNs, often resulting in weaker local feature representations. Vision-language models, particularly contrastive language-image pre-training (CLIP), have emerged as a promising direction by aligning textual and visual features, which can be leveraged for localization tasks using class-level labels Lin *et al.* (2023). Yet, the predominant approaches for extracting localization maps from CLIP, such as gradient-based and attention manipulation methods, rely heavily on ground truth (GT) class information. This dependency leads to performance degradation when predicted classes derived from an external model are employed because of feature misalignment and class confusion – exemplified by the frequent conflation by CLIP of similar classes like "airplane" and "aircraft" Wang & Kang (2024). Consequently, employing a CLIP model without fine-tuning introduces substantial errors in downstream tasks, prompting the need for strategies that can learn precise localization cues while minimizing reliance on explicit class labels.

Recently, GenPrompt Zhao *et al.* (2023) attempted to address these challenges by framing WSOL as a conditional denoising process, leveraging CLIP embeddings to capture discriminative regions. While GenPrompt improves localization by using CLIP embeddings, they still rely on external classifiers or GT class labels during inference, adding to their complexity and limiting their applicability in real-world scenarios. Moreover, despite robust map generation capabilities, a fundamental limitation remains the inability of CLIP-based methods to localize objects within an image without prior class information. This constraint poses a significant challenge for downstream tasks, as they require computing class labels beforehand (Fig.3.1a). Given these challenges, we seek to effectively harness vision-language models to learn precise localization cues for WSOL, while mitigating misclassification and reducing reliance on GT labels during inference.

To address these challenges, we propose Text Distillation for Localization (TeD-Loc), which learns to localize by distilling knowledge from CLIP text embeddings because they serve as a powerful link between visual and textual modalities. It learns localization information by

transferring knowledge from text embeddings to the localization module (Fig.3.1b). Using contrastive learning within a teacher-student framework, the visual representations of our model are aligned with text embeddings. This alignment is guided by pseudo-labels that can be extracted from any CAM-based method. Learning from text embedding allows TeD-Loc to achieve state-of-the-art performance using one model selected using the best localization performance without requiring separate classifiers trained and selected using the validation set over the classifier’s score. TeD-Loc introduces a new paradigm where classification is achieved through localization, thereby eliminating the need for model selection over the classifier’s score. By training the E_V to distinguish between foreground (FG) and background (BG) regions based on their similarity to text embeddings, TeD-Loc enables the model to classify and localize objects simultaneously. Furthermore, to address the limitations of CLIP’s frequent conflation of semantically similar classes, we propose a method to orthogonalize text embeddings before distillation. By default, text embeddings in CLIP may not sufficiently discriminate between similar classes because of their proximity in the embedding space. To mitigate this issue, we decompose the embeddings using QR decomposition GANDER (1980) and utilize the resulting orthogonal basis vectors for alignment.

More specifically, our method employs a transformer-based architecture that decomposes an image into a set of patches, generating upsampled patch embeddings through our model backbone. Each patch embedding is individually classified to estimate its likelihood of representing a FG or BG region. These classification scores are then stitched together to produce a localization map, highlighting regions of interest within the image. The global classification score for the entire image is a weighted average of the patch embeddings, where the weights are derived from the classifier scores. Our approach is inspired by the multiple instance learning (MIL) framework, where each image consists of a “bag” containing multiple “instances” (image patches) with only bag-level labels available during training. Leveraging MIL, object localization and classification are performed simultaneously by assigning higher weights to discriminant patches, without relying on external classifiers or prior class information. This aligns well with the weakly

supervised nature of object localization, enabling our model to independently produce accurate localization maps and classification scores.

Our main contributions are summarized as follows.

- (1) A novel TeD-Loc method is introduced that distills knowledge from the CLIP E_T to guide object localization using pseudo-labels extracted from CLIP. By leveraging the synergy between visual and textual embeddings, TeD-Loc effectively identifies FG regions and suppresses BG noise without resorting to manual prompt engineering or defining BG categories for each dataset.
- (2) A new module is proposed to transfer knowledge from text embeddings to pixel levels, ensuring that FG and BG embeddings are separated. By maximizing the similarity between FG patch embeddings and text embeddings, the model aligns visual and textual modalities. Further, we employed a localization module to map these patch embeddings to FG/BG regions producing localization maps.
- (3) A classification module is introduced to leverage localization scores to compute the expected embeddings of FG regions. Using a weighted average of patch embeddings, where the weights are derived from the FG localization map, we ensure that FG embeddings align with the correct class embeddings. This eliminates the need for an external classifier and allows our model to classify and localize simultaneously.
- (4) To mitigate the tendency of CLIP to conflate semantically similar classes due to proximity in its embedding space, we propose orthogonalizing text embeddings before alignment. This reduces embedding overlap, allowing our model to achieve competitive performance across both localization and classification tasks.
- (5) Extensive experiments on the challenging CUB and ILSVRC datasets show that our proposed strategy can outperform state-of-the-art methods in terms of **top-1 loc.** performance, yet significantly reduce computational complexity versus GenPrompt.

3.2 Related Work

(a) Weakly supervised object localization. WSOL is a challenging task that seeks to localize objects using only image-level supervision. The foundational work in WSOL Zhou *et al.* (2016)

proposes to harvest CAMs from pre-trained CNNs, leveraging global average pooling (GAP) to guide the network’s attention toward specific regions in an image. Despite its impact, CAM and related CNN-based approaches are constrained to highlight discriminative regions, often neglecting complete object extents. This limitation has led to the development of different WSOL methods that can look beyond discriminative regions.

Erasing-based methods aim to mitigate CAM partial activation by selectively obscuring parts of an image to encourage broader localization. HaS Singh & Lee (2017) and CutMix Yun *et al.* (2019) employ random erasure, which forces the network to explore different object parts beyond discriminative regions. Building on this, adversarial erasing methods like ACoL Zhang *et al.* (2018c) and ADL Choe & Shim (2019) use dual classifiers to identify and erase dominant regions, uncovering complementary object regions. Techniques like SPG Zhang *et al.* (2018d) go further, integrating pixel-wise correlation constraints to maintain context and consistency across object regions.

Other works target the inherent challenge in CNNs to capture only local semantic features due to limited receptive fields. Consequently, newer methods leverage structural cues and integrate BG suppression techniques. SPA Pan *et al.* (2021) enhances structural consistency, while PSOL Zhang *et al.* (2020c) introduces a two-stage WSOL approach that decouples classification from localization tasks, providing robust pseudo annotations for regression without class constraints. Methods such as BAS Wu *et al.* (2022) reinforce this separation by suppressing BG regions, emphasizing FG areas critical to localization.

To overcome the inherent CAM limitations in capturing long-range dependencies, transformer-based approaches for WSOL are gaining traction. Transformers, known for their self-attention mechanism, enable networks to capture both local and global feature dependencies. Vision Transformer (ViT) Sharir, Noy & Zelnik-Manor (2021) and DETR Carion *et al.* (2020) demonstrate the potential of self-attention in vision, and in WSOL, TS-CAM Gao *et al.* (2021) leverages token-patch fusion with semantic maps to improve spatial coherence. By exploiting

and NACLIP Hajimiri *et al.* (2024) propose modifying the last attention block to produce score maps for each patch embedding, enabling segmentation map generation without requiring a backward pass. Similarly, Yang & Gong (2024) introduces learnable prompts in CLIP and uses pseudo-labels from SAM Kirillov *et al.* (2023) to fine-tune for WSSS.

While CLIP-based methods yield competitive maps, these methods rely on hand-crafted textual templates and predefined class representations — such as prompts like “a photo of [CLS].” This requires prior knowledge of the specific class name before producing a localization map of each image. This reliance restricts the model’s adaptability across different computer vision tasks. It also leads to substantial performance degradation when using predicted classes, due to feature misalignment and class confusion. These limitations underscore a critical challenge: the need for WSOL methods that can learn precise localization cues without reliance on explicit class labels or predefined textual templates. To address this challenge, we propose a novel method that distills knowledge from CLIP’s text-image representations to guide the localization network. Additionally, we mitigate the tendency of CLIP to conflate semantically similar classes by orthogonalizing the text embeddings before alignment, reducing semantic overlap and improving discriminability. This enables the model to classify the image by computing the similarity between visual embeddings and class anchors as visual embeddings are pushed toward class-text embeddings.

3.3 The Proposed Method

Let us consider a training set $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^N$ of N images, where each image $x_i \in \mathbb{R}^{H \times W \times 3}$ is associated with an image-level label $y_i \in \{1, \dots, K\}$, representing one of K object classes, with no bounding box (bbox) supervision. WSOL methods seek to train a model for object localization and classification using only image-level labels. In this paper, we leverage text embeddings from a pre-trained vision-language model, specifically CLIP Hajimiri *et al.* (2024).

Our model (see Fig.3.2) consists of three modules, —a patch embedding backbone network and a compact head for localization and classification tasks. The backbone network is comprised

of the (i) *Encoder E*, a ViT (ViT-EVA-L) Fang, Wang, Xie & et. al (2023a) that decomposes images into patches and produces patch-level embeddings. It is pre-trained for classification and frozen during training. (ii) The *Decoder D* upscales these patch embeddings to a high spatial resolution where the p^{th} output patch is denoted as z_p . The CLIP text encoder is denoted as E_t . Moreover, we introduce a binary patch classifier $g(z_p)$ that predicts the FG/BG for each patch. Its response over all the patches forms a localization map M containing the FG object associated with the image class. This map provides the localization generated by our method. Furthermore, we define the classification scoring function $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which is parametrized with *frozen* class weight vectors $t_k \in \mathbb{R}^d$ for a class k for $k \in \{1, \dots, K\}$. An embedding vector $v \in \mathbb{R}^d$ is required. It computes its score for a class k via its dot product with the class anchor t_k : $f(v, t_k) = \langle v, t_k \rangle$. In this work, we consider t_k as the CLIP text embedding of the class k , while v could be a patch embedding z_p , or the global image embedding h of our method.

3.3.1 Key Components

The rest of this subsection introduces two important components of our TeD-Loc method – the generation of pseudo-labels of patches and the pre-processing of text embeddings.

Patch-level pseudo-label generation. To train our model we propose to leverage patch-level pseudo-annotation corresponding to FG and BG patches. However, since such annotations are not available in the WSOL setting, we consider an off-the-shelf pre-trained CAM-based classifier. Such models can yield a discriminative map to localize a target class which is adequate during training. Generally, any CAM-based model can be used Choe *et al.* (2020); Murtaza, Belharbi, Pedersoli & Granger (2025); Rony *et al.* (2023) to generate localization cues that can be leveraged for training. During the training of our model, we randomly sample few FG/BG patch locations for each training image at each training step to avoid overfitting. This pseudo-labelling technique has been shown to be effective in guiding learning Murtaza *et al.* (2023b); Belharbi *et al.* (2022c). In our experiments, we sample the same number of FG/BG locations to maintain a balanced ratio between the two classes. ω denotes the set containing the sampled patches for both FG/BG while ω^+ contains only the selected FG patches. $y'_p \in \{0, 1\}$ is used as the patch

pseudo-label where 0 is BG and 1 is FG. Additional details on this pseudo-labelling process are provided in suppl. materials.

Text embeddings orthogonalization. CLIP text embeddings of classes are used to distill localization knowledge as they provide a powerful link between visual and textual representations. However, they can sometimes conflate similar classes due to semantic overlap (e.g., “airplane” and “aircraft”) Wang & Kang (2024). This overlap between classes limits the benefit of those embeddings, especially when used in discriminative scenarios. To mitigate this issue, we propose to pre-process the class-text embeddings before using them in our method. We consider a transformation that projects all the text embeddings into a space where the distance between each pair of embeddings is maximum. In this work, we use orthogonal projection, in particular, QR orthogonalization GANDER (1980) and conserve the basis of the projection. In the rest of this paper, the orthogonalized version of text embeddings for class k are referred to as t_k . These new embeddings are kept frozen and play the role of class anchors that are carried with our model allowing us to discard the text encoder. We illustrate the issue of class overlap and the impact of orthogonalization over text embeddings in Fig.3.3.

3.3.2 The TeD-Loc Training Method

Text embedding distillation to local patches. Our objective is to distill text-class embeddings into patch embeddings and to discard the text encoder. Since text and global image embeddings in CLIP are not directly tied to patch embeddings, text embeddings cannot directly be used to localize objects. Patch embeddings are not necessarily correlated with text embeddings Hajimiri *et al.* (2024). In this work, we propose to create a direct link between text embeddings and patch embeddings for semantic localization. To achieve this, we propose to use knowledge distillation where we transfer text embedding of the image class to the patch embeddings allowing us to perform object localization. In particular, a contrastive learning loss Radford *et al.* (2021) is employed to ensure that FG patch embeddings are similar to text embedding of the image class while being dissimilar from embeddings of other classes. To this end, only the FG patches z_p where $p \in \omega^+$ are used for this loss. It can be simply defined through a standard cross entropy as

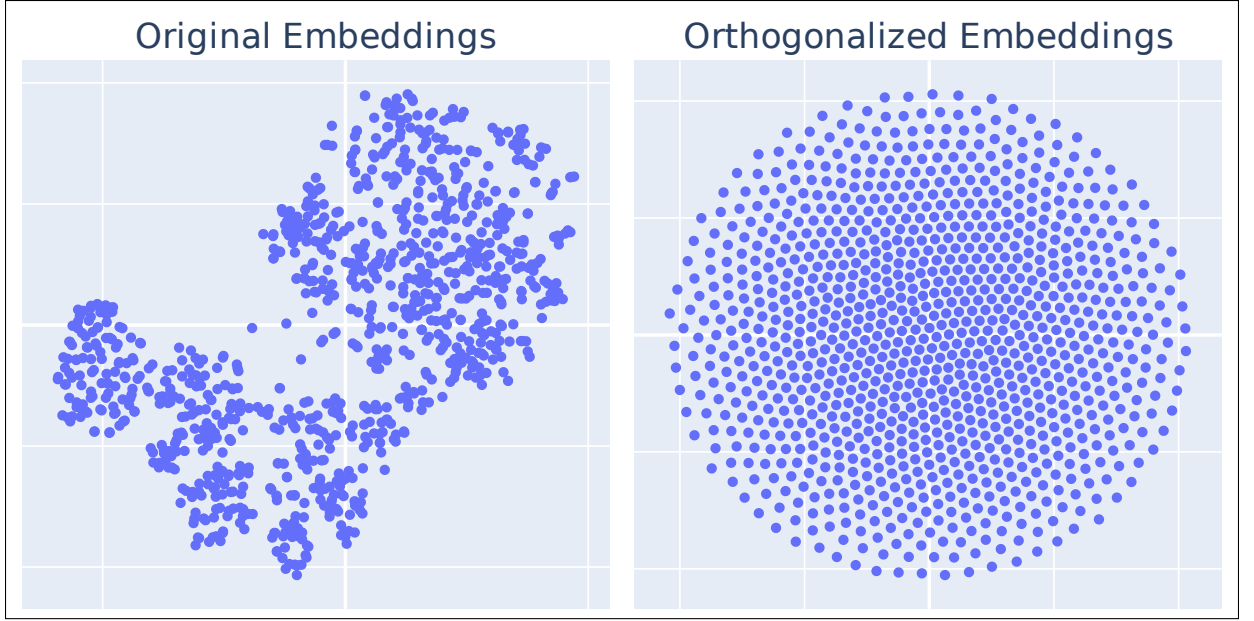


Figure 3.3 t-SNE visualizations of CLIP text embeddings for ILSVRC Deng *et al.* (2009) classes before and after orthogonalization. **(Left)** Prior to orthogonalization, embeddings of semantically similar classes (e.g., “airplane” and “aircraft”) cluster closely together, leading to potential confusion. **(Right)** After orthogonalization (QR decomposition), the embeddings are more uniformly distributed and orthogonal, reducing overlap.

follows Radford *et al.* (2021),

$$\mathbf{L}_{\text{KD}} = \sum_{p \in \omega^+} CE(y, f(z_p, t_y)). \quad (3.1)$$

Equation 3.1 is computed only on the selected FG patches ω^+ , as BG patches lack corresponding class embeddings, rendering them unsuitable for this contrastive loss. Since BG is not considered this can lead to poor localization as the BG region is present in most images. To mitigate this issue, we introduce a patch binary FG/BG classifier g which repels BG patch embeddings from FG text embeddings. It is trained using both FG/BG patches z_p and their pseudo-labels y'_p via standard cross-entropy loss,

$$\mathbf{P}_{\text{CL}} = \sum_{p \in \omega} CE(y'_p, g(\mathbf{z}_p)). \quad (3.2)$$

Minimizing this loss allows for both FG and BG regions to be present in the image which helps avoid imbalanced localization.

Global image embeddings from local patch embeddings for classification. So far, our method can only perform localization. To further allow it to perform image classification, our aligned FG patch embeddings are leveraged to construct global image embedding that describes the object in the image. This creates a reversed link from local patch representations to global image representation allowing to learn to classify the image. This aligns perfectly with our distillation approach from class text embedding to patch embeddings described previously, where we ensure that FG patch embeddings are correlated with the text embedding of the image class. Therefore, we leverage this property to construct a global image embedding h using all the patch embeddings and the patch classifier g as follows,

$$h = \sum_p a_p z_p \quad , \text{ where } a_p = g(z_p) / \sum_j g(z_j). \quad (3.3)$$

Eq.3.3 performs a weighted average of the embeddings for all the patches by giving more importance to patches that are classified as FG since their $g(z_p)$ will be closed to 1. In addition, BG patches are discarded since $g(z_p)$ close to 0. This effectively performs a differentiable selection of FG patches allowing for training with gradient-based methods. Most importantly, the final aggregated embedding h is expected to resemble the text embedding of the image class t_k . To furthermore ensure this, this embedding is trained to be as close as possible to t_k using standard cross entropy as follows,

$$\mathbf{I}_{\text{CL}} = \text{CE}(y, f(h, t_y)). \quad (3.4)$$

Overall training loss. Our overall training loss contains the three terms discussed previously: knowledge-distillation loss ($\mathbf{L}_{\text{KD}}$), patch classifier loss ($\mathbf{P}_{\text{CL}}$) and global image classification loss ($\mathbf{I}_{\text{CL}}$) as follows,

$$\mathbf{L} = \lambda_1 \mathbf{L}_{\text{KD}} + \lambda_2 \mathbf{P}_{\text{CL}} + \lambda_3 \mathbf{I}_{\text{CL}} , \quad (3.5)$$

where λ_1 , λ_2 and λ_3 are weighting factors that balance the contribution of each term. Stochastic Gradient Descent (SGD) is used for optimizing the parameters of our model which consist of the parameters of the decoder D_I and patch classifier g . By jointly optimizing these loss functions, our model learns to produce discriminative and well-aligned visual representations, enabling simultaneous classification and localization.

3.4 Results and Discussion

3.4.1 Experimental Methodology

Dataset. Two common challenging datasets were used for our WSOL experiments: (i) Caltech-UCSD birds-200-2011 (CUB) Wah, Branson, Steve & et. al (2011) consists of 11,788 images spanning 200 bird species. The dataset is partitioned into 5,994 training images and 5,794 testing images. For validation, an independent set of 1,000 images (five per class) collected by Choe *et al.* (2020) is utilized. (ii) ImageNet large-scale visual recognition challenge (ILSVRC) Deng *et al.* (2009) includes approximately 1.2 million training images and 10,000 validation images across 1,000 classes. We use the original validation split as our test set due to its sufficient size for robust evaluation. For validation purposes, ILSVRC-V2, collected by Recht, Roelofs, Schmidt & Shankar (2019) and annotated by Choe *et al.* (2020), is employed to mitigate biases toward the test set. For a fair comparison, we strictly adhere to the commonly used WSOL protocol proposed in Choe *et al.* (2020) for both datasets.

Evaluation measures. Following earlier work by Choe et al. Choe *et al.* (2020), we employ three localization measures alongside one classification measure to evaluate the proposed method. The localization measures are as follows: (1) **MaxBoxAcc** (referred to in previous work as **CorLoc** Deselaers, Alexe & Ferrari (2012) or **GT-Known** Singh & Lee (2017)), which quantifies the proportion of images for which the predicted bbox achieves an Intersection over Union (IoU) threshold of $\sigma = 50\%$ with the ground-truth bbox, independent of classification accuracy (CL); (2) **Top-1 localization accuracy (top-1 loc.)**, measuring the proportion of images where the model’s top predicted class is correct and the bbox IoU with ground truth exceeds $\sigma = 50\%$; and

(3) Top-5 localization accuracy (**top-5 loc.**), defined as the proportion of images for which the true class label is within the model’s top-five predictions and the bbox meets an IoU of $\sigma = 50\%$.

Implementation details. We closely followed the experimental setup of Choe et al. Choe *et al.* (2020), dataset splits, evaluation of localization maps across multiple thresholds, and training epochs—specifically, 50 epochs for the CUB dataset and 10 epochs for ILSVRC. Furthermore, our model is trained with a batch size of 32 and 16 for ILSVRC and CUB, respectively. In Eq.3.5, the hyperparameters λ_1 , λ_2 and λ_3 used in the total training loss (Eq.3.5) terms that are optimized over the values $(0, 1]$. Optimization of our model was performed using SGD, with a learning rate from $1e-6$ up to 0.01 . We also fine-tuned the weight decay and momentum parameters. Generated localization maps were evaluated at a resolution of 256×256 pixels.

Baseline methods. To extensively evaluate the performance of TeD-Loc, we compare against several recent state-of-the-art WSOL methods, including TS-CAM (Gao *et al.*, 2021), SCM (Bai *et al.*, 2022), LCTR (Chen *et al.*, 2022), C²AM (Xie *et al.*, 2022b), PSOL (Zhang *et al.*, 2020c), DiPS Murtaza *et al.* (2023b), CATR Chen, Ding, Cao & et. al (2023b), DA-WSOLZhu, She, Chen & et. al (2024), BAS Wu *et al.* (2022), and GenPrompt (Zhao *et al.*, 2023). Additionally, CLIP-ES Lin *et al.* (2023), which utilizes Grad-CAM to extract localization maps from CLIP was employed in a zero-shot setting. Specifically, we considered two variants of this method: CLIP-ES (GT-Known) and CLIP-ES (Pred). The CLIP-ES (GT-Known) variant requires ground-truth class labels during inference to generate localization maps. While this provides an upper bound on performance, it relies on privileged information unavailable in practical WSOL scenarios, thereby limiting its applicability. In contrast, CLIP-ES (Pred) depends solely on predicted class labels, aligning with the standard weakly supervised setting and offering a fair basis for comparison. This comprehensive evaluation enables us to demonstrate the robustness of our method across diverse settings.

Table 3.1 Performance comparison of TeD-Loc with state-of-the-art methods on the ILSVRC and CUB datasets. The metrics reported are MaxBoxAcc, top-1 loc., and top-5 loc.. The first two rows correspond to Grad-CAM for CLIP (?), where class labels are required to produce localization maps, making direct comparison unfair as it leverages privileged information.

TeD-Loc outperforms existing methods without relying on ground-truth class labels or thresholds derived from ground-truth data.

Method	Backbone		ILSVRC			CUB		
	CL	loc.	MaxBoxAcc	Top-1 loc.	Top-5 loc.	MaxBoxAcc	Top-1 loc.	Top-5 loc.
CLIP-ES (?) (cvpr,2023)	Text & Vision: ViT		74.2	43.0	62.9	92.8	36.3	67.2
TS-CAM (Gao <i>et al.</i> , 2021) (iccv,2021)	DeiT-S		67.7	53.4	64.3	71.3	83.8	87.7
SCM (Bai <i>et al.</i> , 2022) (eccv,2022)	DeiT-S		68.8	56.1	66.4	76.4	91.6	96.6
LCTR (?) (aaai,2022)	DeiT-S		68.7	56.1	65.8	92.4	79.2	89.9
PSOL (Zhang <i>et al.</i> , 2020c) (cvpr,2020)	DenseNet161	EfficientNet-B7	66.3	58.0	65.0	91.8	80.9	90.0
C ² AM (Xie <i>et al.</i> , 2022b) (cvpr,2022)	DenseNet161	EfficientNet-B7	68.5	59.6	67.1	92.9	81.8	91.1
GenPrompt (Zhao <i>et al.</i> , 2023) (iccv,2023)	Stable Diffusion	EfficientNet-B7	75.0	65.2	73.4	98.0	87.0	96.1
CATR Chen <i>et al.</i> (2023b) (iccv,2023)	DeiT-S		69.2	56.9	66.6	94.9	79.6	92.0
DA-WSOL Zhu <i>et al.</i> (2024) (pami,2024)	ResNet50		71.8	55.3	–	88.4	71.1	–
BAS ? (ijcv,2024)	InceptionV3		72.0	58.5	69.0	94.6	72.0	88.1
TeD-Loc (ours)	ViT-EVA-L		75.6	70.0	75.1	98.7	91.7	97.6

3.4.2 Comparison with State-of-the-Art

Quantitative results. TeD-Loc sets a new benchmark in WSOL by achieving SOTA results across different measures. On the ILSVRC dataset, TeD-Loc attains a MaxBoxAcc of 75.6%, Top-1 localization accuracy of 70.0%, and Top-5 localization accuracy of 75.1%, a significant improvement over existing methods. For instance, GenPrompt (Zhao *et al.*, 2023), which leverages Stable Diffusion and EfficientNet-B7, achieves a MaxBoxAcc of 75.0%, Top-1 localization accuracy of 65.2%, and Top-5 localization accuracy of 73.4%. Similarly, on the CUB dataset, TeD-Loc excels by achieving a MaxBoxAcc of 98.7%, Top-1 localization accuracy of 91.7%, and Top-5 localization accuracy of 97.6%. This surpasses the previous best method, GenPrompt (Zhao *et al.*, 2023), which reports a MaxBoxAcc of 98.0%, Top-1 localization accuracy of 87.0%, and Top-5 localization accuracy of 96.1%. The substantial gains on both datasets underscore the effectiveness of TeD-Loc in handling diverse and challenging scenarios.

The first two rows of Tab.3.1 pertain to CLIP-ES method (Lin *et al.*, 2023), which utilizes the CLIP model in a zero-shot setting. Notably, CLIP-ES (GT-Known) uses GT class labels to generate predictions. This approach yields a high MaxBoxAcc of 74.2% on ILSVRC and 92.8% on CUB, respectively. Moreover, TeD-Loc achieves a top-1 cls. accuracy of 89.9% on the ILSVRC dataset, significantly outperforming CLIP, which achieves 67.3%. Additionally, on the

CUB dataset, TeD-Loc attains a top-1 cls. accuracy of 93.0%, compared to CLIP’s 46.42%. This highlights the significance of learning from text embeddings and using them as anchors to compute classification scores. Furthermore, extended results are presented in suppl. materials.

It is interesting to observe that if we simplify the task for CLIP by using only the general “bird” class for all categories in the CUB dataset, the performance can increase by 4% when measuring MaxBoxAcc. However, this does not reflect the model’s ability to handle fine-grained classification and localization, which is essential for applications requiring precise object differentiation.

Qualitative results. Fig.3.4 presents a visual comparison of our method against recent state-of-the-art WSOL approaches, specifically GenPrompt Zhao *et al.* (2023) and TS-CAM Gao *et al.* (2021), on the ILSVRC dataset. While these methods yield competitive quantitative performance, they often struggle with accurately localizing complex objects. They tend to highlight irrelevant parts or even entirely different objects, especially in intricate scenes. GenPrompt, for instance, relies on CLIP’s discriminative and representative embeddings during inference, which necessitates class labels at test time using the external classifier. While this approach aims to adaptively focus on the object of interest, it can erroneously localize incorrect objects when the classifier predicts the wrong class. This dependency on class information during inference increases the chances of mistakes, especially in critical applications where precise localization without prior class knowledge is required. In contrast, our method consistently achieves high localization accuracy by effectively capturing both discriminative and non-discriminative regions of the target object. These localization maps can produce bboxes that encompass the entire object, enhancing both localization performance and interpretability. Unlike other methods that produce low-activation regions—resulting in bboxes over areas without meaningful content—our approach ensures that activations correspond closely with the actual visual appearance of the object.

Ablations. Tab.3.2 shows that the combination of our proposed loss functions: knowledge-distillation loss ($\mathbf{L}_{KD}$), patch classifier loss ($\mathbf{P}_{CL}$) and global image classification loss ($\mathbf{I}_{CL}$), is

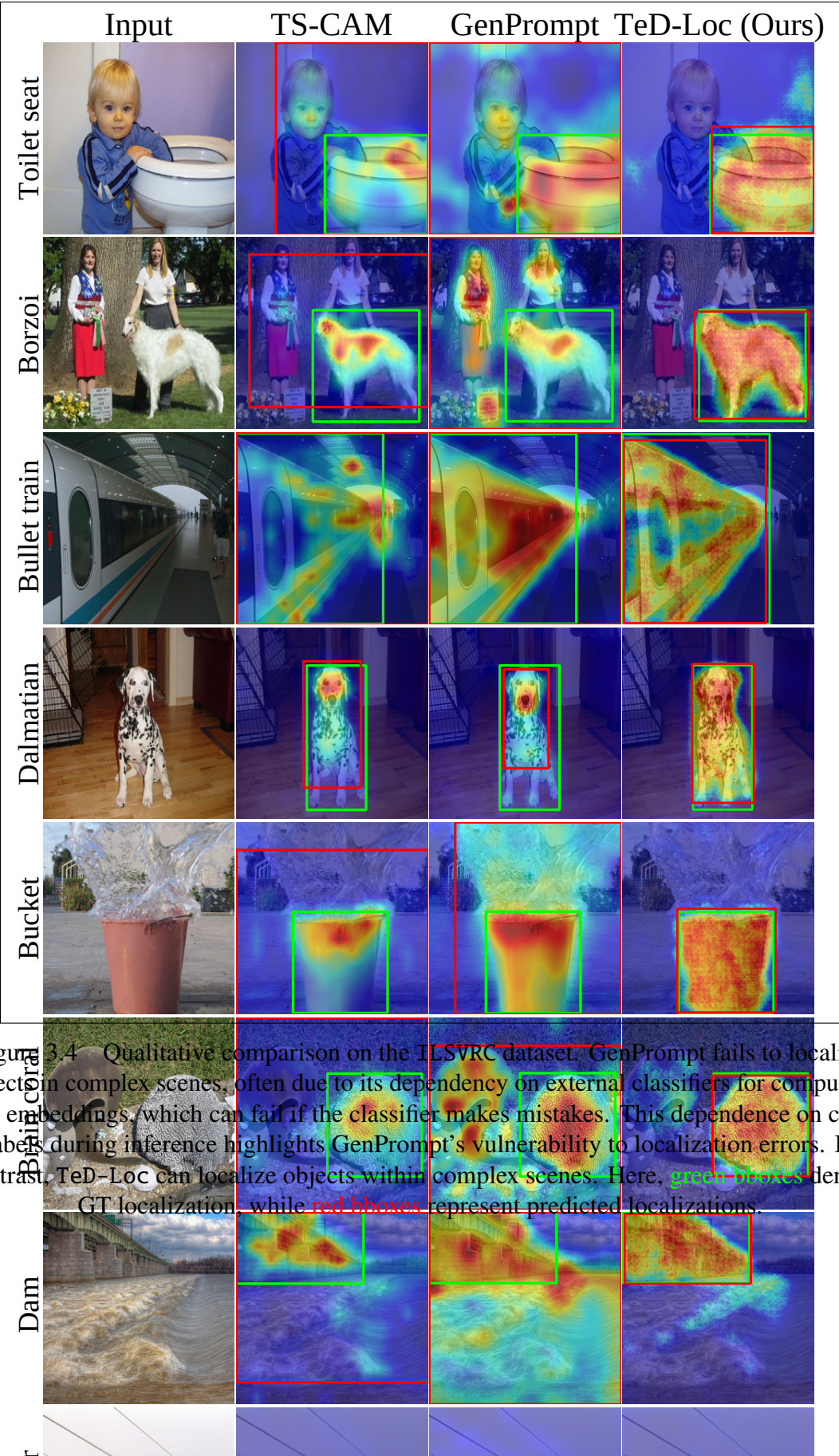


Figure 3.4 Qualitative comparison on the ILSVRC dataset. GenPrompt fails to localize objects in complex scenes, often due to its dependency on external classifiers for computing text embeddings, which can fail if the classifier makes mistakes. This dependence on class labels during inference highlights GenPrompt’s vulnerability to localization errors. In contrast, TeD-Loc can localize objects within complex scenes. Here, **green boxes** denote GT localization, while **red boxes** represent predicted localizations.

essential for achieving state-of-the-art localization performance. Using only the main loss L_{KD} , our method yields a **MaxBoxAcc** of 62.3%, **MaxBoxAcc** of 62.3%, as it focuses on learning FG embeddings without yielding explicit localization, thereby explaining its limited performance. Adding patch classifier loss significantly improves accuracy to 95.4% by enhancing FG/BG separation. Finally, incorporating image class loss boosts the performance to 98.7%, emphasizing the importance of discriminative learning to distinguish between correct and incorrect class alignments. This demonstrates the effect of integrating these losses to achieve highly accurate weakly supervised object localization.

Table 3.2 Ablation study on the CUB dataset showing the impact of different loss combinations on **MaxBoxAcc** performance.

Losses	CUB (MaxBoxAcc)
L_{KD}	62.3
$L_{KD}+P_{CL}$	95.4
$L_{KD}+P_{CL}+I_{CL}$	98.7

3.4.3 Complexity Analysis

To show the efficacy of our proposed method, we presented a detailed complexity analysis (Tab.3.3) in comparison with two recent WSOL approaches: TS-CAM Gao *et al.* (2021) and GenPrompt Zhao *et al.* (2023). Although GenPrompt yields SOTA localization performance compared to TS-CAM, it introduces significant computational overhead. To deal with this issue, our method offers a computationally efficient model without sacrificing localization accuracy. TS-CAM utilizes a transformer-based architecture to generate class activation maps directly from image features. It maintains a relatively modest computational footprint, with 25.12 million parameters and an inference time of 19.04ms. Conversely, GenPrompt employs a generative framework involving diffusion models and CLIP embeddings to generate LOC maps, resulting in a considerable increase in complexity due to the multiple-modules pipeline and iterative nature of the diffusion process. Specifically, it comprises an EfficientNet-based classifier for label

prediction (66.35 million parameters), a variational autoencoder (VAE) for latent embeddings (83.65 million parameters), and CLIP E_T for discriminative and representation embeddings, having 123.83 million parameters. The diffusion model, utilizing a U-Net architecture, adds 859.5 million parameters and iterates across 100-time steps, drastically increasing computation time. Our proposed method thus emerges as a significantly more efficient alternative, achieving robust performance without the exorbitant computational cost characteristic of GenPrompt. Furthermore, for fair and consistent inference time measurement, we utilized an idle machine equipped with an NVIDIA-A100 GPU. We first conducted 20 warm-up epochs with a batch size of 1, followed by 1,000 inference steps, and calculated the average inference time across these steps.

Table 3.3 Computational complexity and localization performance of our proposed TeD-Loc against GenPrompt.

Methods	Complexity Analysis	
	# Para.	Infer. Time
GenPrompt Zhao <i>et al.</i> (2023)	1133.35M	272ms
TeD-Loc (ours)	569.67M	121ms

3.5 Conclusion

We have introduced TeD-Loc, a novel approach that integrates textual and visual modalities by transferring knowledge directly from CLIP’s text embeddings to our patch embedding module. This alignment enables our model to localize objects at the patch-level while simultaneously performing classification. TeD-Loc exemplifies how the fusion of language and visual information can help in WSOL, demonstrating that classification can be achieved through localization, we discover that bridging modalities is not only feasible but also beneficial for complex vision tasks. Future research could explore deeper integrations of multimodal embeddings and extend this framework to other domains requiring weak supervision.

3.6 Supplementary material

This supplementary material contains the following content:

1. Sampling FG/BG regions for pseudo-label generation.
2. Extended results:
 - a) Patch-level Localization with text-anchors.
 - b) Impact of orthogonalization on results.
 - c) Extended complexity analysis.

3.6.1 Sampling FG/BG Regions for Pseudo-label Generation

To train our model, we employ pseudo-labels for FG and BG regions following recent methods Belharbi *et al.* (2022c); Murtaza *et al.* (2023a,b). We obtain these pseudo-labels by utilizing CAM $C \in \mathbb{R}^{H \times W}$ extracted from a pre-trained discriminative model, specifically using Grad-CAM Lin *et al.* (2023) from CLIP Radford *et al.* (2021). These CAMs indicate regions of the image highlighting an object belonging to the FG class, and we use them to guide the sampling of FG and BG regions. We first apply Otsu’s thresholding method Otsu (1979) to the CAM C to segment FG and BG regions. This method automatically determines a threshold, effectively separating high-activation regions (FG) from low-activation ones (BG). We denote the set of pixel locations in the image domain as Ω .

For FG sampling, we focus on regions with high activation values in C . Specifically, we select the top n^+ pixels with the highest activation values inside the image, forming the set of potential FG locations $\omega_+ \subset \Omega$. We then randomly sample a subset of these locations to be used as FG samples during training. For BG sampling, we consider regions with low activation values in C . We select the bottom n^- pixels with the lowest activation values, excluding any pixels that are within the FG regions. This forms the set of potential BG locations $\omega_- \subset \Omega$. We randomly sample from these locations to obtain BG samples for training. We note by $\omega = \omega_+ \cup \omega_-$ the set of all sampled pixels in both FG and BG in one sampling step.

To ensure that our model generalizes well and avoids overfitting, we perform this sampling process at every training step for each image. This dynamic sampling allows the model to explore different regions of the image during training, promoting robustness and consistency in learning.

The sampled FG and BG pixel locations are used to create a partial pseudo-label mask $y'_p \in \{0, 1\}$, where $y'_p = 1$ for FG pixels, $y'_p = 0$ for BG pixels, and locations with unknown labels are left undefined. The set of sampled locations p is defined by $p \in \omega$.

3.6.2 Extended Results

This section provides additional results to gain further insight into our method.

3.6.2.1 Patch-level Localization with Text Anchors

In standard CLIP model Radford *et al.* (2021), text class embedding is not necessarily correlated with the local vision patch embedding. To show this, we conduct the following experiment: consider the text embedding of the image class label y : t_y . Then, to localize this class object within the image, we perform a dot product across all patch embeddings: $\langle z_p, t_y \rangle$ where $z_p, \forall p \in \Omega$. High scores at location p should indicate the high likelihood of the object y presence at this location. The obtained score map is then considered as a CAM. We perform this experiment over three variants of CLIP model: Vanilla CLIP Radford *et al.* (2021), SCLIP Wang *et al.* (2023), and NACLIP Hajimiri *et al.* (2024), in addition to our method. The obtained results are presented in Tab.3.4. Vanilla CLIP yielded poor results confirming that class text embeddings are not necessarily correlated with the local patch embeddings making them less useful for this task. This justifies using a gradient-based method over the dot product score between the global image embedding and a class text embedding in CLIP-ES Lin *et al.* (2023). However, we notice a greater improvement in localization for the next recent CLIP variants SCLIP Wang *et al.* (2023), and NACLIP Hajimiri *et al.* (2024). However, their **Top-1**, and **Top-5 loc.** are still low indicating poor classification performance. On the other hand, our

method achieves the highest performance over the three metrics indicating better localization and classification scores over both datasets. This is the result of our text-to-patch distillation, which ensures that local patch embeddings are correlated with the class text embedding, allowing direct localization based on the text embedding. This equips our method with a secondary localization approach and the patch FG/BG classifier g . This second localization approach yielded relatively better performance than when using g . Fig. 3.5 visualizes this localization strategy of different CLIP variants and our method.

Table 3.4 Localization performance via (patch, class) embeddings dot product: $\langle z_p, t_y \rangle$ where $z_p, \forall p \in \Omega$ is the patch embeddings, and y is the true image class. We report localization performance (**MaxBoxAcc**, **Top-1 loc.**, **Top-5 loc.**) using different variants of CLIP, and our method.

Localization via $\langle z_p, t_y \rangle$	CUB			ILSVRC		
	MaxBoxAcc	Top-1 loc.	Top-5 loc.	MaxBoxAcc	Top-1 loc.	Top-5 loc.
Vanilla CLIP Radford <i>et al.</i> (2021)	18.8	8.9	15.2	41.1	26.6	37.0
SCLIP (CoRR'23) Wang <i>et al.</i> (2023)	85.8	14.4	37.1	70.4	33.9	55.1
NACLIP (wacv'25) Hajimiri <i>et al.</i> (2024)	80.8	12.0	32.3	71.7	28.6	49.4
TeD-Loc (ours)	98.7	92.0	97.5	77.18	71.8	76.7

3.6.2.2 Impact of Orthogonalization on Performance

Tab. 3.5 shows the impact of orthogonalization of class text embedding in our method for both tasks: classification and localization. These results suggest that original class text embeddings of CLIP Radford *et al.* (2021) are not well adequate to perform discriminative learning as these embeddings overlap as shown visually in the main paper (Fig.3). However, their orthogonalization allows better separation of these embeddings making them more suitable for classification task. This also positively affects localization as well in our method since both tasks are strongly related by design.

Table 3.5 Impact of class text embeddings (text anchors) orthogonalization over localization and classification performance in our method over **CUB** dataset.

Method	CUB			
	MaxBoxAcc	cls.	Top-1 loc.	Top-5 loc.
w/o orthogonalization (default anchors)	97.73	56.00	54.84	85.79
w/ orthogonalization	98.7	93.0	91.7	97.6

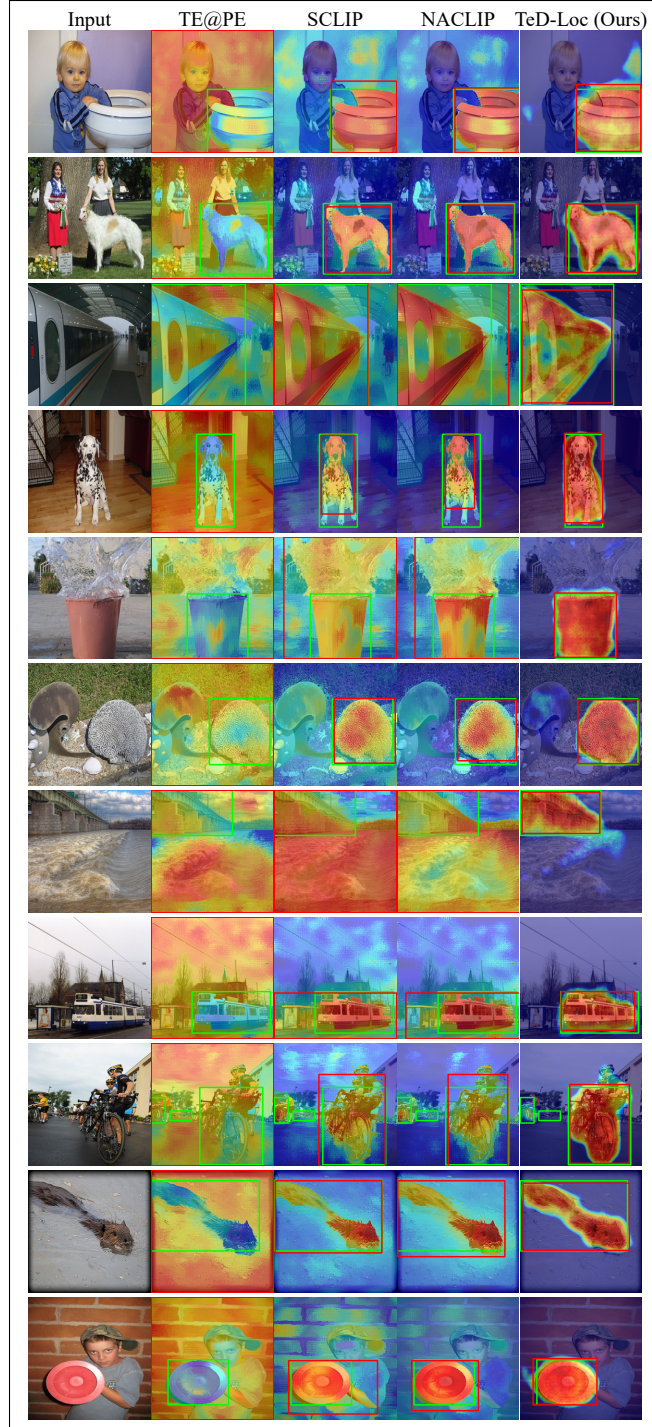


Figure 3.5 Visualization of localization map defined via (patch, class) embeddings dot product: $\langle z_p, t_y \rangle$ where $z_p, \forall p \in \Omega$ is the patch embeddings, and y is the true image class over different variants of CLIP, and our method. TE@PE is the vanilla CLIP Radford *et al.* (2021) where TE is the text embedding, and PE is the patch embedding.

CHAPTER 4

A REALISTIC PROTOCOL FOR EVALUATION OF WEAKLY SUPERVISED OBJECT LOCALIZATION

Shakeeb Murtaza^a, Soufiane Belharbi^a, Marco Pedersoli^a, Eric Granger^a

^aDepartment of Systems Engineering, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*,
February 2025

Abstract Weakly Supervised Object Localization (WSOL) allows training deep learning models for classification and localization (LOC) using only global class-level labels. The absence of bounding box (bbox) supervision during training raises challenges in the literature for hyper-parameter tuning, model selection, and evaluation. WSOL methods rely on a validation set with bbox annotations for model selection, and a test set with bbox annotations for threshold estimation for producing bboxes from localization maps. This approach, however, is not aligned with the WSOL setting as these annotations are typically unavailable in real-world scenarios. Our initial empirical analysis shows a significant decline in LOC performance when model selection and threshold estimation rely solely on class labels and the image itself, respectively, compared to using manual bbox annotations. This highlights the importance of incorporating bbox labels for optimal model performance. In this paper¹, **a new WSOL evaluation protocol** is proposed that provides LOC information without the need for manual bbox annotations. In particular, we generated noisy pseudo-boxes from a pretrained off-the-shelf region proposal method such as Selective Search, CLIP, and RPN for model selection. These bboxes are also employed to estimate the threshold from LOC maps, circumventing the need for test-set bbox annotations. Our experiments² with several WSOL methods on challenging natural and medical

¹ **Our scope.** This work focuses on WSOL (Belharbi *et al.*, 2022c; Murtaza *et al.*, 2023b,a; Murtaza, Belharbi, Pedersoli, Sarraf & Granger, 2022), as opposed to other related tasks such as weakly supervised detection, segmentation, or instance segmentation, which are often mixed in earlier works Shi & Ferrari (2016); Cinbis, Verbeek & Schmid (2014); Deselaers, Alexe & Ferrari (2010).

² Our code and generated pseudo-bounding boxes can be accessed at github.com/shakeebmurtaza/wsol_model_selection.

image datasets show that using the proposed pseudo-bboxes for validation facilitates the model selection and threshold estimation, with LOC performance comparable to models selected using GT bboxes on the validation set and threshold estimation on the test set. It also outperforms models selected using class-level labels, and then dynamically thresholded with only LOC maps.

4.1 Introduction

Methods for WSOL provide efficient learning strategies (Bai *et al.*, 2022; Gao *et al.*, 2021; Wu *et al.*, 2023) to mitigate the cost of annotating datasets with bboxes. Given only coarse class-labels, a deep learning model can be trained to perform image classification (*cls.*) tasks while yielding the spatial image region of the object (localization). Class-Activation Mapping (CAM) (Choe *et al.*, 2020; Rony *et al.*, 2023) is the dominant WSOL approach that builds a spatial heatmap to localize an object.

Despite its popularity, WSOL still faces several challenges, in particular with its evaluation protocol (Choe *et al.*, 2020). We argue that the common protocol presently used in the literature may not be reliable for real-world applications limiting WSOL benefits. WSOL methods

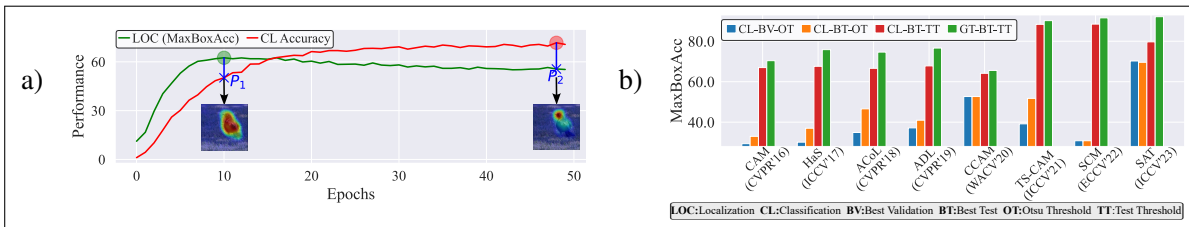


Figure 4.1 (A) Example of LOC (MaxBoxAcc) and *cls.* accuracy w.r.t the number of epochs on the CUB validation set in a WSOL setting, where only class-labels are used to train the model. These curves show that LOC and *cls.* tasks are loosely correlated where convergence is reached at different training epochs. Typically, high LOC performance is achieved early in the training, followed by degradation. However, the classifier takes a longer time to converge. Selecting the best model for LOC may need a LOC annotation in the validation set. (B) Significant bias arises when using GT bounding boxes and test set thresholds, leading to overestimated LOC accuracy, as bbox annotations are typically absent in real-world scenarios. In a realistic WSOL scenario (CL-BV-OT), where model selection is based on *cls.* accuracy and OT over the validation set, performance declines considerably

should only require class-labels for supervision. During training (with held-out validation sets), the model should not have access to data with manual localization (LOC) supervision, i.e., ground truth (GT) bounding box (bbox) annotations. Model selection and hyper-parameter search are therefore very challenging (Fig.4.1a). Early works observed the test performance for hyper-parameter selection leading to an overestimation of model performance (Choe & Shim, 2019; Singh & Lee, 2017; Zhang *et al.*, 2018c)³. As illustrated in Fig.4.1b, the test-set LOC performance of these methods may decline in real-world scenarios, where only class-labels are available.

Choe *et al.* (2020) proposed an improved evaluation protocol to address some of the above limitations. To perform model selection, they propose using a held-out validation set with full manual annotations (class labels and bboxes). Although this protocol allows us to avoid using the test set and to perform unbiased modal evaluations, it is somewhat unrealistic since annotated bboxes are not available in practice. Having access to such manual LOC supervision during training goes against the WSOL setting, and can overestimate model performance (Fig.4.1b). Moreover, the same authors (Choe *et al.*, 2020) have shown that a model directly trained on those annotated images would provide higher LOC accuracy than any weakly supervised approach. Although authors annotated a limited number of images per class to create a held-out validation set, the total cost of annotation adds up with the number of classes. For instance, they annotated 10 images/class for the ILSVRC dataset, which amounts to a total of 10,000 images. Given the cost, these fully annotated images can, for example, be directly used to fit the model weights in a semi-supervised learning (SSL) setting rather than being used for model selection in a WSOL setting.

Another unrealistic practice prevalent in the literature is the use of GT bboxes from the test set to estimate the threshold for binarizing the LOC map to generate bboxes (Choe *et al.*, 2020). However, this practice leads to an overestimation of model performance (LOC accuracy) because

³ Refer to the official implementation (Choe & Shim, 2019; Singh & Lee, 2017; Zhang *et al.*, 2018c) that is publicly available. Unfortunately, even recent works after (Choe *et al.*, 2020) still use the test set as a held-out validation set (Wu *et al.*, 2023) as shown in their public code. This practice potentially biases the selection process in favor of LOC performance.

bbox annotations are typically absent in real-world scenarios, making it impossible to estimate the threshold during inference. Models selected using GT bboxes and test set thresholds may exhibit significantly higher LOC accuracy than models selected based on *cls.* accuracy and thresholds automatically estimated using the Otsu thresholding (OT) algorithm, for example, as one would do when bbox annotations are unavailable on the validation. An example of the inflated performance resulting from using the test set for model selection and threshold estimation is illustrated in Fig.4.1b, where the LOC accuracy is reported for various WSOL methods on the CUB test set using two model selection criteria: *cls.* and LOC accuracy. Additionally, the results of different thresholding methods are reported for LOC maps used to generate bboxes. Additionally, we presented the test LOC accuracy when the model was selected with the best *cls.* accuracy on the validation or test set, and when automatic thresholding algorithm (OT) was used for producing bboxes. In a more realistic evaluation protocol, where model selection is based on *cls.* accuracy and the OT on the validation set (blue bar Fig.4.1b), a noticeable decrease in performance is observed. This assessment highlights the necessity for unbiased thresholding and model selection to evaluate the real-world capabilities WSOL of methods.

This paper provides a realistic evaluation protocol for WSOL that can be applied in real-world scenarios. Instead of assuming that manual GT bbox annotations are available for validation, we estimate pseudo-bboxes by leveraging an off-the-shelf pretrained model that can generate region-of-interest (ROI) proposals (Hosang, Benenson & Schiele, 2014) such as Contrastive Language–Image Pre-training (CLIP) (Lin *et al.*, 2023) (Fig.4.3). This model is pretrained on generic datasets requiring less supervision, such as class-level labels. We also consider Selective Search (SS) (Uijlings, Sande, Gevers & Smeulders, 2013), an unsupervised method that does not require training, and Region Proposal Networks (RPN) (Ren *et al.*, 2015), trained on large datasets with class-agnostic bboxes. They can generate multiple bboxes with objectness scores for ranking. The impact of using these different levels of supervision is explored for pretrained proposal generators. Given a set of proposed bboxes for an image, and using pointing game analysis (Zhang *et al.*, 2018a) and class-level labels, the most discriminative bboxes are selected and ranked using classifier response to produce the final bbox. This approach allows to create

pseudo-annotated bboxes for the held-out validation set, and therefore, avoids manual annotation, leading to more realistic model selection. Despite their lower accuracy, these pseudo-bboxes can effectively select models and enable the estimation of thresholds using the validation set for generating bboxes from LOC maps in the test set.

To address the biased estimation obtained with current approaches, a threshold is estimated using the validation-set pseudo-boxes and subsequently applied to the test set to convert LOC maps into bboxes. Estimating the threshold from the validation set enables WSOL methods to achieve realistic and competitive performance in real-world scenarios. This avoids estimating the threshold from the test set, which leads to an overestimation of performance, and using OT which leads to low performance (see Fig.4.1b).

Our main contributions are summarized as follows.

- (1) An initial experimental analysis of state-of-the-art WSOL methods on the CUB and ILSVRC datasets showing the impact on the test-set performance of bbox supervision in the held-out validation set. Results indicate that utilizing only class-labels for validation leads to poor test-set LOC accuracy, highlighting the importance of using LOC cues for model selection. Conversely, using manually annotated bboxes for validation improves LOC accuracy, leading to an overestimated and unrealistic performance evaluation. Further experiments evaluating the use of less precise pseudo-bbox supervision in the validation set indicate that noisy bboxes can still effectively guide model selection.
- (2) A realistic evaluation protocol is proposed for WSOL. First, we propose to estimate pseudo-bboxes without manual intervention and use only class-labels to generate annotations for a held-out validation set. Given different application scenarios, we explored three different bbox generation strategies based on off-the-shelf models: SS (Uijlings *et al.*, 2013), RPN (Ren *et al.*, 2015), and CLIP (Lin *et al.*, 2023). Secondly, these pseudo-bboxes generated for the validation set allow for estimation of the threshold, which can be subsequently applied to test images for producing bboxes on the LOC maps.
- (3) An extensive set of experiments was conducted with prominent WSOL methods using our new protocol on the CUB and ILSVRC datasets for model selection and threshold estimation to convert

localization maps into bboxes. To assess the generalizability of our model, our analysis was extended to include GLaS and CAMELYON medical image data (see the supplementary material). Results show that models selected and thresholds estimated using pseudo-bboxes achieve a level of accuracy consistent with models selected using GT bboxes (currently-used but unrealistic selection protocol). Furthermore, using a single threshold estimated from pseudo-boxes reduced the threshold search space from 1,000 values to just one.

4.2 Related Work

In supervised learning, hyper-parameter and model selection typically rely on a fully annotated validation set. Methods like early stopping leverage this set for realistic estimation of model performance on unseen test data (Finnoff, Hergert & Zimmermann, 1993; Lodwich, Rangoni & Breuel, 2009; Morgan & Bourlard, 1989) since the three sets (train, val and test) employ the same supervisory signal and perform the same task. This section reviews model selection protocols in different settings, highlighting issues in model selection and hyperparameter search. It reviews early stopping methods without a validation set and concludes with model selection in WSOL.

(a) Protocols with limited supervision. Beyond the fully supervised setting, other learning settings have been derived with their protocol. This typically concerns learning with less annotated data (Cheplygina, de Bruijne & Pluim, 2019; Zhou, 2018). This may affect the availability of annotations over the held-out set used for model selection and hyper-parameter search. Despite the efforts of the community to design standardized protocols, unrealistic practices may emerge. Different attempts have been made to ensure that such protocols are realistic to benefit from the unbiased application in real-world scenarios.

In the SSL setting, researchers have investigated whether exiting works can perform well on realistic benchmarks compared to standard benchmarks used by the community (Oliver, Odena, Raffel & et. al, 2018; Su, Cheng & Maji, 2021). The authors argue that standard SSL benchmarks fail to reflect real-world challenges such as sensitivity to unlabeled data size, distribution shifts,

class imbalance, and novel classes, and reserving some labeled samples as a held-out validation set reduces the training set size.

Unsupervised domain adaptation (UDA) is another learning setting with less annotation that raises several questions about how realistic is the protocol. UDA involves adapting a source model on labeled source data and unlabeled target data. Model selection and hyper-parameter search are challenging, with recent works improving evaluation protocols (Dinu, Holzleitner, Beck & et. al, 2023; Ericsson, Li & Hospedales, 2023; Saito, Kim, Teterwak & et. al, 2021; Salvador, FATRAS, Mitliagkas & Oberman, 2022; Yang, Qian, Xu & Xie, 2023; You, Wang, Long & Jordan, 2019). For instance, Ericsson et al. (Ericsson *et al.*, 2023) show that using an annotated target set for hyper-parameter search leads to largely overestimated and unrealistic performance. They proposed using source data, evaluating another UDA model such as InfoMax (Shi & Sha, 2012), or a labeled held-out target set. Yang et al. Yang *et al.* (2023) introduced the Transfer Score for UDA model evaluation for hyper-parameter search and early stopping. The protocol issue is much more severe in the case of source-free UDA (Fang, Yap, Lin, Zhu & Liu, 2023b) where the user cannot access the source data.

Self-supervised (Caron *et al.*, 2021), unsupervised learning (Chen & Cummings, 2023; Vaithyanathan & Dom, 1999), and data generation (Borji, 2019) also face model selection and hyper-parameter search difficulties due to limited supervision. For instance, in image generation, quantitative measures (Heusel, Ramsauer, Unterthiner & et. al, 2017; Salimans, Goodfellow, Zaremba & et. al, 2016; Zhang, Isola, Efros & et. al, 2018b) may not fully capture how realistic a generated image is, necessitating visual inspections that affect reproducibility and real-world applicability.

The aforementioned works show that the evaluation protocol is an issue for many settings. This work focuses on realistic WSOL evaluation protocols, in particular model selection for hyperparameter optimization and threshold estimation using test set for generating bboxes.

(b) Early stopping without validation set. Some works have explored early stopping without a validation set, crucial when data is limited. Large held-out validation sets improve gener-

alization estimates but reduce training data size at the cost of significantly reducing the data available for the training set. Mahsereci et al. (Mahsereci, Balles, Lassner & Hennig (2017) proposed a criterion using gradient statistics over the training set, effective in classification and regression tasks. Duvenaud et al. (Duvenaud, Maclaurin & Adams (2016) used weight statistics, monitoring the entropy change in network parameters as an indicator of generalization. Yuan et al. (Yuan, Feng & Liu (2024) introduced *Label Wave*, tracking model prediction changes on noisy training labels to prevent overfitting. Li et al. (Li, Geng & Chen (2020) suggested using augmented training data instead of a real validation set to reduce computational costs. While these works aim to halt the training of the entire model, Bonet et al. (Bonet, Ortega, Hidalgo & Shekkizhar (2021) per-class channel early stopping for CNNs without validation sets, leveraging DeepNN (Shekkizhar & Ortega (2020), a non-negative kernel regression. Note that in these works, the training and test task is the same. In WSOL, however, the model is trained for the `cls.` task but it is evaluated on the LOC task.

(c) Protocols for WSOL. Before the work of Choe et al. (Choe *et al.*, 2020), bbox annotations were not considered to create the held-out validation set. Therefore, practitioners may unintentionally inspect the test LOC performance, leading to unrealistic WSOL (Choe & Shim, 2019; Singh & Lee, 2017; Zhang *et al.*, 2018c). To alleviate this, Choe et al. (Choe *et al.*, 2020) investigated several aspects of WSOL protocol. They mainly showed that state-of-art methods (at that time) do not improve accuracy beyond that of the baseline CAM method (Zhou *et al.*, 2016) after searching for the optimal decision threshold. They proposed a concise WSOL evaluation protocol, including new metrics and a held-out validation with bboxes. Although their work largely improved the WSOL protocol, it is somewhat unrealistic since manually-labelled bboxes are not available in real-world applications of the WSOL setting. Strong supervision is used to select a model, which implicitly helps to select a model with the best LOC performance which can contribute to overestimating LOC accuracy on the test set. In addition, it has been reported (Choe *et al.*, 2020) that using manually annotated bboxes for model training on the held-out validation set, rather than class-labels on the training set, can outperform baseline WSOL methods. Furthermore,

Choe et al. (Choe *et al.*, 2020) also employ the test set for estimating the threshold needed to convert the localization map to bboxes, rendering WSOL setting unrealistic.

Although more realistic protocols have been developed for other learning settings like SSL and UDA, WSOL represents a challenging setting because of the different tasks (cls. and LOC) performed with the same model during training vs testing. The supervision available during training allows us to select models for the cls. task. At test time, the model is evaluated additionally over LOC. The disconnection between available annotation at training time, and the required task at test time creates a challenging situation for model selection. Furthermore, §4.3 provides a more realistic evaluation protocol within the WSOL framework.

4.3 A Realistic Protocol for Model Evaluation

In this section, we describe a new protocol for evaluating WSOL methods. First LOC accuracy is shown to be overestimated when using manual bbox supervision for the held-out validation set. To address this, automatically generating pseudo-bboxes is proposed for the validation set using only class-label supervision, eliminating the need for manual annotation. It involves generating pseudo-bboxes for the validation set using off-the-shelf ROI proposal generators and utilizing these pseudo-bboxes to estimate a threshold. Then this can be applied to localization maps in the test set for producing bboxes, providing a more systematic and realistic evaluation protocol.

4.3.1 Model Selection

Consider a WSOL training set $\mathcal{D}_{train} = \{(x_i, y_i, \cdot)\}$ composed of samples with x_i the input image, y_i its class-label, and $\cdot$ is its missing bbox supervision. Let us define $\mathcal{D}_{val} = \{(x_i, y_i, \cdot)\}_{i=1}^N$ ⁴, and $\mathcal{D}_{test} = \{(x_i, y_i, o(x_i))\}$ as the held-out validation and test sets, respectively, where $o(x_i)$ is the oracle manual bbox annotation for the sample x_i . In the WSOL setting, bbox supervision is unavailable for any dataset. However, in a laboratory, the test set contains them to evaluate LOC performance of a model.

⁴ In Choe et al. (Choe *et al.*, 2020) the validation set contain $o(x_i)$.

The WSOL setting involves training a model $f(\cdot; \theta)$ with parameters θ to correctly classify an input image while yielding LOC boxes. Training is performed on $\mathcal{D}_{train}$ that lacks LOC supervision. In this paper, we introduce a pseudo-annotator $\hat{o}$ that automatically annotates an image with pseudo-bboxes (without human intervention) and only requires the class-labels. $\hat{o}$ yields noisy bboxes, providing LOC pseudo-bbox annotations $\hat{o}(x_i)$ that are less accurate than oracle bbox annotations $o(x_i)$. Moreover, we redefine the validation set $\mathcal{D}_{val}$ to incorporate pseudo-bboxes $\hat{o}(x_i)$, yielding $\mathcal{D}_{val} = \{(x_i, y_i, \hat{o}(x_i))\}_{i=1}^N$.

Let $f(x; \theta)$ be the model's cls. prediction, while $f(x; \theta)^b$ is the predicted bbox for image x . We denote by $\ell(\cdot, \cdot)$ a LOC similarity measure between two bboxes. Following Choe et al. (Choe *et al.*, 2020), $\ell(\cdot, \cdot)$ is commonly `MaxBoxAcc` or simply the Intersection-Over-Union (IoU) measure. Let us define a hypothetical ideal scenario where one can have access to the oracle for fully annotated bboxes $o(x_i)$ in the validation set $\mathcal{D}_{val}$ allowing, for instance, to perform early stopping and threshold selection. Therefore, we can define the average LOC performance of the model f with a specific parameter value of θ over this validation set as,

$$\mathcal{M}(\theta, \mathcal{D}_{val})_o = \frac{1}{N} \sum_i^N \ell(f(x_i; \theta)^b, o(x_i)). \quad (4.1)$$

Similarly, the same measure over $\mathcal{D}_{val}$ can be defined using our pseudo-annotator $\hat{o}$ as,

$$\mathcal{M}(\theta, \mathcal{D}_{val})_{\hat{o}} = \frac{1}{N} \sum_i^N \ell(f(x_i; \theta)^b, \hat{o}(x_i)). \quad (4.2)$$

Eqs. 4.1 and 4.2 provide an assessment of the model LOC accuracy over a held-out $\mathcal{D}_{val}$. Eq.4.1 is accurate, while Eq.4.2 accounts for errors due to inaccurate LOC annotation. Since practitioners do not have access to the oracle bboxes $o(x_i)$, the measure $\mathcal{M}(\theta, \mathcal{D}_{val})_o$ is not realistic. When one has access to oracle $o(x_i)$ bboxes for training, it is better to perform SSL to directly fit the model weights instead of WSOL (Choe *et al.*, 2020). However, one can still assess $\mathcal{M}(\theta, \mathcal{D}_{val})_{\hat{o}}$ for an approximate but realistic estimation of the model's LOC accuracy. The latter measure provides a realistic assessment of LOC accuracy for WSOL. A direct application of our proposed

noisy measure $\mathcal{M}(\theta, \mathcal{D}_{val})_{\hat{o}}$ in the WSOL setting is early stopping for model selection, and hyper-parameter search. In these applications, the measure must follow a similar behaviour to $\mathcal{M}(\theta, \mathcal{D}_{val})_o$, as shown in Fig.4.2.

Pseudo-bbox annotation. For generation of pseudo-bboxes (using $\hat{o}$), we leverage pretrained of-the-shelf region proposal models to generate LOC annotations for a held-out validation set $\mathcal{D}_{val} = \{(x_i, y_i, \hat{o}(x_i))\}_{i=1}^N$. This set is originally labeled with class-labels only. These models can be unsupervised without the need for training or require weak supervision for training. We consider the three following scenarios of the pretrained model depending on its level of supervision.

(a) *Unsupervised*: In this case, the model does not have access to any supervision. Typical examples are conventional region proposal methods (Hosang *et al.*, 2014) that are generally used to build pseudo-supervision to train object detector models (Meethal, Pedersoli, Zhu,

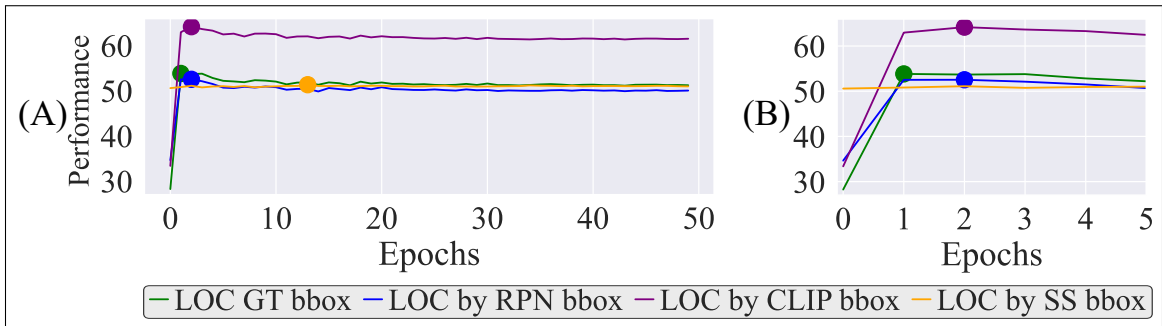


Figure 4.2 Model selection with early stopping at the epoch indicated with a dot using LOC accuracy. Different approaches are compared for pseudo-bbox annotation versus GT annotations (oracle). Fig(B) is a zoom of Fig.(A) between epochs 0 and 5. Results are reported over CUB validation set using the CAM method (Zhou *et al.*, 2016) with IoU as a LOC measure. LOC curves with pseudo-bbox annotations typically have similar behaviour to the Oracle GT bbox annotations, making them suitable for WSOL model selection. They increase and reach their peak for a similar number of epochs, followed by a decline and stagnation in performance. In contrast, the *cls.* curve reaches its peak toward the end of training when LOC performance has already degraded. *cls.* accuracy is therefore inadequate as a WSOL selection criterion to achieve high LOC accuracy performance. Misalignment between LOC and *cls.* behaviour has been studied further in (Choe *et al.*, 2020)

Romero & Granger, 2022). SS (Uijlings *et al.*, 2013) is a commonly used method where no parameters are learned. Carefully engineered features and score functions are used to greedily merge low-level superpixels that produce the final region proposals. The per-proposal score can be used to rank the likelihood that the proposal contains an object.

(b) *Supervised with class-labels*: A middle-ground approach is to use class-labels which are available in the WSOL setting. The CLIP model (Lin *et al.*, 2023) is a model trained on millions of generic text-image pairs from the Internet. Pretrained CLIP is used to pseudo-annotate our held-out $\mathcal{D}_{val}$. Class labels are used as a prompt to generate a CAM for the labeled object in the image. The CAM is automatically thresholded using the Otsu method (Otsu, 1979). Then, a bbox is constructed around each connected object. Since CLIP points to the discriminative region associated with the input class-label, we select the largest bbox as a pseudo-bbox.

(c) *Supervised with class agnostic bboxes*: An alternative approach involves using a region proposal model that is trained using a general dataset like Pascal VOC and MS-COCO, with LOC bboxes. This methodology differs in that it does not employ class-labels, but rather LOC bboxes that are class-agnostic. A typical example is the RPN (Ren *et al.*, 2015), a fully convolutional network that simultaneously predicts object bounds at each position, and provides objectness scores that allow to rank proposals.

For SS and RPN several bboxes are generated per image. To discard irrelevant bboxes and select the most discriminative one, we leverage the pointing game analysis (Zhang *et al.*, 2018a) (see Fig.4.3). To this end, a classifier pretrained over $\mathcal{D}_{train}$ is considered. It is trained until convergence using only class-labels. The pointing game uses the maximum CAM response to select the most discriminative bboxes. In the case where multiple bboxes are ultimately selected, they are scored by the classifier response, and the bbox with the highest score is ultimately selected. Pseudo-bbox annotations are constructed for $\mathcal{D}_{val}$, where each box contains the most discriminative object labeled in the image. This process is only executed once before performing any WSOL training. The generated bboxes are stored on disk, and used for future WSOL training. Therefore, our approach does not add any computation time to the training itself.

Fig.4.2 shows the behaviour of LOC performance when using pseudo-bboxes vs oracle bboxes over $\mathcal{D}_{val}$. The overall trend is similar as they both increase during the first epochs and reach their peak, decline and stagnate over the following epochs. This is extremely helpful for early stopping and hyper-parameter search. We also observe that LOC using CLIP pseudo-annotation yields higher LOC than the with the oracle. This is plausible since a model prediction can be further aligned with some annotations than others.

To provide a realistic evaluation protocol for WSOL, we proposed a strategy to generate pseudo-bboxes for $\mathcal{D}_{val}$. It bypasses the need for manually annotated (oracle) bboxes by leveraging pretrained region proposal models. Then, the pointing game analysis is used to select the most discriminative proposals. In §4.4, we show that using manually annotated bboxes for $\mathcal{D}_{val}$ set can increase LOC accuracy on $\mathcal{D}_{test}$, leading to performance overestimation. We also show that noisy pseudo-bboxes can effectively be used for model selection in the WSOL. This is initially shown using synthetic boxes, then using our generated pseudo-bboxes. These pseudo-bboxes are made public to the researchers for the design of more realistic WSOL methods.

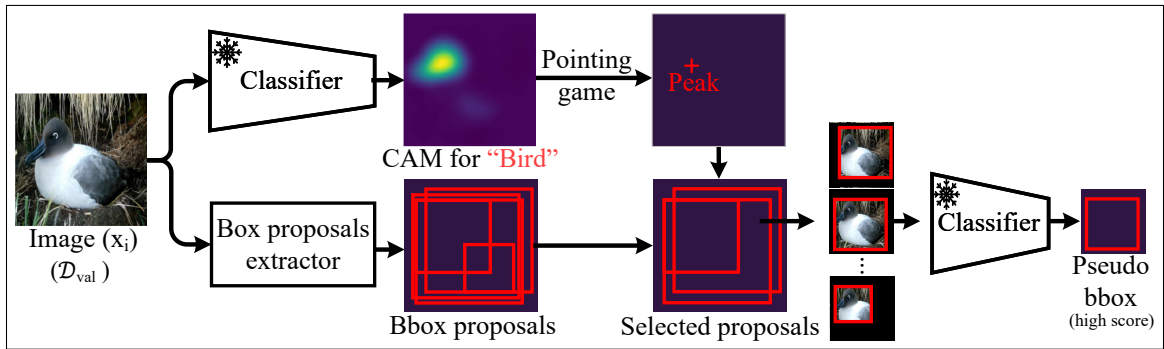


Figure 4.3 Our proposed LOC pseudo-bbox annotator $\hat{o}$. A set of bbox proposals is initially extracted using a region proposal model, and from them, discriminative ones are selected using pointing game analysis (Zhang *et al.*, 2018a). In the case where multiple bboxes are selected, the classifier confidence over the foreground region is used to select the most discriminative bbox

4.3.2 Threshold Estimation

WSOL methods seek to produce bboxes corresponding to particular objects. They generate a LOC map $\mathcal{M} = \mathcal{M}^c \in \mathbb{R}^{H,W}$ highlighting the object of a class c , where $\mathcal{M}_{ij}$ denotes the pixel value at location (i, j) for $i \in \{1, 2, \dots, H\}$ and $j \in \{1, 2, \dots, W\}$. Higher values of $\mathcal{M}_{ij}$ indicate the foreground and lower values indicate the background. The LOC map $\mathcal{M}$ is first normalized due to varying score statistics across images. After normalization, WSOL methods apply a threshold $\tau \in [0, 1]$ to $\mathcal{M}$ to create a binary mask $\{\mathcal{M}_{ij} \geq \tau\}$ for producing bbox. Conventionally, researchers either fix τ or tune it as a hyperparameter. However, the optimal τ varies based on the architecture and dataset. Choe et al. Choe *et al.* (2020) proposed using multiple thresholds, $\tau \in [0, 1]$, to compute bboxes for different masks, and selecting the optimal threshold τ^* based on performance statistics. While effective, this method tends to overestimate performance by using ground-truth bboxes for threshold estimation, which is unrealistic in the WSOL setting. Moreover, for τ estimation, different automatic thresholding methods like OT⁵ lead to an underestimation of performance, as they fail to consider the contextual relationships within the image. To address these issues, we propose using pseudo-bbox labels $\hat{o}(x_i)$ produced by our method for τ estimation. This approach is more realistic and achieves performance that is competitive to the case where τ is estimated with the test set.

4.4 Experimental Validation

4.4.1 Methodology

Datasets. We employed well-established and challenging datasets that are commonly used for evaluating WSOL models. They are, (i) **CUB** Wah *et al.* (2011), encompasses 200 classes with a total of 11,788 images. These are divided into 5,994 images for training and 5,794 for testing. Additionally, for validation, we employed an independent validation set comprising 1,000 images

⁵ OT evaluates all possible threshold values, calculates the variance within each of the two clusters (background and foreground), and selects the threshold that minimizes the weighted sum of these variances.

representing five per class, as provided by Choe *et al.* (2020). (ii) **ILSVRC** (Deng *et al.*, 2009) consists of around 1.2 million images for training and 10,000 images for the validation set, distributed across 1,000 classes. We utilized the validation split included with the original dataset as our test set, given its ample sample size suitable for testing. However, for validation, we utilized **ILSVRC-V2**, collected by Recht *et al.* (2019) and annotated by Choe *et al.* (2020), which minimizes biases toward the test set. Moreover, to ensure a consistent and fair comparison, we closely adhered to the dataset-splitting strategy proposed by Choe *et al.* (2020) for both datasets. (iii) Extended results on two medical datasets (GLaS and CAMELYON) are presented in supplementary material.

WSOL methods. To compare the performance of the proposed protocol, we employed eight state-of-the-art WSOL methods, which include CAM (Zhou *et al.*, 2016), HaS (Singh & Lee, 2017), ACoL (Zhang *et al.*, 2018c), ADL (Choe & Shim, 2019), NL-CCAM (Yang *et al.*, 2020), TS-CAM (Gao *et al.*, 2021) SCM (Bai *et al.*, 2022), and SAT (Wu *et al.*, 2023). Further details regarding these methods can be found in the supplementary material.

Implementation details. For each dataset, we adopted a uniform batch size of 32, and resized each image to a size of 256×256 pixels and then employed random cropping to 224×224 pixels followed by random horizontal flipping. Experiments were performed on 50 training epochs for CUB and 10 for ILSVRC. Furthermore, each method was trained using the stochastic gradient descent optimizer while utilizing four shared hyperparameters for each method: learning rate, step size, weight decay, and gamma. For these shared hyperparameters, a total of 180 configurations were chosen. Additionally, for different methods, specific parameters were selected from their feasible range, and a cartesian product of these hyperparameters with the shared ones was computed to produce the final grid for hyperparameter⁶ optimization. This extensive set of configurations, substantially larger than the scope explored in prior studies, e.g. Choe *et al.* (2020), provides more fair comparisons among the different methods. Moreover, to study the impact of noisy bboxes on the model selection, we train CAM on CUB until 100 epochs, exploring 1,080 configurations using common hyperparameters.

⁶ The comprehensive details on the hyperparameter space are included in the supplementary material.

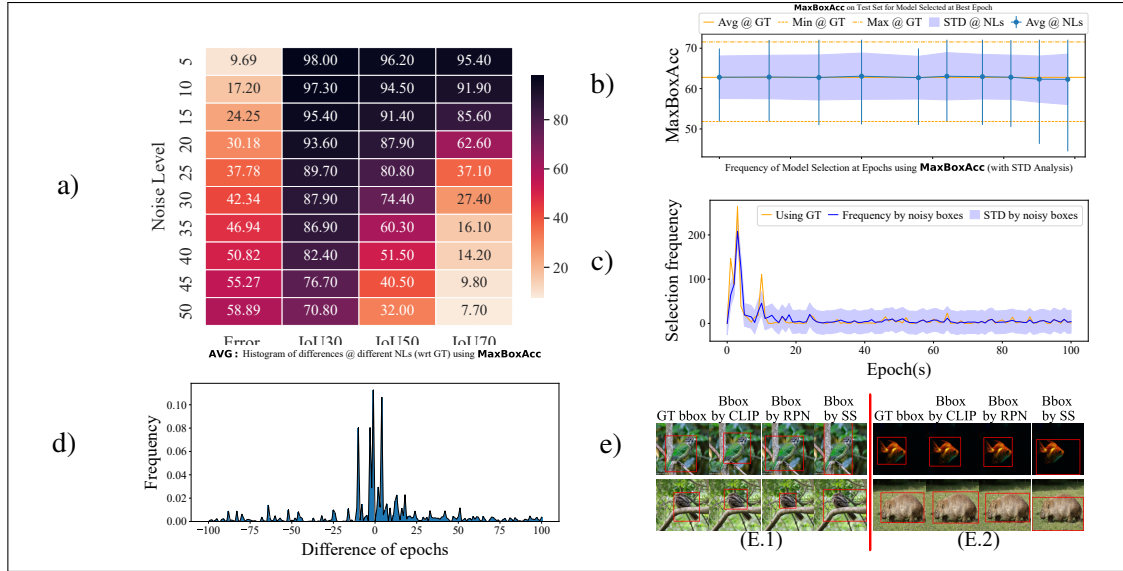


Figure 4.4 (A) The heatmap illustrates the inaccuracies in noisy bboxes at various noise levels generated by augmenting GT bboxes. (B) The blue line represents the average LOC accuracy over various configurations on the test set, incorporating maxima, minima, and standard deviation; the model is selected using a validation set comprising noisy bboxes with varying noise levels. In contrast, orange lines represent the average maximum and minimum performance of a model chosen during hyperparameter optimization over GT for configurations analogous to those employed with noisy labels. (C) Histogram illustrating the variance in model selection epochs when using bboxes at various noise levels w.r.t ground truth boxes for experiments with identical configurations, highlighting tendency around zero. (D) Model selection epoch frequency using GT and noisy boxes across different configurations and noise levels. (E) Illustration of pseudo-bboxes along with the GT over (E.1) CUB and (E.2) ILSVRC validation set

4.4.2 Results and Discussions

(1) Model selection with noisy ground truth bboxes. This section presents an initial results designed to validate the robustness of the proposed model selection techniques. We systematically perturbed the GT bbox, initially employed for model selection, to simulate the scenario for noisy or imprecise GT annotations. The primary aim is to evaluate the sensitivity of model selection to errors in the GT bbox. *The objective is to determine whether growing degrees of GT bboxes noise affects model selection performance?* Positive results would substantiate the reliability of our model selection mechanism under real-world inaccuracies in bbox annotations. Our experiments encompass ten noise levels (see Fig.4.4a), and involve 300 experiments using the

CAM method with perturbed bboxes for validation set model selection. Our findings reveal that the WSOL method maintains its localization performance even with very noisy validation set bboxes (Fig.4.4b). This suggests that model selection can be effectively performed with coarse localization cues. The selection frequency across different epochs using GT and noisy bboxes is remarkably similar (Fig.4.4c). In Fig.4.4d histogram comparing model selection epochs using GT and noisy bboxes indicates a central tendency around zero, showing that coarse GT bboxes from various sources enable reliable model selection while sustaining a high level of accuracy. *More information regarding the generation of bboxes can be found in supplement materials.*

(2) Generation of pseudo-GT bboxes. Three different off-the-shelf models, SS, RPN, and CLIP, are employed for generating pseudo-bboxes. Since, these methods generate a set class-agnostic pseudo-bboxes and *pointing game* is employed for selecting discriminative boxes. This involves harvesting CAM from a pretrained classifier, pinpointing the peak activation, and evaluating its spatial overlap with annotated GT-bboxes. A successful LOC of pointing game is counted if the peak activation in CAMs overlaps with annotated bboxes, using $\frac{Hits}{Hits+Misses}$. The pointing game accuracy of pretrained classifier’s CAMs used to filter discriminative bboxes, achieving 98.70% accuracy on the CUB validation split, 98.00% on the test split, 88.05% on the ILSVRC validation split, and 89.03% on the test split.

Details about generating pseudo-bboxes $\hat{o}(x_i)$ is presented in §4.3, and the IoU accuracy of the pseudo-bboxes w.r.t. GT bboxes on the validation set is reported in Tab.4.1. Despite the noise, these pseudo-boxes are expected to be useful for model selection and threshold estimation, as demonstrated in the previous section. Examples of pseudo-bboxes generated by RPN and SS are shown in Fig.4.4e

(3) Pseudo-bboxes and τ estimation. We employed our proposed model selection protocol across different region proposal methods as outlined in §4.4.1. During the training of each method, we employed a validation set for model selection with which encompassed image, class labels, pseudo-bboxes generated by different off-the-shelf methods. We compared the performance of the models trained with these pseudo-bboxes against those trained with ground

Table 4.1 IoU performance of generated pseudo-bboxes w.r.t GT bboxes on the validation set

Annotator	CUB	ILSVRC
SS + Proposed bbox selection	39.98	45.07
RPN + Proposed bbox selection	71.23	61.08
CLIP + Proposed bbox selection	68.80	64.41

truth bboxes. For every region proposal method, we selected a model using validation accuracy. Its performance was tested using a test set, and then the best-performing model was selected using the same type of supervision used to select the best on the validation set. The results from these experiments are reported in Tab.4.2, where WSOL models employing pseudo label bboxes for model selection achieved results comparable to those using GT bboxes. This suggests that reliable model selection is possible with coarse bboxes.

Results also show that selecting models based on best validation (BV) across different hyperparameter configurations yields competitive results and is a more realistic approach when compared to the model selected using best test (BT) accuracy. Furthermore, when the best model is selected using pseudo-bboxes $\hat{o}(x_i)$ and the threshold τ is estimated using $\hat{o}(x_i)$ (BV-VT), we achieve accuracy comparable to methods that utilize manual bbox annotations for validation and use the BT for τ estimation (BT-TT). *This indicates that model selection should be based on validation set performance across different experiments, avoiding overestimating model performance.* Additionally, if the foreground saliency mask predictor Liu, Yuan, Sun & et. al (2010) is employed to fit oracle bboxes (converted into a binary map) over the $\mathcal{D}_{val}$, it achieves a MaxBoxAcc of 68.7% and 94.0% on the ILSVRC and CUB, respectively Choe *et al.* (2020). Performance is comparable to models trained in the WSOL setting, which utilize a much larger $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ for model selection. Our results also suggest that the pseudo-box annotation $\mathcal{D}_{val}$ should be used for model evaluation and τ estimation.

(4) Recommendation for realistic evaluation. In line with our proposed evaluation protocol and empirical results, we recommend the following practices. Training and evaluation should be constrained to only utilizing class-level labels without resorting to manual bbox annotations for

Table 4.2 Test-set MaxBoxAcc of WSOL models with different selection criteria on CUB and ILSVRC. The *select* column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the *cls.* performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT

Method	Select	CUB (MaxBoxAcc)					ILSVRC (MaxBoxAcc)				
		cls.	GT	RPN	CLIP	SS	cls.	GT	RPN	CLIP	SS
CAM (<i>cvpr'16</i>) ResNet50	BT-TT	32.99	70.40	71.10	70.62	69.89	46.29	64.06	63.60	63.90	63.60
	BT-VT	–	69.50	70.43	69.88	27.97	–	63.90	62.89	63.88	46.97
	BV-TT	29.37	70.40	70.40	69.76	61.20	46.83	64.06	63.60	64.01	63.60
	BV-VT	–	69.20	68.93	67.98	22.79	–	63.88	62.89	63.88	45.91
HaS (<i>iccv'17</i>) ResNet50	BT-TT	37.00	75.85	75.85	75.85	74.73	37.82	63.77	63.94	63.77	63.30
	BT-VT	–	74.68	75.69	75.69	25.42	–	63.86	62.08	63.28	49.02
	BV-TT	30.11	75.49	74.92	73.38	72.69	45.80	63.94	63.34	63.54	63.30
	BV-VT	–	74.68	74.02	73.24	22.43	–	63.86	62.08	63.06	48.21
ACoL (<i>cvpr'18</i>) ResNet50	BT-TT	46.59	74.64	74.64	75.37	74.14	50.46	62.93	62.75	63.45	62.92
	BT-VT	–	73.50	73.83	74.83	33.43	–	63.48	61.61	63.47	52.38
	BV-TT	34.89	74.64	74.64	72.29	56.04	50.46	63.70	62.75	62.93	62.92
	BV-VT	–	73.50	73.83	72.21	27.85	–	63.48	61.31	62.80	52.38
ADL (<i>cvpr'19</i>) ResNet50	BT-TT	40.97	76.63	76.63	76.06	74.99	39.80	65.11	65.97	65.11	65.19
	BT-VT	–	76.21	76.52	75.92	27.39	–	65.2	64.29	65.04	49.61
	BV-TT	37.17	75.83	75.97	75.80	32.08	44.23	65.28	65.18	65.32	65.06
	BV-VT	–	72.73	75.85	74.73	22.21	–	65.2	62.30	65.04	47.09
CCAM (<i>wacv'20</i>) VGG	BT-TT	52.67	65.58	65.58	65.22	45.97	49.80	60.63	60.63	60.63	52.72
	BT-VT	–	64.44	65.27	62.54	31.89	–	60.52	60.52	60.56	46.63
	BV-TT	52.67	65.58	65.58	65.22	45.97	49.80	60.63	60.63	60.63	52.72
	BV-VT	–	62.44	64.84	62.54	31.89	–	60.52	60.52	60.56	46.63
TS-CAM (<i>iccv'21</i>) DeiT-S	BT-TT	51.77	90.19	90.35	89.52	88.71	6.476	66.75	66.75	66.75	66.17
	BT-VT	–	89.33	90.16	89.52	29.49	–	66.65	65.12	65.32	47.63
	BV-TT	39.17	89.52	89.52	89.16	77.39	6.476	66.75	66.14	66.75	55.7
	BV-VT	–	88.85	88.90	88.95	24.19	–	66.65	64.97	65.32	46.80
SCM (<i>eccv'22</i>) DeiT-S	BT-TT	30.82	91.56	92.25	92.26	91.76	47.55	61.76	61.75	61.75	59.76
	BT-VT	–	90.26	92.00	92.16	26.44	–	61.76	61.15	61.38	49.27
	BV-TT	30.82	91.56	92.25	92.26	80.16	47.55	61.76	61.75	61.75	59.76
	BV-VT	–	90.26	92.00	92.16	22.02	–	61.76	60.21	61.38	49.27
DiPS (<i>IVC'23</i>) DeiT-S	BT-TT	76.2	97.0	96.2	96.9	96.2	63.2	69.7	67.6	67.6	67.3
	BT-VT	–	97.0	96.3	96.9	96.2	–	69.3	68.2	68.2	60.8
	BV-TT	75.3	96.8	96.7	96.8	96.3	62.9	69.3	68.2	68.2	67.3
	BV-VT	–	96.4	96.2	96.3	96.5	96.1s	69.3	68.2	68.2	60.4
SAT (<i>iccv'23</i>) DeiT-S	BT-TT	69.60	92.14	92.45	91.45	92.23	64.59	70.12	67.08	70.13	70.13
	BT-VT	–	91.66	92.26	91.00	41.62	–	69.46	64.78	69.05	57.06
	BV-TT	70.24	91.33	91.75	89.97	91.33	66.17	70.12	67.08	70.13	70.13
	BV-VT	–	90.86	91.71	89.93	28.04	–	69.46	64.78	69.05	57.06

model selection and threshold estimation: (i) We recommend utilizing pseudo-bboxes generated by off-the-shelf methods for validation set. Our experiments show that these pseudo-bboxes can effectively provide the LOC annotation for model selection, achieving performance comparable

Table 4.3 Analysis of MaxBoxAcc (IoU-50) versus IoU using oracle bboxes for model selection. Analysis using pseudo-bboxes is presented in *supplementary material* exhibiting same behaviour

Method	CUB-200-2011		ILSVRC	
	MaxBoxAcc (IoU-50)	IoU	MaxBoxAcc (IoU-50)	IoU
CAM (Zhou <i>et al.</i> , 2016)	70.40	56.71	64.06	57.89
HaS (Singh & Lee, 2017)	75.85	59.81	63.77	58.49
ACoL (Zhang <i>et al.</i> , 2018c)	74.64	58.29	62.93	56.39
ADL (Choe & Shim, 2019)	76.63	60.12	65.11	58.46
NL-CCAM (Yang <i>et al.</i> , 2020)	65.58	54.76	60.63	54.98
TS-CAM (Gao <i>et al.</i> , 2021)	90.19	69.78	66.75	59.54
SCM (Bai <i>et al.</i> , 2022)	91.56	70.27	61.76	54.55
SAT (Wu <i>et al.</i> , 2023)	92.14	73.67	70.12	62.80

to using GT annotations. **(ii)** Estimating thresholds using pseudo-bboxes from the validation set instead of the test set to produce bboxes from LOC maps. This approach addresses the unrealistic practice of thresholding. **(iii)** Our experiments suggest that model selection across various experiments, each employing different hyperparameters, should be based on the performance of pseudo-bboxes on the validation set. This approach mitigates bias in model performance evaluation. **(iv)** Be cautious when using thresholded-IoU metrics (IoU-30, IoU-50, IoU-70) and their variants, as these can mislead by considering uniform performance across instances above the set threshold. Our analysis reveals a non-linear relationship between thresholded-IoU and IoU (Tab.4.3). We recommend the use of non-thresholded-IoU for realistic evaluation.

4.5 Conclusion

The currently-used protocol (Choe *et al.*, 2020) for evaluation of WSOL methods has driven significant progress in the field. However, its reliance on manually annotated bboxes during validation for model selection, and annotated test set for threshold estimation leads to an overestimation of localization performance on the test set. In this work, we propose generating pseudo-bboxes for the validation set using off-the-shelf pretrained region proposal models for both model selection and threshold estimation. This threshold is then employed to produce bboxes from LOC maps on the test set. Our approach was evaluated with different WSOL

methods and shows that employing pseudo-bboxes for model evaluation achieves localization performance comparable to models selected using GT bboxes where the threshold is estimated on the test set. Our approach is therefore a promising and more realistic alternative than using a GT annotated held-out dataset. Despite the promising results with our protocol, performing model selection in WSOL without LOC annotation remains an open issue.

Supplementary materials include a description of evaluated methods, steps to generate noisy GT bboxes, the performance of pseudo-bboxes at different selection steps, the impact of thresholding on results, and results on medical datasets. Results obtained using addition methods, specifically F-CAM (Belharbi *et al.*, 2022c) and DiPS (Murtaza *et al.*, 2023b), are available: github.com/shakeebmurtaza/wsol_model_selection.

Acknowledgements. This research was supported by the Natural Sciences and Engineering Research Council of Canada. We also thank the Digital Research Alliance of Canada for the use of their computing resources.

4.6 Supplementary Material

This supplementary material contains the following content:

- A. Review of evaluated methods and their search spaces.** In this section, we provided a detailed description of methods employed to evaluate the efficacy of our proposed protocol for model selection. Additionally, it outlines various hyperparameters for each method and their respective feasible range.
- B. Generation of noisy ground truth bboxes.** This section delineates the methodology for generating noisy ground truth (GT) bounding boxes (bboxes) to analyze their impact on model selection.
- C. Pseudo-bboxes performance across different selection steps.** This section presents the performance of pseudo-bboxes at different selection steps of our proposed method.

- D. Localization evaluation in WSOL (thresholded-IoU vs IoU).** This section describes an ongoing challenge in WSOL where discrepancies in localization performance exist between the commonly used MaxBoxAcc and the IoU metric.
- E. Localization Evaluation with different thresholds.** This section presents the results at different thresholds along with their average.
- F. Experiments with medical datasets.** To show the generalizability of our proposed evaluation protocol across various domains, we employ medical datasets.

4.6.1 Review of Evaluated Methods and their Search Spaces

4.6.1.1 Evaluated Methods

To assess the efficacy of the proposed protocol, we employ eight methods published in top-tier venues from 2016 to 2023. These methods are presented chronologically within this section, providing a comprehensive overview.

Class activation mapping (CAM) (Zhou *et al.*, 2016) is able to extract an activation map for a particular class using a pre-trained CNN-classifier with global average pooling. It generates the final map by aggregating different activation maps from the penultimate convolution layer based on the contribution towards each class by using weights from the last fully connected layer.

Hide-and-seek (HaS) (Singh & Lee, 2017) force the network to look beyond discriminative regions of a particular object by augmenting the input image. It hides patches of input image during training by employing two hyper-parameters; drop rate and grid size.

Adversarial complementary learning (ACoL) (Zhang *et al.*, 2018c) employees an architecture with two parallel classifier heads that tires to find complementary regions by adversarially erasing high-scoring activations.

Attention-based dropout layer (ADL) (Choe & Shim, 2019) works similarly as ACoL by erasing high-score activation to force by employing drop masks generated without second classifiers head.

Non-local combinational class activation maps (NL-CCAM) (Yang *et al.*, 2020). In this paper, the author argues that employing the activation map of the class with the highest classifier's score may only highlight discriminative regions and for certain images it tends to focus on background regions. To address this limitation, the author proposes to combine activation maps of different classes. This combination is based on the respective probability score of each, encompassing a spectrum from the highest to the lowest.

Token semantic coupled attention map (TS-CAM) (Gao *et al.*, 2021). TS-CAM is a cascaded ViT-CNN architecture that proposes to redistribute class information to patch tokens. This is achieved by implementing a CNN-based classification (cls.) head atop the patch tokens, thereby rendering the class token class-agnostic. Therefore, these class tokens are combined with the activation map extracted from the last convolutional layer to produce an activation map highlighting different object parts of a particular class.

Spatial calibration module (SCM) (Bai *et al.*, 2022). This paper introduced an SCM module atop the transformer features to align the boundaries of the generated map with the object boundaries by avoiding partial activation in different areas of the activation map. This module integrates semantic similarities presented in patch tokens and their spatial relationships into a unified model. SCM effectively recalibrates the transformer's attention and semantic representations to mitigate the background noise and sharpen object boundaries.

Task-specific spatial-aware token (SAT). (Wu *et al.*, 2023) This paper introduces a spatial-aware token (SAT) into the transformer's input space. Like class token that is able to accumulate information for cls. tasks, it is incorporated to aggregate the global representation of the object of interest. Furthermore, the SAT is a passed-to spatial-query attention module that treats the SAT as a query to calculate similarity with different patches and produces probabilities for foreground object for producing accurate localization maps.

4.6.1.2 Hyperparameter Search Space

To fairly compare different WSOL methods, we took steps to minimize human biases during training. This includes employing pseudo-bboxes for the evaluation and sampling hyperparameter values from the feasible range, except for annotations on the test set, which were used only to assess the trained model’s performance.

Each method was trained using four shared hyperparameters, while the additional hyperparameters were specific to each model. We sampled the values for these hyperparameters from their feasible range. A cartesian product of these and shared hyperparameters was computed to create the final grid of hyperparameters for training each model. A detailed summary of the hyperparameters employed to train different WSOL models is presented in Tab.4.4.

Table 4.4 Hyperparameter search space for different methods

Method	Hyperparameter	Sampling Distribution	Range
Common HPs	LR, WD, Gamma	LogUniform	$[10^{-5}, 10^0]$
	Step Size	Uniform	CUB: $[5 - 45]$ ILSVRC: $[2 - 9]$
CAM (Zhou <i>et al.</i> , 2016), TS-CAM (Gao <i>et al.</i> , 2021) SCM (Bai <i>et al.</i> , 2022), NL-CCAM (Yang <i>et al.</i> , 2020)	Common HPs	-	-
HaS (Singh & Lee, 2017)	Drop Rate, Drop Area	Uniform	$[0, 1]$
ACoL (Zhang <i>et al.</i> , 2018c)	Erasing Threshold	Uniform	$[0, 1]$
ADL (Choe & Shim, 2019)	Drop Rate, Erasing Threshold	Uniform	$[0, 1]$
SAT (Wu <i>et al.</i> , 2023)	Area Threshold	Uniform	$[0, 1]$

4.6.2 Generation of Noisy Ground Truth Bboxes

In the main paper, we introduce a validation protocol designed to evaluate the robustness of the proposed model selection techniques in the presence of noisy GT bboxes. This protocol systematically perturbs the GT bboxes, initially used for model selection, to emulate conditions of noisy or imprecise GT annotations. To produce noisy GT bboxes, a sequence of random transformations is applied to the GT bboxes, creating varying noise levels. For each transformation, a total of ten unique noise levels are defined. These levels signify the maximum likelihood of deformation at each noise level. This likelihood is derived by sampling the deformation value using a uniform distribution that varies from 5 to 50, with intervals set at 5. To generate

nosy bboxes we, first apply, scaling transformation to the GT bboxes between -50% and +50% with a maximum likelihood of a particular noise level. Following the scaling, we apply shift transformation to the scaled bbox by choosing a random shift length, where the shift length is set between 0% and the maximum size percentage corresponding to a particular noise level. Finally, we modify the aspect ratio of bbox based on a probability factor ‘p’ which indicates the likelihood, representing a specific noise level.

4.6.3 Pseudo-bboxes Performance Across Different Selection Steps

Different off-the-shelf models, SS, RPN, and CLIP, are employed to produce pseudo-bboxes. These methods generate a set of class-agnostic pseudo-bboxes. To select discriminative boxes from a pool of object proposals, the pointing game analysis was employed (Zhang *et al.*, 2018a). This involves harvesting CAM from a pre-trained classifier and pinpointing the peak activation that is used to select discriminative boxes. Despite the initial filtering of bboxes via the pointing game, a substantial number of bboxes remained. To address this, we employ a sequential refinement process, in which we initially filter the top 20% based on objectness or classifier score for boxes obtained from RPN and SS, respectively. Subsequently, the pointing game was employed to refine this selection, followed by selection of top-performing boxes based on score.

Table 4.5 Performance of pseudo-bboxes obtained using different off-the-shelf region proposal methods across multiple refinement stages over the validation set. The table highlights the incremental improvement in IoU through various selection steps

Method	CUB (IoU)			ILSVRC (IoU)		
	SS	RPN	CLIP	SS	RPN	CLIP
Mean IoU (PG*)	32.21	28.71	–	23.38	27.57	–
Mean IoU (PG*, 20% Filtered)	33.69	37.89	–	34.99	37.17	–
Mean IoU (Top Box, 20% by Objectness Score)	39.90	69.80	–	44.90	54.02	–
Mean IoU (Top Box, 20%, PG*+Scoring)	39.98	71.23	–	45.07	61.08	–
Upper Bound (Select by IoU with GT)	64.07	83.66	–	65.46	84.42	–
IoU from Otsu	–	–	68.80	–	–	64.41
IoU using 1K Threshold	–	–	69.56	–	–	65.78

*PG: Pointing Game

Table 4.6 Comparative Analysis of MaxBoxAcc (IoU-50) versus IoU on CUB and ILSVRC with different model selection methods

Method	Backbone	CUB-200-2011 (MaxBoxAcc)					CUB-200-2011 (IoU)					ILSVRC (MaxBoxAcc)					ILSVRC (IoU)				
		c.l.s.	GT	RPN	CLIP	SS	c.l.s.	GT	RPN	CLIP	SS	c.l.s.	GT	RPN	CLIP	SS	c.l.s.	GT	RPN	CLIP	SS
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr</i> ,2016)	ResNet50	66.98	70.40	71.10	70.62	69.89	55.53	56.71	56.88	56.65	56.76	61.48	64.06	63.60	63.90	63.60	56.17	57.89	57.42	57.68	57.42
HaS (Singh & Lee, 2017) (<i>iccv</i> ,2017)	ResNet50	67.62	75.85	75.85	75.85	74.73	57.01	59.81	59.81	59.81	59.39	61.69	63.77	63.94	63.77	63.30	56.81	58.49	58.39	58.49	57.32
ACoL (Zhang <i>et al.</i> , 2018c) (<i>cvpr</i> ,2018)	ResNet50	66.62	74.64	74.64	75.37	74.14	55.32	58.29	58.29	58.55	58.13	61.98	62.93	62.75	63.45	62.92	55.62	56.39	56.32	56.94	56.39
ADL (Choe & Shim, 2019) (<i>cvpr</i> ,2019)	ResNet50	67.82	76.63	76.63	76.06	74.99	55.64	59.12	59.12	58.93	58.32	62.81	65.11	65.97	65.11	65.19	56.39	58.46	58.37	58.55	58.54
NL-CCAM (Yang <i>et al.</i> , 2020) (<i>wacv</i> ,2020)	VGG-GAP	64.15	65.58	65.58	65.22	45.97	54.11	54.76	54.76	54.59	47.88	58.42	60.63	60.63	60.63	52.72	51.59	54.98	54.98	54.98	49.44
TS-CAM (Gao <i>et al.</i> , 2021) (<i>iccv</i> ,2021)	DeiT-S	88.36	90.19	90.35	89.52	88.71	69.14	69.78	69.83	69.95	68.36	56.40	66.75	66.75	66.75	66.17	53.67	59.54	59.54	59.54	59.00
SCM (Bai <i>et al.</i> , 2022) (<i>eccv</i> ,2022)	DeiT-S	88.47	91.56	92.25	92.26	91.76	68.64	70.27	70.89	70.93	70.34	57.92	61.76	61.75	61.75	59.76	52.13	54.55	54.56	54.56	53.47
F-CAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv</i> ,2022)	ResNet50	24.95	89.83	89.23	89.81	88.26	37.72	68.79	68.12	68.62	68.30	—	—	—	—	—	—	—	—	—	—
SAT (Wu <i>et al.</i> , 2023) (<i>iccv</i> ,2023)	DeiT-S	79.70	92.14	92.45	91.45	92.23	63.13	73.67	73.59	72.92	73.61	64.94	70.12	67.08	70.13	70.13	56.09	62.80	58.55	62.80	62.80

In the case of CLIP, we utilized Otsu’s thresholding method to identify binary maps, upon which bboxes were delineated around the largest connected areas. A comprehensive description of the proposed method for generating pseudo-bboxes is provided in the main paper.

The performance of pseudo-bboxes at different selection steps is presented in Tab.4.5. This table shows that as we select relevant bboxes generated by SS or RPN at each stage, we progressively choose better-performing bboxes, resulting in reliable performance relative to the upper bound performance when using GT bboxes to select top-performing bbox. Initially, pseudo-bboxes generated by SS and RPN are filtered based on objectness or classifier scores, followed by a pointing game analysis for further refinement. The results indicate a significant improvement in the mean intersection over union (IoU) across these selection stages. For example, after the initial selection and filtering of the top 20% based on objectness scores, the IoU increases substantially, demonstrating the efficacy of our proposed method. In contrast, CLIP generates activation maps that highlight particular objects. Otsu’s thresholding method is employed to convert these maps into binary images, enabling the delineation of bboxes around the largest connected areas. Despite the single-stage selection process for CLIP-generated maps, the resulting bounding boxes achieve competitive performance.

4.6.4 Localization Evaluation in WSOL (Thresholded-IoU vs IoU)

So far, we have reported the localization performance using MaxBoxAcc metric (Choe *et al.*, 2020). It is also known as as *GT-known localization* metric. It scores one point when the IoU between the GT bbox and the predicted box is above 50%, otherwise, it scores 0. It is referred

to IoU 5-0 as well. It is a well established and commonly used metric in WSOL. In addition to IoU-50, we report the IoU in Tab.4.3 of the main paper. These results show that model selection using cls. accuracy still lead to poor IoU. However, selection using our proposed pseudo-bboxes yields competitive IoU compared to when using oracle bboxes.

The comparison based on IoU (Tab.4.3 of the main paper) lead us to an interesting result presented in Tab.4.6. This table shows that **MaxBoxAcc**, using the commonly oracle bboxes, gives largely higher localization scores compared to the exact localization accuracy reported by IoU. For instance, SAT method (Wu *et al.*, 2023) scores 92.14% in **MaxBoxAcc**, while it only scores 73.12% over CUB dataset. When considering only **MaxBoxAcc**, the results give the impression that CUB dataset is saturated, especially when the same authors (Wu *et al.*, 2023) have reported a **MaxBoxAcc** of 98.45%. However, when inspecting IoU metric, localization is still low at 73.12%.

In addition, since **MaxBoxAcc** is based on thresholding, extreme localization scores can hit the same scoring point. For instance, a prediction with $\text{IoU} = 50.1\%$ scores the same point as when the prediction is $\text{IoU} = 99.99\%$. However, both $\text{IoU} = 49.9\%$, $\text{IoU} = 1\%$ scores 0 point in **MaxBoxAcc**. This makes localization evaluation less efficient.

Despite its common usage in the literature, the aforementioned limitations of **MaxBoxAcc** suggest that reporting IoU along with **MaxBoxAcc** could be beneficial in better assessing localization performance of different methods in WSOL.

Since CUB dataset is relatively easier than ILSVRC dataset, the latter paints a realistic evaluation of the progress that has been done in WSOL. Since the work of Zhou et al. (Zhou *et al.*, 2016) in 2016 up to now, only $\approx 6\%$, and $\approx 4.9\%$ of improvement has been done in term of **MaxBoxAcc** and IoU, respectively. This suggests that a lot of work is still needs to be done to furthermore improve WSOL methods.

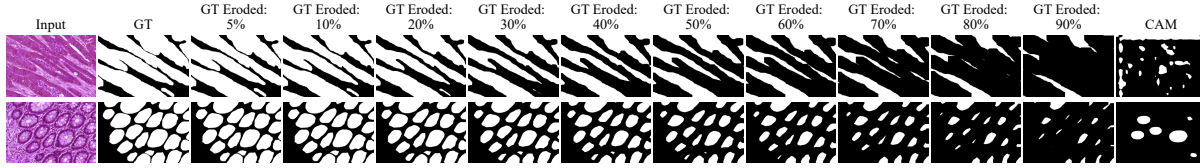


Figure 4.5 Illustration GT masks, noisy masks at various erosion levels

4.6.5 Localization Evaluation with different thresholds

We have previously evaluated localization performance using the **MaxBoxAcc** metric, which assigns a score of one when the IoU between the GT bbox and the predicted bounding box exceeds 50%, and a score of zero otherwise, as well as using raw IoU without any threshold. In addition, we extend the evaluation of **MaxBoxAcc** by examining performance trends at multiple IoU thresholds, namely **IoU-30**, **IoU-50**, and **IoU-70**, along with their average, termed **MaxBoxAccV2**, a metric commonly used in the weakly supervised object localization (WSOL) literature Choe *et al.* (2020). The performance results for these thresholded IoU metrics and their average are detailed in Tab.4.7&4.8. In contrast to raw IoU, the thresholded IoU metrics demonstrate consistent trends when compared to **MaxBoxAcc**, which are sufficient to offer meaningful insights into the results.

4.6.6 Experiments with medical datasets

To show the robustness and generalizability of our evaluation protocol across datasets with varying characteristics, we extended our protocol to the task of localization in histology images—a particularly challenging problem due to its complexity and sensitivity, especially for non-experts attempting to identify regions of interest. For this, we employed two histology image datasets and first simulated noisy masks by applying erosion to the GT masks. This approach allowed us to assess the protocol’s robustness against different levels of noise. Additionally for realistic setup, we extracted pseudo-masks from the activation maps of a pre-trained classifier, thereby evaluating the protocol’s performance in a realistic setting. Our results indicate that the

Table 4.7 Test-set MaxBoxAccV2 that is average of three threshold IoU-30, IoU-50, IoU-70 (here MaxBoxAccV2 is aerage at three threshold) along with the it of WSOL models with different selection criteria on CUB. The *select* column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the cls. performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT

Method	Select	CL				GT				RPN				CLIP				SS			
		IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2
CAM (cvpr'16) ResNet50	BT-TT	73.76	32.99	6.161	37.63	93.95	70.40	20.05	61.46	94.30	71.10	19.74	61.71	94.33	70.62	19.46	61.47	94.06	69.89	20.38	61.44
	BT-VT	—	—	—	—	93.13	69.50	19.07	60.56	94.25	70.43	19.22	61.30	93.73	69.88	18.46	60.69	70.05	27.97	12.80	36.94
	BV-TT	69.72	29.37	5.143	34.74	93.95	70.40	20.05	61.46	93.95	70.40	20.05	61.46	94.70	69.76	18.46	60.97	90.05	61.20	16.44	55.89
	BV-VT	—	—	—	—	93.61	69.20	19.33	60.71	93.61	68.93	19.03	60.52	94.70	67.98	16.58	59.75	58.66	22.79	15.84	32.43
HaS (iccv'17) ResNet50	BT-TT	71.48	37.00	11.25	39.91	94.59	75.85	28.49	66.31	94.59	75.85	28.49	66.31	94.59	75.85	28.49	66.31	94.64	74.73	27.42	65.59
	BT-VT	—	—	—	—	92.33	74.68	29.15	65.38	94.44	75.69	28.27	66.13	94.14	75.69	27.71	65.84	67.22	25.42	12.08	34.90
	BV-TT	66.15	30.11	8.715	34.99	93.16	75.49	30.18	66.27	93.57	74.92	29.20	65.89	94.87	73.38	23.76	64.00	92.85	72.69	30.53	65.35
	BV-VT	—	—	—	—	92.33	74.68	29.15	65.38	93.49	74.02	28.99	65.5	94.51	73.24	23.45	63.73	61.28	22.43	27.09	36.93
ACoL (cvpr'18) ResNet50	BT-TT	89.48	46.59	7.490	47.85	96.35	74.64	20.81	63.93	96.35	74.64	20.81	63.93	97.20	75.37	20.29	64.28	96.85	74.14	19.41	63.46
	BT-VT	—	—	—	—	95.73	73.50	20.65	63.29	96.25	73.83	20.59	63.55	96.87	74.83	19.60	63.76	88.29	33.43	9.164	43.62
	BV-TT	80.47	34.89	6.161	40.50	96.35	74.64	20.81	63.93	96.35	74.64	20.81	63.93	97.42	72.29	16.87	62.19	93.71	56.04	8.560	52.77
	BV-VT	—	—	—	—	95.73	73.50	20.65	63.29	96.25	73.83	20.59	63.55	97.25	72.21	16.60	62.01	88.60	27.85	6.817	41.08
ADL (cvpr'19) ResNet50	BT-TT	81.08	40.97	7.973	43.34	96.47	76.63	23.14	65.41	96.47	76.63	23.14	65.41	95.35	76.06	23.16	64.85	95.46	74.99	21.86	64.10
	BT-VT	—	—	—	—	96.27	76.21	22.62	65.03	96.20	76.52	23.00	65.24	95.21	75.92	22.73	64.61	76.85	27.39	10.01	38.08
	BV-TT	77.89	37.17	6.144	40.40	95.40	75.83	23.16	64.79	93.63	75.97	27.25	65.61	96.63	75.80	20.31	64.24	75.45	32.08	6.040	37.85
	BV-VT	—	—	—	—	94.87	72.73	22.69	63.43	93.04	75.85	24.02	64.30	96.08	74.73	19.57	63.46	64.03	22.21	5.505	30.58
CCAM (wacv'20) VGG	BT-TT	90.36	52.67	12.92	51.98	91.49	65.58	19.38	58.81	91.49	65.58	19.38	58.81	91.54	65.22	18.89	58.54	76.68	45.97	17.77	46.80
	BT-VT	—	—	—	—	88.74	64.44	18.55	57.24	90.81	65.27	19.12	58.4	90.42	62.54	18.55	57.17	72.02	31.89	15.15	39.68
	BV-TT	90.36	52.67	12.92	51.98	91.49	65.58	19.38	58.81	91.49	65.58	19.38	58.81	91.54	65.22	18.89	58.54	76.68	45.97	17.77	46.80
	BV-VT	—	—	—	—	90.12	62.44	18.77	57.11	91.37	64.84	19.27	58.49	90.42	62.54	18.55	57.17	72.02	31.89	15.15	39.68
TS-CAM (iccv'21) DeiT-S	BT-TT	85.07	51.77	20.34	52.39	99.22	90.19	55.31	81.57	99.15	90.35	55.16	81.55	98.96	89.52	55.91	81.46	98.92	88.71	51.25	79.62
	BT-VT	—	—	—	—	99.17	89.33	51.89	80.13	99.03	90.16	54.21	81.13	98.60	89.52	55.10	81.07	66.13	29.49	13.87	36.49
	BV-TT	72.41	39.17	13.72	41.76	98.96	89.52	55.91	81.46	98.96	89.52	55.91	81.46	99.11	89.16	53.50	80.58	95.27	77.39	40.90	71.18
	BV-VT	—	—	—	—	98.56	88.85	53.40	80.27	98.87	88.90	51.58	79.78	98.44	88.95	50.98	79.45	59.52	24.19	22.19	35.30
SCM (eccv'22) DeiT-S	BT-TT	62.49	30.82	8.767	34.02	99.25	91.56	56.10	82.30	99.30	92.25	58.40	83.31	99.36	92.26	58.49	83.37	99.24	91.76	56.50	82.5
	BT-VT	—	—	—	—	98.96	90.26	55.35	81.52	99.20	92.00	55.41	82.20	98.94	92.16	58.24	83.11	63.54	26.44	16.86	35.61
	BV-TT	62.49	30.82	8.767	34.02	99.25	91.56	56.10	82.30	99.30	92.25	58.40	83.31	99.36	92.26	58.49	83.37	97.39	80.16	35.86	71.13
	BV-VT	—	—	—	—	98.96	90.26	55.35	81.52	99.20	92.00	55.41	82.20	98.94	92.16	58.24	83.11	63.53	22.02	10.06	31.87
SAT (iccv'23) DeiT-S	BT-TT	91.76	69.60	37.78	66.38	99.37	92.14	66.63	86.04	99.24	92.45	66.79	86.16	99.15	91.45	64.29	84.96	99.30	92.23	66.67	86.06
	BT-VT	—	—	—	—	99.36	91.66	67.34	86.12	99.03	92.26	62.82	84.70	98.49	91.00	64.03	84.50	70.65	41.62	28.04	46.77
	BV-TT	92.33	70.24	39.02	67.19	99.17	91.33	65.06	85.18	99.17	91.75	68.31	86.41	99.13	89.97	60.57	83.22	99.17	91.33	65.06	85.18
	BV-VT	—	—	—	—	98.87	90.86	65.06	84.93	98.79	91.71	66.44	85.64	97.82	89.93	60.40	82.71	77.59	28.04	14.68	40.10

evaluation protocol consistently maintains performance across varying noise levels, confirming its robustness in both simulated and real-world scenarios.

Datasets. We employed two additional datasets to show the robustness of our proposed protocol; (i) *GLaS dataset* is collected for the diagnosis of colon cancer and comprises 165 images derived from 16 Hematoxylin and Eosin (H&E) stained slides. It includes pixel-level and image-level annotations (benign or malignant). The dataset is divided into 67 images for training, 18 for validation, and 80 for testing. Our evaluation follows the protocol established by Rony *et al.* (2023), where three fully supervised examples per class are used for best model over LOC. (ii) *CAMELYON dataset* is a patch-based benchmark extracted using Camelyon16 dataset, which

Table 4.8 Test-set MaxBoxAccV2 that is average of three threshold IoU-30, IoU-50, IoU-70 (here MaxBoxAccV2 is aerage at three threshold) along with the it of WSOL models with different selection criteria on ILSVRC. The *select* column presents (i) BT and BV indicate model selection based on hyperparameter configurations using the test set and validation set, respectively; (ii) TT and VT indicate that the threshold τ is selected using either the test set or validation set. For model selection on the validation set, we consider the GT as a reference, a selection based on the c1s. performance and the three different pseudo-bboxes generation proposed in this work: RPN, CLIP and SS. Our results for models selected with pseudo-bboxes are comparable to those of GT

Method	Select	CL				GT				RPN				CLIP				SS			
		IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2	IoU-30	IoU-50	IoU-70	MaxBoxV2
CAM (cvpr'16) ResNet50	BT-TT	73.16	46.29	20.86	46.77	81.99	64.06	40.22	62.09	81.59	63.60	39.55	61.58	82.02	63.90	39.98	61.96	81.59	63.60	39.55	61.58
	BT-VT	—	—	—	—	81.98	63.90	39.59	61.82	79.49	62.89	37.64	60.00	81.09	63.88	39.85	61.60	68.59	46.97	39.84	51.80
	BV-TT	73.81	46.83	21.26	47.30	81.99	64.06	40.22	62.09	81.59	63.60	39.55	61.58	82.06	64.01	40.02	62.03	81.59	63.60	39.55	61.58
	BV-VT	—	—	—	—	81.88	63.88	39.94	61.9	79.49	62.89	37.64	60.00	81.09	63.88	39.85	61.60	70.13	45.91	39.51	51.84
HaS (iccv'17) ResNet50	BT-TT	65.09	37.82	16.15	39.68	81.3	63.77	42.08	62.38	81.54	63.94	41.7	62.39	81.3	63.77	42.08	62.38	81.43	63.30	39.26	61.33
	BT-VT	—	—	—	—	81.16	63.86	41.67	62.23	78.59	62.08	37.35	59.34	80.35	63.28	40.95	61.52	68.92	49.02	37.13	51.69
	BV-TT	72.59	45.80	21.01	46.46	81.54	63.94	41.7	62.39	81.55	63.34	39.77	61.55	81.82	63.54	39.56	61.63	81.43	63.30	39.26	61.33
	BV-VT	—	—	—	—	81.16	63.86	41.67	62.23	78.59	62.08	37.35	59.34	80.36	63.06	39.21	60.87	68.56	48.21	39.23	52.0
ACoL (cvpr'18) ResNet50	BT-TT	77.99	50.46	22.76	50.40	81.34	62.93	38.23	60.83	81.89	62.75	37.56	60.73	82.21	63.45	39.13	61.59	82.14	62.92	37.50	60.85
	BT-VT	—	—	—	—	81.83	63.48	39.11	61.47	79.94	61.61	36.75	59.43	80.38	63.47	39.11	60.98	76.95	52.38	33.11	54.14
	BV-TT	77.99	50.46	22.76	50.40	82.23	63.70	39.03	61.65	81.89	62.75	37.56	60.73	82.14	62.93	37.5	60.85	82.14	62.92	37.50	60.85
	BV-VT	—	—	—	—	81.83	63.48	39.11	61.47	80.45	61.31	35.89	59.21	81.37	62.80	37.43	60.53	76.95	52.38	33.11	54.14
ADL (cvpr'19) ResNet50	BT-TT	67.71	39.80	16.84	41.44	82.67	65.11	41.63	63.13	82.51	65.97	41.70	63.39	82.51	65.11	41.63	63.08	82.64	65.19	41.92	63.25
	BT-VT	—	—	—	—	82.66	65.2	41.49	63.11	80.68	64.29	37.49	60.82	81.65	65.04	41.33	62.67	69.47	49.61	41.59	53.55
	BV-TT	70.90	44.23	20.57	45.23	82.67	65.28	41.88	63.27	82.64	65.18	41.92	63.24	82.69	65.32	41.79	63.26	82.58	65.06	41.28	62.97
	BV-VT	—	—	—	—	82.66	65.2	41.49	63.11	80.41	62.30	37.63	60.11	81.65	65.04	41.33	62.67	72.96	47.09	41.32	53.79
CCAM (wacv'20) VGG	BT-TT	73.84	49.80	24.99	49.54	77.84	60.63	38.39	58.95	77.84	60.63	38.4	58.95	77.83	60.63	38.39	58.95	72.34	52.72	31.22	52.09
	BT-VT	—	—	—	—	77.72	60.52	38.02	58.75	77.19	60.52	37.76	58.49	77.44	60.56	38.17	58.72	67.94	46.63	31.16	48.57
	BV-TT	73.84	49.80	24.99	49.54	77.84	60.63	38.39	58.95	77.84	60.63	38.4	58.95	77.83	60.63	38.39	58.95	72.34	52.72	31.22	52.09
	BV-VT	—	—	—	—	77.72	60.52	38.02	58.75	77.19	60.52	37.76	58.49	77.44	60.56	38.17	58.72	67.94	46.63	31.16	48.57
TS-CAM (iccv'21) DeiT-S	BT-TT	12.45	6.476	2.144	7.023	82.91	66.75	43.77	64.47	82.91	66.75	43.77	64.47	82.91	66.75	43.77	64.47	82.56	66.17	42.87	63.86
	BT-VT	—	—	—	—	82.74	66.65	43.77	64.38	79.95	65.12	41.18	62.08	80.42	65.32	43.55	63.09	67.53	47.63	35.03	50.06
	BV-TT	12.45	6.476	2.144	7.023	82.91	66.75	43.77	64.47	82.52	66.14	42.94	63.86	82.91	66.75	43.77	64.47	73.60	55.7	35.03	54.77
	BV-VT	—	—	—	—	82.74	66.65	43.77	64.38	80.68	64.97	42.80	62.81	80.42	65.32	43.55	63.09	66.87	46.8	34.08	49.25
SCM (eccv'22) DeiT-S	BT-TT	74.53	47.55	17.72	46.6	80.60	61.76	34.44	58.93	80.60	61.75	34.44	58.93	80.60	61.75	34.44	58.93	79.42	59.76	32.95	57.37
	BT-VT	—	—	—	—	80.48	61.76	34.42	58.88	78.61	61.15	30.44	56.73	79.90	61.38	34.27	58.51	68.54	49.27	30.88	49.56
	BV-TT	74.53	47.55	17.72	46.6	80.60	61.76	34.44	58.93	80.60	61.75	34.44	58.93	80.60	61.75	34.44	58.93	79.42	59.76	32.95	57.37
	BV-VT	—	—	—	—	80.48	61.76	34.42	58.88	79.50	60.21	34.39	58.03	79.90	61.38	34.27	58.51	68.54	49.27	30.88	49.56
SAT (iccv'23) DeiT-S	BT-TT	79.54	64.59	37.47	60.53	82.92	70.12	52.93	68.65	82.33	67.08	47.01	65.47	82.92	70.13	52.94	68.66	82.92	70.13	52.94	68.66
	BT-VT	—	—	—	—	82.91	69.46	52.82	68.39	79.96	64.78	45.18	63.30	81.19	69.05	52.25	67.49	73.13	57.06	51.02	60.40
	BV-TT	80.09	66.17	43.52	63.26	82.92	70.12	52.93	68.65	82.33	67.08	47.01	65.47	82.92	70.13	52.94	68.66	82.92	70.13	52.94	68.66
	BV-VT	—	—	—	—	82.91	69.46	52.82	68.39	79.96	64.78	45.18	63.30	81.19	69.05	52.25	67.49	73.13	57.06	51.02	60.40

consists of 399 whole slide images with two classes (normal and metastatic) used for detecting metastases in H&E-stained tissue sections of sentinel lymph nodes from breast cancer patients. Following the extraction protocol outlined in Rony *et al.* (2023), patches of size 512×512 are annotated at both image and pixel levels. This dataset comprises a total of 48,870 images, with 24,348 for training, 8,850 for validation, and 15,664 for testing. From the validation set, six fully supervised examples per class are randomly selected to determine best model, as suggested by Rony *et al.* (2023).

Evaluation measures. The GLaS and CAMELYON datasets provide pixel-wise annotations (masks) rather than bboxes, necessitating the use of pixel average precision (PxAP) Choe *et al.* (2020);

Rony *et al.* (2023) to evaluate localization accuracy. Following the standard WSOL pipeline, we first employ min-max normalization on these activation maps and apply various thresholds to the activation maps for producing localization maps.

Generation of noisy and pseudo-GT masks. The objective of this study is to evaluate the robustness of the proposed model selection techniques under conditions of noisy masks. To this end, we perturb the GT masks through erosion, simulating scenarios with noisy or imprecise GT annotations commonly encountered in practical applications. The noisy GT masks are generated by applying erosion with varying filter sizes and iterations until the masks are degraded to a specified extent, corresponding to eleven predefined noise levels, as detailed in Tab.4.9 and examples of noisy masks are illustrated in Fig.4.5.

Results. Tab.4.9 compares the performance of models (ResNet50) trained with pseudo-masks and generated pseudo-masks against those trained with GT masks, demonstrating the robustness of our evaluation protocol across varying noise levels. The results include localization performance for models using pseudo-masks generated by a pre-trained InceptionV3 classifier, an architecture distinct from the one used for training our models. Despite significant increases in mask error across different noise levels on both the GLaS and CAMELYON datasets, localization performance remains relatively stable, even at the highest noise level. Additionally, pseudo-masks generated from a different architecture show reasonable localization accuracy, further validating the generalizability of our evaluation protocol to diverse types of annotation noise, and underscoring its robustness in practical settings.

Table 4.9 Impact of varying levels of erosion in GT masks on LOC performance (PxAP) compared to GT-mask and pseudo-masks. It shows that despite substantial increases in mask error (1-AUC) across different noise levels, the LOC performance remains stable, particularly when using pseudo-masks, highlighting the robustness and generalizability of our evaluation protocol across different domains

Pseudo Mask	GLaS		CAMELYON	
	Mask Error 1-AUC	LOC PxAP	Mask Error 1-AUC	LOC PxAP
GT-Mask	0.0	69.78	0.0	30.55
Perturbed GT-Mask with erosion level:				
5%	1.86	70.09	1.64	29.74
10%	4.30	70.3	4.23	38.87
20%	9.22	70.40	9.29	29.74
30%	14.61	70.02	14.26	30.55
40%	19.15	69.83	19.17	29.74
50%	24.34	70.58	24.22	29.74
60%	29.18	67.68	29.14	29.74
70%	34.16	68.38	34.11	29.74
80%	39.21	67.98	39.17	29.74
90%	44.12	68.65	44.14	29.74
CAM - Pseudo-Mask	46.54	66.41	28.10	29.74

CONCLUSION AND RECOMMENDATIONS

5.1 Summary of Contributions

In this thesis, we make significant contributions to the domain of WSOL, focusing on improving the localization capability of models under weak supervision. Traditional WSOL approaches are inherently explainability-centric (e.g. CAMs) and try to extract activation maps from pre-trained classifiers. The classification objective of these networks forces the network to minimize mutual information among instances of the same class. Consequently, these models lead towards under-activation over different object parts, leading to map highlighting only the most discriminative regions. We address this by proposing to train localizers using pseudo-labels, thereby shifting WSOL reliance on post hoc explanations to a predictive framework. Firstly, we proposed DiPS, which enables us to train models in end-to-end fashion by utilizing pseudo labels that are harvested from self-supervised transformers. In the same realm, we proposed TeDLoc to distill localization information from vision-language to our model guided by pseudo-labels. This enables our model to produce localization maps without requiring class information during inference. Our experiments show that DiPS and TeDLoc achieved state-of-the-art performance on challenging datasets for WSOL. Beyond the development of new WSOL models, we emphasize the importance of a realistic evaluation protocol for WSOL. Current protocols often rely on GT bboxes for model evaluation, e.g model selection, hyperparameter tuning and threshold estimation. This contradicts the WSOL paradigm, leading to the overestimation of performance. To deal with this issue, we show that realistic evaluation can be done with noisy pseudo-bboxes, yielding localization accuracy comparable to methods utilizing GT bboxes. Therefore, it is the most reliable alternative to utilizing GT for model evaluation.

5.2 Limitations and Recommendations

Despite the significant advancement proposed in this thesis, there are limitations that open avenues for future research. This section provides a critical analysis of these limitations.

Relevance of WSOL in the context of foundation models. Recent advancement of foundation models, such as the segment anything model (SAM), raises questions regarding WSOL methods. SAM and similar models yield zero-shot segmentation capabilities by utilizing large-scale training data. However, SAM is not without its limitations: it lacks discriminative capability for fine-grained object categories and requires carefully crafted prompts for high performance. Moreover, SAM’s segmentations are not class-aware, and its reliance on prompts or points rather than semantic understanding restricts its deployment in localization tasks. Finally, SAM underperforms in other domains, e.g., medical, without prompt engineering or fine-tuning. We therefore foresee a *complementary* relationship, where WSOL provides class-aware cues that serve as prompts while SAM, in return, produces pseudo-masks for training WSOL models in an end-to-end fashion.

Domain shift: a realistic protocols beyond natural images. While the proposed evaluation protocol maintains competitive performance on natural image datasets, its adaptability to specialized domains such as medical imaging remains limited. To produce pseudo-labels for the validation set, we require a pre-trained model in a particular domain, e.g., a classifier. The generalization capacity of pre-trained backbone models is restricted to a particular domain. Therefore, deploying the proposal evaluation protocol to medical imaging would require model pre-training within the target domain or fine-tuning using few-shot learning techniques. Future research should investigate incorporating domain adaptation frameworks or foundation models pre-trained on large-scale, domain-specific datasets.

Scope of adaptation to weakly supervised detection and segmentation. The current research focuses on WSOL, where a tight box around an object of interest suffices. However, adapting proposed methods to other tasks such as weakly supervised object detection (WSOD) and weakly supervised semantic segmentation (WSSS) introduces new challenges. Unlike WSOL, WSOD requires detecting multiple object instances, and WSSS aims to produce a pixel-level class-aware map. For WSOD and WSSS, we must replace the localization head with a multi-class detection or segmentation head, respectively, for producing a class-aware region of interest. Moreover, the pseudo-label module must be redesigned to produce bounding boxes or segmentation masks for multiple objects, a non-trivial task under weak supervision. Without these changes, the proposed methods cannot readily transfer to other tasks.

Self-refinement using pseudo-labels from localization maps. The model does not improve through iterative self-training with its own pseudo-labels, primarily due to two factors. First, when training is based solely on the model’s own predictions, the gradients quickly diminish, causing the network to remain stuck in a local optimum. Second, since the pseudo-labels generated are inherently noisy, the model accurately connects different parts and then when it learns different object parts, it will stick with those parts. This limitation arises because pseudo-labels mainly highlight areas the model already recognizes, leaving insufficient gradient information for unexplored regions. Consequently, the model is unable to improve its performance in subsequent training iterations.

End-to-end threshold estimation. Investigating the prospect of models that can learn thresholds to produce bboxes alongside the localization map eliminates the need for an extensive threshold search. This aligns with the realistic evaluation protocols of the WSOL model, enabling the model to deploy in real-time environments.

Multi-task learning for WSOL. We also recommend the development of architectures that can be jointly optimized for classification and localization tasks, such as TeD-Loc. Such learning frames enable convergence at the same epoch across different tasks.

Relastic evaluation framework. In addition to our proposed realistic evaluation protocol for WSOL, we recommend just utilizing class-level labels during evaluations. For validation in WSOL, we should only utilize pseudo-boxes that can be generated using any off-the-shelf region proposal generators. Moreover, threshold estimation should be done using pseudo-boxes instead of using test set annotations. Finally, we should not use thresholded-IoU metrics (e.g., IoU-30, IoU-50, IoU-70) and their variants, as they provide misleading statistics by considering all scores uniform that are above a particular threshold.

APPENDIX I

DISCRIMINATIVE SAMPLING OF PROPOSALS IN SELF-SUPERVISED TRANSFORMERS FOR WEAKLY SUPERVISED OBJECT LOCALIZATION

Shakeeb Murtaza^a, Soufiane Belharbi^a, Marco Pedersoli^a, Aydin Sarraf^b, Eric Granger^a

^aDepartment of Systems Engineering, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

^bEricsson, Global AI Accelerator,
8275 Trans Canada Route, Saint-Laurent, Québec H4S 0B6

Paper published in *IEEE/CVF Winter Conference on Applications of Computer Vision, Workshop in Real-World Surveillance, Waikoloa, HI, USA*, December 2023

Abstract Drones are employed in a growing number of visual recognition applications. A recent development in cell tower inspection is drone-based asset surveillance, where the autonomous flight of a drone is guided by localizing objects of interest in successive aerial images. In this paper, we propose a method to train deep weakly-supervised object localization (WSOL) models based only on image-class labels to locate object with high confidence. To train our localizer, pseudo labels are efficiently harvested from a self-supervised vision transformers (SSTs). However, since SSTs decompose the scene into multiple maps containing various object parts, and do not rely on any explicit supervisory signal, they cannot distinguish between the object of interest and other objects, as required WSOL. To address this issue, we propose leveraging the multiple maps generated by the different transformer heads to acquire pseudo-labels for training a deep WSOL model. In particular, a new **D**iscriminative **P**roposals **S**ampling (DiPS) method is introduced that relies on a CNN classifier to identify discriminative regions. Then, foreground and background pixels are sampled from these regions in order to train a WSOL model for generating activation maps that can accurately localize objects belonging to a specific class. Empirical results on the challenging TelDrone dataset indicate that our proposed approach can outperform state-of-art methods over a wide range of threshold values over produced maps. We also computed results on CUB dataset, showing that our method can be adapted for other tasks.

1. Introduction

Due to its efficiency and flexibility, drone based surveillance has recently emerged as a feasible alternative for monitoring assets at numerous cell tower sites. Globally, millions of cell towers are being monitored for verification of assets, e.g., antennas. However, manual cell tower inspection is highly dangerous and expensive (Allred, 2022). To deploy drones for surveillance we need to localize objects that help drones to fly autonomously. Additionally, once a drone is able to identify and localize objects, it can perform different visual recognition tasks. For visual recognition, the drone should be able to fly at a safe distance from obstacles. Given the difficulty incurred in acquiring aerial images of concerned object at different viewpoints, and their associated bounding box annotations, we are unable to employ object localization models trained using supervised learning. To deal with this issue, we propose using model that can efficiently learn to localize objects of interest in a weakly-supervised manner (Choe, Han, Yun & et. al, 2021a), and efficiently guide the drone as it captures cell tower.

Commonly used methods for weakly supervised object localization are based on class activation maps (CAMs) built on top of standard convolutional neural networks (CNNs) (Belharbi *et al.*, 2022c; Choe & Shim, 2019; Lee *et al.*, 2019; Rahimi *et al.*, 2020; Singh & Lee, 2017; Wei *et al.*, 2021, 2017; Xue *et al.*, 2019; Yang *et al.*, 2020; Yun *et al.*, 2019; Zhang *et al.*, 2020c, 2018d). These methods allow producing a spatial class activation map highlighting features belonging to a particular class using features from the penultimate layer (Zhou *et al.*, 2016) of a CNN. Strong activation of a map indicates the potential presence of the corresponding class, allowing for object localization. To improve the robustness of CAM methods, several techniques have been proposed to obtain maps from different layers by utilizing gradient information (Fu *et al.*, 2020; Selvaraju *et al.*, 2017; Chattopadhyay *et al.*, 2018). Despite their popularity, CAM methods have limitations leading to inaccurate localization. Activation maps tend to focus on small discriminant areas of objects (Belharbi *et al.*, 2022c, 2023a) – common to different instances of an object belonging to a specific class. Current efforts in the WSOL literature focus on improving the CAMs to cover the full object. However, for aerial cell tower photos, regions of interest (RoI) are quite small relative to the entire images, and can be quite distant from the

drone. Hence, current CAM-based methods produce blobby results, and are unable to adequately localize objects (Belharbi *et al.*, 2022c; Murtaza *et al.*, 2022).

Recently, self-supervised transformers (SSTs) (Caron *et al.*, 2021) has attracted much attention for WSOL tasks. Using only self-supervision, these models can yield impressive saliency maps. Given their long-range receptive fields, SSTs analyze an entire input image, allowing them to build saliency maps that cover full objects. Transformers are able to identify multiple objects by decomposing the scene into different spatial saliency maps. Such localization information is accumulated in maps, *i.e.*, class tokens, at the top layer (see Fig. I-1(b)). However, without any class supervision, these tokens are arbitrary and class-agnostic, and objects of interest cannot be dissociated from others. Each token focuses on a random object with semantics that differ from one image to another, making them less reliable for localization. Unless ground truth localizations are provided to select the best token (Caron *et al.*, 2021), their application in WSOL remains limited.

Few recent work have been proposed to extract discriminative localization from class tokens. In particular, TS-CAM (Gao *et al.*, 2021) has been proposed to leverage localization in transformers, where the average of all class tokens is multiplied with a semantic-aware map to produce the final activation map. Since TS-CAM averages an abundance of class-agnostic attention maps, it is prone to localization error by including non-discriminative regions. Additionally, maps from earlier layers attend to background regions, as shown in Fig.I-1(a). Therefore, accumulating all maps introduces noise in the final activation map that can reduce the model's localization accuracy. Different transformer-based methods have been proposed (Bai *et al.*, 2022; Chen *et al.*, 2022; Gupta *et al.*, 2022; Su *et al.*, 2022). (Bai *et al.*, 2022) propose a spatial calibration module to capture semantic relationships and spatial similarities. Similarly, ViTOL (Gupta *et al.*, 2022) proposes to incorporate a patch-based attention dropout layer into the transformer attention blocks to improve localization maps. (Chen *et al.*, 2022) proposes a relational patch-attention module to enhance local perception quality, and to retain global information. (Su *et al.*, 2022) introduces a mechanism to suppress background regions and focus on the target object. All these methods are unable to minimize the loss over the localization maps as they solely focus on

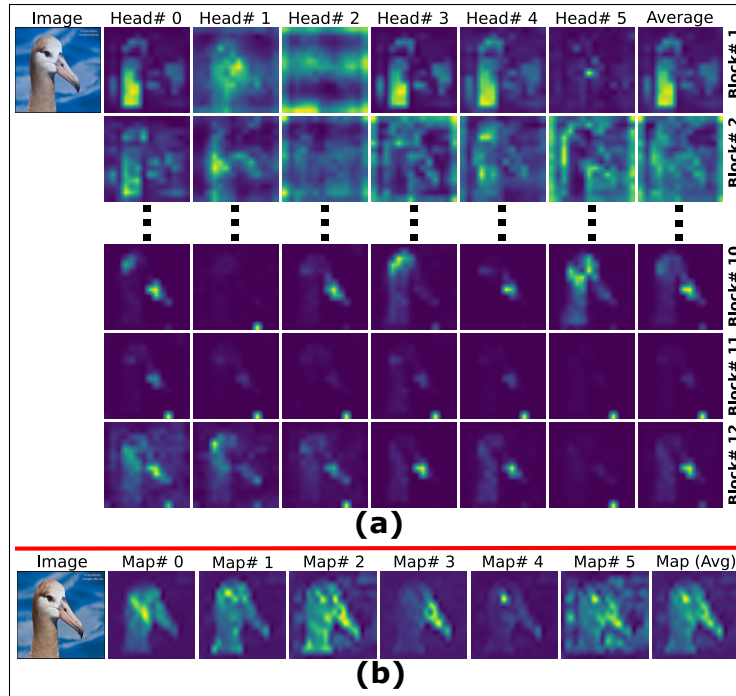


Figure-A I-1 (a): Attention of each class token learned by TS-CAM (Gao *et al.*, 2021) map from the first two and last three blocks. The first half of the blocks attends to different features, including background regions, but TS-CAM is able to accumulate all of them. Attention maps of each class token are shown in Supplementary Material. (b): The attention map of class tokens corresponding to each token head from the last layer of transformer learned in a self-supervised fashion. These maps show that the last layer is capable of accumulating fine-grained localization of the concerned object

classification tasks, limiting their ability to fully cover an object while requiring a search for an optimal threshold.

In this paper, a new Discriminative Proposals Sampling (DiPS) method is proposed to leverage localization information from SSTs for accurate WSOL in aerial images captured by drones. During training, class token maps are extracted from the top block of pretrained SST. Given a limited number of high-quality attention maps (class tokens) covering different objects, we

employ a CNN classifier to localize potential discriminative regions (see Fig. I-1(a)). Then, pixel-wise pseudo-labels are sampled from these regions to train a U-Net style localization network. To benefit from these tokens, a method is introduced for collecting appropriate pseudo-labels, and sampling foreground/background regions from them to train our localization network such that a particular object can be localized with a high level of confidence (Fig.I-2). In particular, with the help of a CNN classifier, sampling areas are identified for building effective pseudo-labels by determining foreground and background regions in class tokens obtained from SST. From these areas, foreground pixels are sampled, while the entire image is used to sample pixels from background regions according to the criteria defined in Section 3. Using these pseudo-pixels, the localization network is trained by optimizing partial cross-entropy over selected pixels. Additionally, the localization network parameters are regularized by the classifier response to ensure consistency between class prediction and localized area. CRF loss (Tang *et al.*, 2018) is employed to yield localization with accurate boundaries. During inference, we retain only the localization network to produce of localization maps, and the CNN classifier to predict the class.

Our DiPS method allows producing localization maps with same size as the input image. Existing transformer and CNN-based methods can only produce low-resolution maps which introduce more localization error. For CNN-based methods, the class activation map is approximately $8\times$ smaller than the input image, and requires interpolation that adds a bloby effect. In contrast to CNNs, which provide a map of size 29×29 , the transformer-based methods can produce a map of size 14×14 (with standard architecture) for an input image of 224×224 . Producing low-resolution maps hinders the performance of localization methods, and we cannot go beyond a threshold in terms of localization accuracy. These maps are unable to encompass precise details about the concerned object of interest. Also, other methods that use activation maps as hard pseudo labels are only able to focus on representative parts of an object which hinders their performance. In contrast, our method selects few pixels as pseudo label that prevent the model from learning false-positives from underlying CAMs.

Our main contributions are summarized as follows.

- (1) A new DiPS method is proposed to leverage the emerging saliency maps in SST to localize cell towers in aerial images. Since maps from SSTs lack discriminative information, a pretrained CNN classifier is employed to produce discriminative proposals based on saliency maps from the top layer of pretrained SST to harvest efficient pseudo-labels for the training of our localizer.
- (2) Instead of minimizing the classification loss, DiPS samples the pseudo labels, allowing to minimize the loss over localization maps.
- (3) DiPS can efficiently infer statistical properties (e.g., size, boundaries) of the object learned from low-resolution maps, resulting in a high-quality localization map.
- (4) An extensive set of experiments was conducted on two challenging WSOL datasets – (i) TelDrone, a private dataset containing aerial images from various cell tower sites, and (ii) CUB-200-2011 (Welinder *et al.*, 2010). The proposed method outperformed state-of-art methods in terms of localization accuracy, with less sensitivity to threshold values. Visual results also show that our method produces CAMs with a better coverage of the entire foreground regions, and a clearer distinction between foreground and background regions.

2. Related Work

Class Activation Mapping (CAM) Methods: A common way to harvest localization maps from network activations is by aggregating the information presented in a specific CNN layer. The representative method for aggregating activations is using CAM methods (Zhou *et al.*, 2016), which weighs each pixel according to its influence on the class prediction. Several methods have been proposed to improve the mechanism for harvesting the activation maps (Chattopadhyay *et al.*, 2018; Fu *et al.*, 2020; Ramaswamy *et al.*, 2020; Selvaraju *et al.*, 2017). Networks used for generating CAMs are solely trained for image classification, and thus focus on discriminative regions. They typically underestimate the object size. To mitigate this issue, (Singh & Lee, 2017; Yun *et al.*, 2019) employ adversarial perturbation to erase discriminative parts, and for the

network to look beyond the discriminative areas. Similarly, in (Choe & Shim, 2019; Zhang *et al.*, 2018c) discriminative features are erased, and adversarial learning is adopted to encroach upon non-discriminative regions of the concerned object. In addition, fusion-based methods improve on CAM methods by combining different activation maps based on the classifier's response (Naidu *et al.*, 2020; Wang *et al.*, 2020b,a).

Different model-dependent techniques have been proposed to alleviate the poor coverage of objects (Lee *et al.*, 2019; Rahimi *et al.*, 2020; Xue *et al.*, 2019; Yang *et al.*, 2020; Zhang *et al.*, 2020c). In (Wei *et al.*, 2018), authors utilized CAMs from multiple convolution layers with different receptive fields to enlarge the discriminative regions. (Yang *et al.*, 2020) combines CAMs of different classes to identify foreground and background regions in an image. (Wu *et al.*, 2022; Zhu *et al.*, 2021) propose to suppress background regions to help to generate a foreground map with high confidence, guided by area constraints. For a robust training, (Meng *et al.*, 2021) incorporates an encoder-decoder layer between the shallow layers, and a generator to mask a part of the image. Low-level feature based activation map (FAM) (Xie *et al.*, 2021) utilizes multiple classifiers to generate a foreground map that is decomposed into multiple part regions. This is used to train a network to produce a final semantic-agnostic map. Self-produced guidance (SPG) (Zhang *et al.*, 2018d) separates foreground and background regions for guiding shallow learning while expanding to less discriminative areas. Shallow feature-aware pseudo supervised object localization (SPOL) (Wei *et al.*, 2021) employs a multiplicative fusion strategy for harvesting highly confident regions, and then uses those masks as pseudo-labels for training a segmentation network. Inter-image communication (I^2C) (Zhang *et al.*, 2020e) increases the robustness of localization maps by considering the correlation of similar images within a class. Structure-preserving activation (SPA) (Pan *et al.*, 2021) seeks to preserve the object's structure within a CAM, although this method still produces blobby boundaries. Similarly, Strengthen learning tolerance (SLT) (Guo, Han, Wan & Zhang, 2021) groups images of similar classes to increase the network's tolerance, and force the localization map to expand towards the object's boundaries. Also, the full resolution CAM (F-CAM) (Belharbi *et al.*, 2022c) has been proposed to expand activation maps to cover full object beyond discriminative regions.

Compared to F-CAM, our proposed method is able to define sampling regions for building efficient pseudo-label with the help of classifier. This helps our model to focus only on target object instead of expanding the activation to other objects in the concerned image.

Most of the above methods rely on internal activations acquired by utilizing activation maps at different layers of the networks. Given their intrinsic properties (e.g., local receptive field), CNNs decompose an object into local semantic elements (Bau *et al.*, 2017; Zeiler & Fergus, 2014). These intrinsic properties prevent the CNN from forming global relationships between different receptive fields, limiting their ability to localize whole object, and producing coarse blobby maps focused on discriminant regions. One solution to form long-range dependencies is to find global cues by calculating pixel-similarities (Wang *et al.*, 2018c, 2020c; Zhang *et al.*, 2020c,e). More recently, (Minsong *et al.*, 2020) proposed a method based on non-local attention blocks to build long-range relationships and efficiently localize objects. It enhances attention maps by incorporating spatial similarity.

Vision Transformer (ViT) Methods: Numerous transformer based methods have recently been proposed for WSOL (Bai *et al.*, 2022; Chen *et al.*, 2022; Gupta *et al.*, 2022; Li, 2022; Su *et al.*, 2022). A pioneering work for object localization is the token semantic coupled attention map (TS-CAM) (Gao *et al.*, 2021), which is capable of localizing objects at a fine-grained level by fusing attention maps with semantic-aware maps. (The detailed working TS-CAM model is presented in supplementary material.) Recently, (Su *et al.*, 2022) proposed a re-attention mechanism for suppressing background regions. Layer-wise relevance propagation (Gupta *et al.*, 2022) and clustering-based (Chen *et al.*, 2022) methods have also been proposed to accumulate attention maps produced by ViTs. For instance, (Gupta *et al.*, 2022) introduces a patch-based attention dropout layer in the attention block, and incorporate an attention roll-out method to improve localization performance.

Transformer-based methods have had a significant impact on WSOL literature. However, they accumulate all attention maps from each layer, leading to a noisy CAM, or a suppression of different regions within the object (Gao *et al.*, 2021) (see Fig.I-1). Most techniques provide a heat

map without sharp boundaries. In contrast, our approach can produce high-resolution maps with sharper boundaries, only using attention from the last layer of the transformer block. Methods based on pseudo-labels fix their map for all epochs (Wei *et al.*, 2021), generating a map very close to the pseudo-labels. Our method samples only a few pixels per epoch to learn foreground and background maps that continue to evolve based on a bi-nominal distribution. Sampling only a few pixels as pseudo-labels helps the network to explore different parts of the object, covering the whole object with sharper boundaries. Since these methods are typically computationally intensive during training, we harvest maps in a self-supervised way using a pretrained transformer, allowing us to obtain clues about potential objects and train our localization network. Our method leverages long-range dependencies as pseudo-labels are harvested from a transformer, and convolution inductive bias to suppress background noise. Finally, in contrast with WSOL baselines, our method focuses on drone-based WSOL for surveillance of cell tower sites. These images are captured at a long distance, and the object of interest covers a small proportion of the image.

3. Proposed Method

3.1 Background

Vision transformers (ViTs) are recognized for their success in image classification (Dosovitskiy *et al.*, 2020; H. Touvron *et al.*, 2021) and WSOL (Gao *et al.*, 2021). They consist of T cascaded encoder blocks, each containing multi-headed attention, followed by a multi-layer perceptron (MLP). First, an input image x of size $W \times H$ is divided into N patches of resolution $(W/S) \times (H/S)$, where S denotes the patch size in pixels. These patches are then linearly projected to a fixed embedding size D along with a learnable class token and passed through T cascaded blocks. Furthermore, the features of a class token from T^{th} block are fed to the MLP for class prediction, defined as $p = \text{Softmax}_{\tau=1}(\text{MLP}(\hat{T}_{vit}(x)))$, where

$$\text{Softmax}_{\tau}(s) = \frac{\exp(s/\tau)}{\sum_{k=1}^K \exp(s_k/\tau)} \quad (\text{A I-1})$$

where $s \in \mathbb{R}^K$ is the MLP output, K corresponds to the number of classes, and $\hat{T}_{vit}$ represents the transformer’s forward pass, generating features f that are passed through T cascaded blocks. The MLP predicts the probabilities of concerned classes with cross-entropy classification loss $-\sum_y y \log(p_y)$, and y is the ground truth label.

Self-Distillation with *no* labels (DINO) involves harvesting regions of interest without class-level labels. To achieve this goal, the authors trained two networks, called student and teacher, to match their probability distribution. Parameters of the teacher are an exponential moving average of the student network. The probability p of the representation for these networks is calculated by normalizing the output of the concerned network using the softmax function with temperature defined in Eq. 3.1 (τ is a hyper-parameter to be optimized). Furthermore, the input x is augmented in different ways, x_1 and x_2 , for the student and teacher networks, respectively. They represent two set of distorted views of image x , calculated using the strategy defined in (Caron *et al.*, 2020). The first set contains several local views, while the second one contains two global views. Here, local views contain less than 50% of the original image, and global views cover more than 50% of the area. Finally, the loss is calculated as cross-entropy between the output probability of the student p^s and the teacher p^t :

$$\min_{\theta_s} -p^s(x_1) \log(p^t(x_2)). \quad (\text{A I-2})$$

3.2 Overview of DiPS

As we discussed, SSTs (Caron *et al.*, 2021) are able to localize all objects in an image. They decompose the objects in an image into different maps because they learn to represent objects in a self-supervised fashion. Each map highlights different objects or their parts without associating categorical information with them. If we can successfully identify maps and corresponding regions that contain the concerned object, then we can use them as pseudo-labels to train a localization network. Therefore, we propose a new DiPS method that can harvest efficient pseudo-labels on the target object for accurate WSOL. DiPS utilizes class tokens of a pretrained

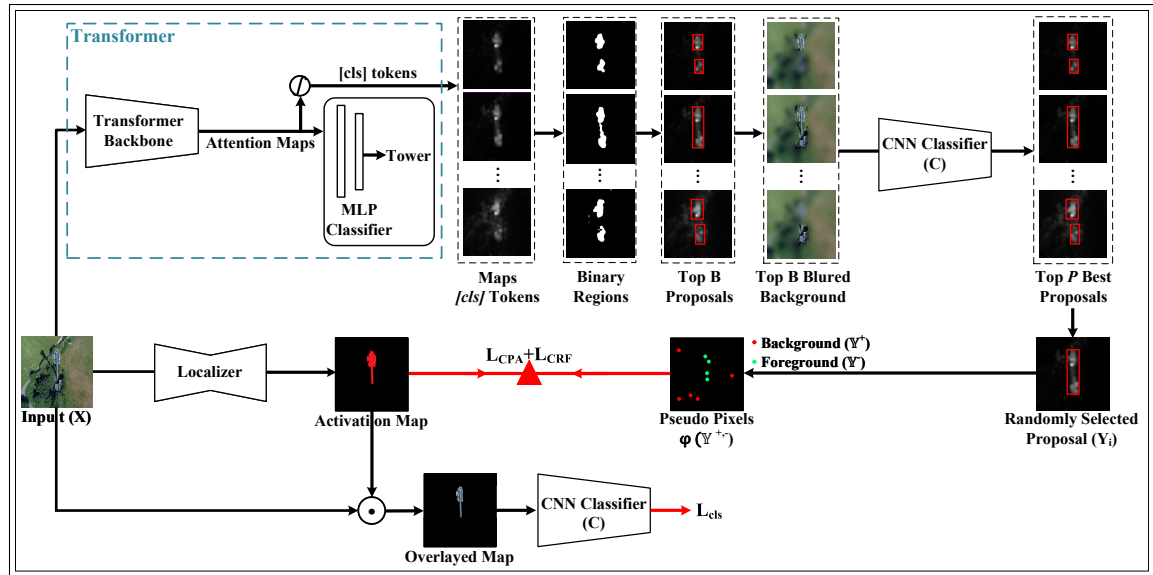


Figure-A I-2 Architecture for training a deep WSOL network using the DiPS method. A pertained transformer is employed along with a classification layer with frozen weights.

After obtaining attention maps, a threshold is applied to produce binary proposals and generate bounding boxes for each proposal. Based on those proposals, areas outside the box are blurred, and the top P best bounding boxes are selected based on the classifier's score that are greater than a minimum score. Ultimately, a bounding box is selected among the P best boxes for sampling from background and foreground regions. Then, the generated pseudo-pixels are considered as pseudo-labels to training of the localization network. After training, an input images are processes localization network without the need of a transformer's backbone

ViT to obtain effective pseudo-labels and a pretrained CNN classifier to select ROIs proposals to train our localization network (see Fig.I-2). It allows us to directly minimize the loss over the generated map for accurate localization maps with a similar confidence for all object parts.

Given a set of tokens from a pretrained SST model (Caron *et al.*, 2021), the associated maps are first binarized, allowing to extract bounding boxes over potential ROIs. They are considered proposal candidates that potentially cover object regions with varying certainty. To measure certainty, the response of a pretrained CNN classifier is employed to indicate the most relevant ROI. To avoid overfitting to a single proposal, we consider top- P confident ROI proposals, and randomly select one of them for sampling. This final proposal is used to constrain sampling

of pixel-wise pseudo-labels where foreground pixels are sampled inside the bounding box of the proposals. Sampling is guided by the activation magnitude, where strong activations are favored to be foreground. Background pixels are sampled outside the bounding box by favoring low activations. Pseudo-labels are sampled randomly at each Stochastic Gradient Descent (SGD) iteration, which has been shown to be more effective and robust against overfitting than static and noisy pseudo-labels (Belharbi *et al.*, 2022a,c). Overall, this sampling allows for the emergence of accurate localization maps, from a few pixels toward segments and the entire object. This can be seen as a *fill-in-the-gap* game where only random pixels are selected as foreground/background, and the localization network must generalize to similar regions. Then, over iterations, this random sampling is expected to stimulate convolution filters to have a similar response to visually similar nearby pixels.

The collected pixel-level pseudo-labels are used to train a U-Net style localization network (Ronneberger *et al.*, 2015), as illustrated in Fig.I-2. It takes the image as input and yields foreground and background CAMs for localization. To train this model, we consider a composite loss that leverages local and global constraints. Local constraints aim to guide learning at pixel level. To this end, standard partial cross-entropy is used to exploit sampled pixel-wise pseudo-labels. To produce a consistent CAM that is well aligned with object boundaries, we also include a Conditional Random Field (CRF) loss (Tang *et al.*, 2018). To further ensure that the localized object is well aligned with the image-class label, we add a global constraint over the foreground CAM. In particular, we feed the product of the image by the foreground CAM to the pretrained classifier. Then, the classifier response is maximized with respect to the true image-class. During training, only the weights of the localization network are optimized. The SST and classifier models are frozen. At the end of the training, only the localization network and classifier models are retained for classification and localization tasks. The rest of this section provides more details on DiPS components.

3.3 Training Architecture

In this paper, our goal is to localize the object of interest through an encoder-decoder localization network that is trained using pseudo-labels. These pseudo-labels are obtained from `class` token of the different heads from the last layer of the transformer. We represent the training set by $\mathbb{T} = \{x_i, y_i\}$ and $x \in R^{W \times H}$ represents an image with a size of $W \times H$ and its corresponding label of K classes is represented by $y \in \{1, \dots, K\}$. DiPS employs a mechanism to sample background and foreground regions from activation maps (`class` token) to build effective pseudo-labels, as shown in Figure I-2. To achieve these goals, we propose a model that consists of three modules (i) transformer T_{SST} that was trained in self-supervised fashion (as explained above) for producing an attention map (ii) localization network F_θ for producing soft-max activation maps represented by $M = F_\theta(x) \in [0, 1]^{n \times m \times 2}$; here first channel represent the foreground M_1 and other represent the background map M_2 trained using best proposal extracted from `class` tokens (iii) a classification layer $\hat{C}$ that is trained on top of transformer’s features that is used to calculate additional matrices that involves class accuracy (see supplementary material).

Selection of Background and Foreground Pixels: In order to train the localization map predictor a pixel-level supervisory signal is employed. At each epoch, new pseudo-labels are generated that consist of a few foreground and background pixels (pseudo-pixels) to increase network certainty of different regions during training. To select pseudo-pixels, we extract attention maps from the last layer of the transformer, corresponding to `class` tokens. From the maps, we select only the first four maps and an average map of all `class` tokens. After this, we apply the Otsu (Otsu, 1979) thresholding method for converting the attention map into a binary map to obtain all connected regions that are greater than a minimum size¹ to extract all B proposal (bounding) boxes enclosing those regions. Then, for an input image, we produce B images corresponding to each bounding box by applying the Gaussian-blur filter to hide the background regions outside of the respective bounding box. Each image is then passed to a classifier C , and the top P best proposals along with their underlying attention maps are selected based on the classifier’s confidence corresponding to target class y_i . Then, one out of P

¹ The minimum size is an hyperparameter value.

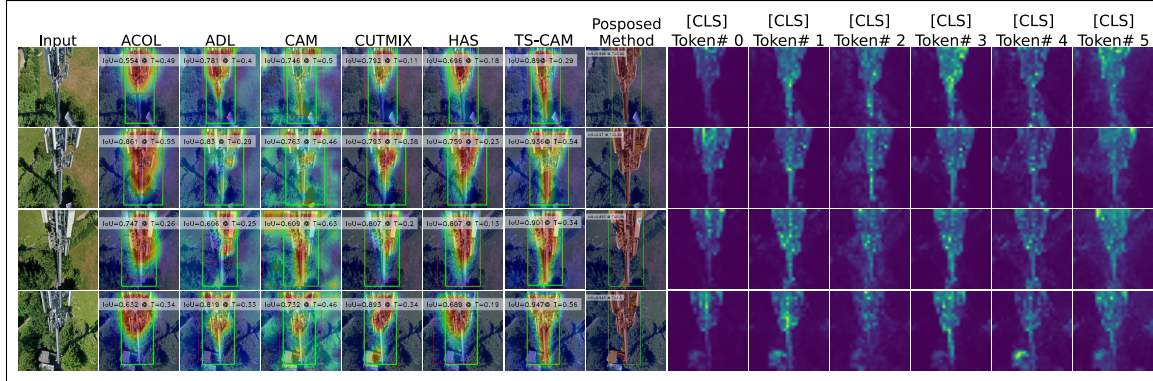


Figure-A I-3 Visualization of test samples from TelDrone. Results of baseline method including TS-CAM shows that these method can only cover some parts of the object and invisible parts are also included in the bounding box due to extensive search of thresholds. In contrast, our method learns to predict map that can cover the whole object. Moreover, maps in red box are used to sample pseudo-labels and our methods learns to localize the object precisely and able remove noise from them

proposals along with its respective bounding box is selected for pseudo labeling. The pixels inside and outside of the bounding box are considered as foreground $\mathbb{Y}_i^+$ and background regions $\mathbb{Y}_i^-$ respectively defined as:

$$\mathbb{Y}_i^+ = \varphi^+(Y_i, n^+) \quad \mathbb{Y}_i^- = \varphi^-(Y_i, n^-) \quad (\text{A I-3})$$

where Y_i set of all pixel in the generated map, $\mathbb{Y}_i^+$ represents the top $n\%$ pixels within the bounding box selected through multinomial sampling and $\mathbb{Y}_i^-$ represents top $-n\%$ pixels anywhere that

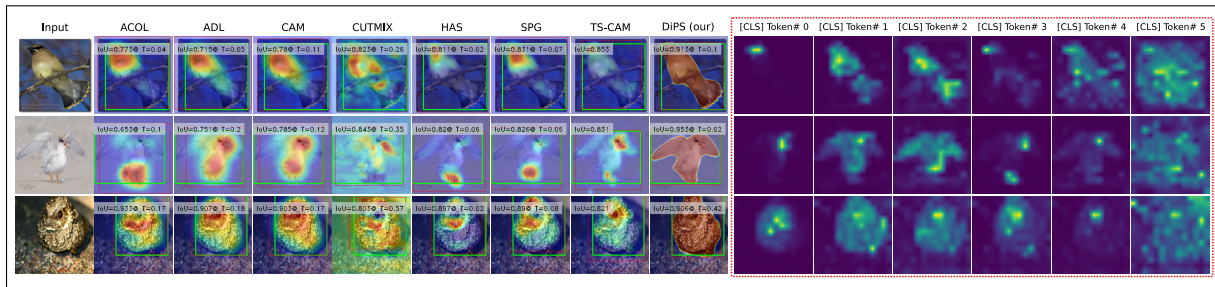


Figure-A I-4 Visualization of test samples from CUB dataset

are derived using uniform sampling from the activations of whole image sorted in reverse order. These pixels are derived from unreliable attention maps and may contain incorrect labels. To deal with this uncertainty, we select only a few (see experimental details for exact values) pixels from the foreground and background while rejecting others (Singh & Lee, 2017; Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). Therefore, the final set of foreground and background pixels for calculating loss are:

$$\varphi(\mathbb{Y}_i^{+, -}) = \varphi(\mathbb{Y}_i^+) \cup \varphi(\mathbb{Y}_i^-) \quad (\text{A I-4})$$

here $\varphi(\mathbb{Y}_i^+)$ and $\varphi(\mathbb{Y}_i^-)$ represent set of pixels (hyper-parameter) sampled using multinomial distribution. At the end, we then generated pseudo label $Y \in \{0, 1\}^2$ by assigning 1's to pixel in $\mathbb{Y}_i^+$ and 0's to pixels in set $\mathbb{Y}_i^-$.

Overall Training Loss: Our training loss consists of three terms; (i) classifier's loss to ensure the integrity of generated map. The generated map M_1 is overlaid on the image x and passed it to the classifier to compute a cross-entropy loss; $\mathcal{L}_{CLS} = -\sum_{k=1}^K y_k \log(\text{Softmax}(C(x \odot M_1)))_k$. (ii) Our constrained pixel alignment loss² for learning foreground/background regions. It aligns the output map M with the selected pixels in $\varphi(\mathbb{Y}_i^{+, -})$ through partial cross-entropy denoted by $\mathcal{L}_{CPA}(\varphi(\mathbb{Y}_i^r), M^r)$; here r represent the selected pixels. (iii) Conditional Random Field (CRF) (Tang *et al.*, 2018) to align the localization map with the object boundaries. The detailed description of the CRF loss $\mathcal{L}_{CRF}$ is presented in supplementary material. Furthermore, the overall can be formulated as,

$$\mathcal{L}_{Total} = \min_{\theta} \lambda_{CLS} \mathcal{L}_{CLS} + \lambda_{CPA} \mathcal{L}_{CPA} + \lambda_{CRF} \mathcal{L}_{CRF} \quad (\text{A I-5})$$

where λ_{CLS} and λ_{CPA} are hyperparameters between interval $[0, 1]$, and λ_{CRF} is set to $2e^{-9}$ as defined in (Tang *et al.*, 2018).

² Loss using few pixels was also employed by (Belharbi *et al.*, 2022c).

4. Results and Discussion

4.1 Experimental Methodology

Datasets: For validation, we employed two datasets: CUB-200-2011 and an internal dataset named TelDrone. **CUB200-2011** is one of the most popular datasets used for visual classification and object localization tasks. This dataset consists of 11,788 images divided into 200 categories; 5,794 for testing and 5,994 for training (Welinder *et al.*, 2010). For validation and hyperparameter search, we employed an independent validation set collected by (Choe *et al.*, 2020) containing 1,000 images. **TelDrone** contains 915 4K images collected using a drone that orbits around the tower site. Furthermore, images are divided into two classes (either containing an inspection site or not). From this dataset, we considered 797 images for training, 13 images for validation and 105 images as a test set.

Evaluation measures: We analyze our results based on the evaluation measure suggested in (Choe *et al.*, 2020) – **MaxBoxAccV2** refers to the proportion of predicted bounding boxes that have an IoU greater than a particular threshold concerning the generated map. It is averaged over three different IoU thresholds $\delta \in \{30\%, 50\%, 70\%\}$. We also reported additional localization matrices in supplementary material as presented in (Choe *et al.*, 2020).

Implementation details: We follow the protocol in (Choe *et al.*, 2020) for all experiments. A batch size of 32 is used for all datasets. For training on CUB, images are resized to 256×256 , and then randomly cropped to 224×224 and randomly flipped horizontally as in (Choe *et al.*, 2020). Moreover, for the TelDrone dataset, we first resize images to 512×512 followed by random cropping of size 448×448 . We trained our model for 50 epochs while the learning rate is decayed by a factor of 0.1 after every 15^{th} epoch for both dataset.

Baseline Models: To evaluate the performance of our model, we compared it with several state-of-the-art methods presented in Tables I-1. We report results from (Choe *et al.*, 2020) for CAM (Zhou *et al.*, 2016), HaS (Singh & Lee, 2017), ACoL (Zhang *et al.*, 2018c), SPG (Zhang *et al.*, 2018d), ADL (Choe & Shim, 2019), and CutMix (Yun *et al.*, 2019).

For all other models, we report the same results as published in their respective articles. For qualitative evaluation, we reproduced the results of CAM (Zhou *et al.*, 2016), HaS (Singh & Lee, 2017), ACoL (Zhang *et al.*, 2018c), SPG (Zhang *et al.*, 2018d), ADL (Choe & Shim, 2019), CutMix (Yun *et al.*, 2019) and TS-CAM (Gao *et al.*, 2021) according to the protocols defined in (Choe *et al.*, 2020).

4.2 Comparison with State-of-Art Methods

Table I-1 shows that DiPS achieves state-of-the-art performance on CUB and TelDrone datasets for **MaxBoxAccV2** metric compared to the other methods in the literature. The visual results of our method along with the corresponding baselines on TelDrone and CUB datasets are presented in Fig.I-3 and Fig.I-4 respectively. The baseline method tends to focus on discriminative regions and expands to less discriminative regions because of the blobby nature of these maps. However, the selection of the optimal threshold allows the less discriminative area to be included in the localization map because of their texture or color similarity. Furthermore, the resultant map becomes unreliable and unable to identify the whole object as it encompasses regions with no concealed activation over an object. For CUB dataset, we compare the results with the baseline methods (Fig.I-4) including TS-CAM and show that our method is capable of covering the whole object with the same intensities instead of highlighting different parts of the object. Additionally, the class tokens of SST are also presented, and DiPS is able to successfully harvest useful information from them by constructing a reliable pseudo-label. Similarly, on the TelDrone dataset, DiPS achieved state-of-the-art visual results by localizing all parts of the object with the same confidence instead of hot-spotting different regions (Fig.I-3). More specifically, current state-of-the-art methods (e.g. TS-CAM) focus on some regions of the concerned object by hot-spotting different parts. Due to this they are able to properly draw a bounding box around the object due to extensive threshold search that can include low scoring areas to the localization map. In contrast, our method can produce a map that covers the whole object with sharper boundaries, which eliminates the need for a precise threshold value. Additionally, we compare the results of our model with SST's activation and show that our method learns to localize well even with noisy pseudo-labels.

Table-A I-1 MaxBoxAccV2 performance on the TelDrone and CUB

Methods	TelDrone (MaxBoxAccV2)	CUB (MaxBoxAccV2)
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr</i> ,2016)	55.9	63.7
HaS (Singh & Lee, 2017) (<i>iccv</i> ,2017)	60.3	64.7
ACoL (Zhang <i>et al.</i> , 2018c) (<i>cvpr</i> ,2018)	59.1	66.5
SPG (Zhang <i>et al.</i> , 2018d) (<i>eccv</i> ,2018)	67.3	60.4
ADL (Choe & Shim, 2019) (<i>cvpr</i> ,2019)	66.0	66.3
CutMix (Yun <i>et al.</i> , 2019) (<i>eccv</i> ,2019)	57.2	62.8
ICL (Minsong <i>et al.</i> , 2020) (<i>accv</i> ,2020)	–	63.1
TS-CAM (Gao <i>et al.</i> , 2021) (<i>iccv</i> ,2021)	72.2	76.7
CAM-IVR (Kim <i>et al.</i> , 2021) (<i>iccv</i> ,2021)	–	66.9
PDM (Meng <i>et al.</i> , 2022a) (<i>tip</i> ,2022)	–	72.4
C ² AM (Xie <i>et al.</i> , 2022b) (<i>cvpr</i> ,2022)	–	83.8
ViTOL-GAR (Gupta <i>et al.</i> , 2022) (<i>cvpr</i> ,2022)	–	72.4
ViTOL-LRP (Gupta <i>et al.</i> , 2022) (<i>cvpr</i> ,2022)	–	73.1
TRT (Su <i>et al.</i> , 2022) (<i>corr</i> ,2022)	–	82.0
BGC (Kim <i>et al.</i> , 2022) (<i>cvpr</i> ,2022)	–	80.1
SCM (Bai <i>et al.</i> , 2022) (<i>eccv</i> ,2022)	–	89.9
CREAM (Xu <i>et al.</i> , 2022) (<i>cvpr</i> ’22)	–	73.5
F-CAM+CAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv</i> ’22)	–	79.4
F-CAM+LayerCAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv</i> ’22)	–	80.1
DiPS (ours)	84.9	90.9

Analysis of distribution shift. We present the effect of varying the threshold values on the localization performance along with the distribution spread of foreground and background regions. The change in MaxBoxAcc at different threshold for CUB test set is presented in Figure I-5. We compare our results with CAM (Zhou *et al.*, 2016), HaS (Singh & Lee, 2017), ACoL (Zhang *et al.*, 2018c), SPG (Zhang *et al.*, 2018d), ADL (Choe & Shim, 2019), and CutMix (Yun *et al.*, 2019). For all of these related methods, we can see that the MaxBoxAcc quickly drops to zero as the threshold increases. Due to this, it is very difficult to find the optimal threshold for each image which makes these methods unfeasible to deploy in practical scenarios. In contrast, the output of our method is less susceptible to the threshold.

5. Conclusion

In this paper, we proposed a DL method for WSOL, capable of producing fine-grained object localization maps of cell tower sites in aerial images for assets surveillance. The proposed method is capable of learning from pseudo-labels instead of only class labels. These pseudo-labels are harvested by sampling foreground and background areas from the `class` token of an SST. Our method is able to generate better maps compared to the `class` token used to build pseudo-labels.

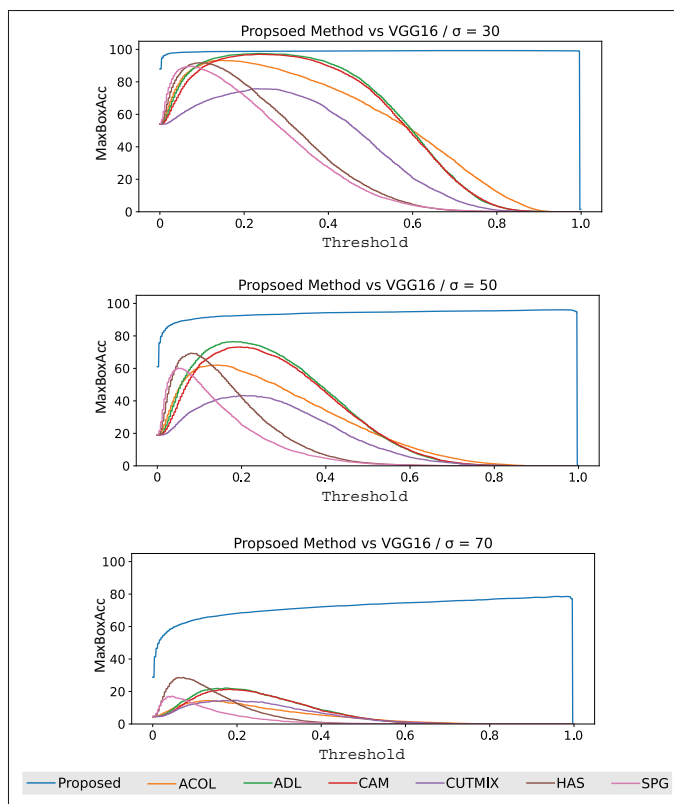


Figure-A I-5 MaxBoxAcc performance at different IoU threshold values for output map using the CUB dataset

The proposed method is capable of generating reliable maps, thus providing better coverage of the whole object. To validate the performance of our model, we compared its results with the existing baseline method, and our method achieved competitive performance both qualitatively and quantitatively. Compared to the baseline, our method can identify all parts of the object instead of decomposing and highlighting different parts of the objects. Furthermore, the proposed method eliminates the need for an extensive threshold search to produce an optimal bounding box covering the concerned object.

Note: Additional details are provided in the supplementary materials, including additional visual results, an evaluation matrix for error dissection, an extended error analysis, ablation studies, and an overview of the CRF loss.

Acknowledgments: This research was supported by MITACS and the Natural Sciences and Engineering Research Council of Canada. We also acknowledge Compute Canada for the computing resources.

6. Supplementary Material

This section contains the following material:

- 6.1 Performance evaluation measures and error dissection;
- 6.2 an overview of CRF loss;
- 6.3 additional error analyses;
- 6.4 ablation studies;
- 6.5 additional visual results;
- 6.6 details on the TS-CAM method.

6.1 Performance Measures and Error Dissection

6.1.1 Evaluation Measure for Error Dissection

In this section, we present the evaluation measures that are used in (Gao *et al.*, 2021) for error dissection over wrong predictions. These measures are useful for deciding threshold values for producing bounding boxes from localization maps. Specifically, localization part error (LPE) and localization more error (LME) help in deciding whether to increase or decrease the threshold value for optimal results. More details on error measures are given below:

Localization Part Error (LPE): This measure identifies that an object partially detected by the localization map with a large margin has an intersection over the predicted bounding box (IoP) > 0.5 .

Localization More Error (LME): It indicates that the predicted bounding box is larger than the actual box and covers other objects or background. This can be identified if intersection over annotated-bounding-box (IoA) > 0.7 .

6.1.2 Additional Performance Measures

Top-1 Localization Accuracy (Top-1 Loc): A prediction is considered true if the predicted class is the same as the ground truth and the intersection over Union (IoU) is greater than 0.5.

Top-5 Localization Accuracy (Top-5 Loc): A prediction is considered true if the IoU is greater than 0.5 and the actual class matches at least one of the top 5 predicted classes.

6.2 Overview of CRF loss

Conditional random fields (CRF) loss, aligns the predicted localization map M with the boundaries of a concerned object presented in input x . CRF loss (Tang *et al.*, 2018) between x and M can be defined as follows:

$$\mathcal{L}_{CRF}(A, M) = \sum_{i=0}^{i=1} M_i^T A (1 - M_i) \quad (\text{A I-6})$$

where A represents an affinity measure that contains mutual similarities between pixels, including proximity and color information. For capturing affinities of pixels, we use a Gaussian kernel (Krahenbuhl & Koltun, 2011) and employ permutohedral lattice (Adams, Baek & Davis, 2010) to reduce the computation overhead.

Table-A I-2 Extended error analysis on the CUB-200-2011 dataset

	LPE ↓	LME ↓
VGG16	21.91	10.53
InceptionV3	23.09	5.52
TS-CAM (Gao <i>et al.</i> , 2021)	6.30	2.85
DiPS (our)	0.05	0.07

6.3 Extended Error Analysis

Further error analysis (according to the error measures defined in Section 6.1.1) on the CUB datasets is presented in Table I-2. Our method localized the correct region of the concerned

object instead of overestimating or underestimating the region. It also shows that the maps generated by DiPS are very robust and have much fewer errors compared to the baseline methods. The statistics of the baseline methods are from (Gao *et al.*, 2021).

6.4 Additional Ablation Studies

The performance of DiPS with various loss function combinations is shown in Table I-3. It shows that all of the auxiliary losses contribute significantly towards the final performance. Also, training through arbitrary selection of pixels (pseudo-labels) rather than the classifier loss or fixed pseudo-labels allows DL models to explore different regions of an object and can provide accurate localization. Adding CRF and classification terms at the same time significantly improves the performance of our model. The **MaxBoxAcc** of our model on TelDrone is presented in table I-4. Furthermore, the **MaxBoxAcc**, **top-1** and **top-5** localization accuracy for CUB dataset is presented in Table I-5. We achieved state-of-the-art performance on the TelDrone and CUB dataset.

Table-A I-3 Localization performance of our DiPS method with different losses

	CUB (MaxBoxAcc)	TelDrone (MaxBoxAcc)
$\mathcal{L}_{CPA} + \mathcal{L}_{CRF}$	95.4	93.3
$\mathcal{L}_{CPA} + \mathcal{L}_{CLS}$	94.6	91.7
$\mathcal{L}_{CPA} + \mathcal{L}_{CRF} + \mathcal{L}_{CLS}$	97.0	96.2

Table-A I-4 **MaxBoxAcc** performance on the TelDrone dataset

	MaxBoxAcc
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr,2016</i>)	50.9
HaS (Singh & Lee, 2017) (<i>iccv,2017</i>)	60.4
ACoL (Zhang <i>et al.</i> , 2018c) (<i>cvpr,2018</i>)	62.3
SPG (Zhang <i>et al.</i> , 2018d) (<i>eccv,2018</i>)	67.9
ADL (Choe & Shim, 2019) (<i>cvpr,2019</i>)	73.5
CutMix (Yun <i>et al.</i> , 2019) (<i>eccv,2019</i>)	54.7
DiPS (ours)	96.2

Table-A I-5 MaxBoxAcc, top-1 and top-5 performance on the CUB dataset

	CUB		
	MaxBoxAcc	top-1 Loc Acc	top-5 Loc Acc
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr,2016</i>)	73.2	56.1	–
HaS (Singh & Lee, 2017) (<i>iccv,2017</i>)	78.1	60.7	–
ACoL (Zhang <i>et al.</i> , 2018c) (<i>cvpr,2018</i>)	72.7	57.8	–
SPG (Zhang <i>et al.</i> , 2018d) (<i>eccv,2018</i>)	63.7	51.5	–
ADL (Choe & Shim, 2019) (<i>cvpr,2019</i>)	75.7	41.1	–
CutMix (Yun <i>et al.</i> , 2019) (<i>eccv,2019</i>)	71.9	54.5	–
ICL (Minsong <i>et al.</i> , 2020) (<i>accv,2020</i>)	57.5	–	–
TS-CAM (Gao <i>et al.</i> , 2021) (<i>iccv,2021</i>)	87.7	71.3	83.8
BR-CAM (Zhu <i>et al.</i> , 2022a) (<i>eccv,2022</i>)	–	–	–
CREAM (Xu <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	90.9	71.7	86.3
BGC (Kim <i>et al.</i> , 2022) (<i>cvpr,2022</i>)	93.1	70.8	88.0
F-CAM (Belharbi <i>et al.</i> , 2022c) (<i>wacv,2022</i>)	92.4	59.3	82.7
DiPS (ours)	97.0	78.8	91.3
DiPS (ours) (w/ TransFG classifier (He, Chen, Liu & et. al, 2022))	97.0	88.2	95.6

6.5 Visual Results

Visual representation of our method compared to baseline methods on CUB is illustrated in Fig.I-7. Our method generates a very smooth map instead of hot-spotting different parts of the concerned object. Ultimately, the map generated by our method does not require an extensive threshold search to find an optimal bounding box. Compared to the `class` tokens of SST (used to harvest pseudo-labels), our method is able to learn an effective localization map from noisy pseudo-labels.

6.6 Details on Baseline Method: TS-CAM

By taking advantage of the attention mechanism, TS-CAM (Gao *et al.*, 2021) is capable of capturing long-range dependency among different parts of an image. As a result, it can efficiently separate background and foreground objects. In other words, it first divides the images into a set of patches for capturing long-range dependency information and records its effects in `class` token. The attention of `class` token is then fused with the semantic aware map to produce the final attention/activation map. The flow diagram of TS-CAM is depicted in Fig.I-6. Lastly, a detailed visualization of the internal representation of the token TS-CAM `class` is presented in Fig.I-8. It shows that the average of all maps could potentially include noise and background regions in the final prediction.

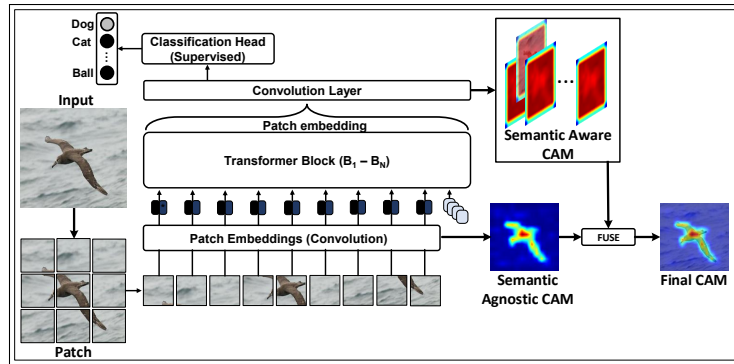


Figure-A I-6 Illustration of the baseline Token Semantic Coupled Attention Map (TS-CAM) model for WSOL

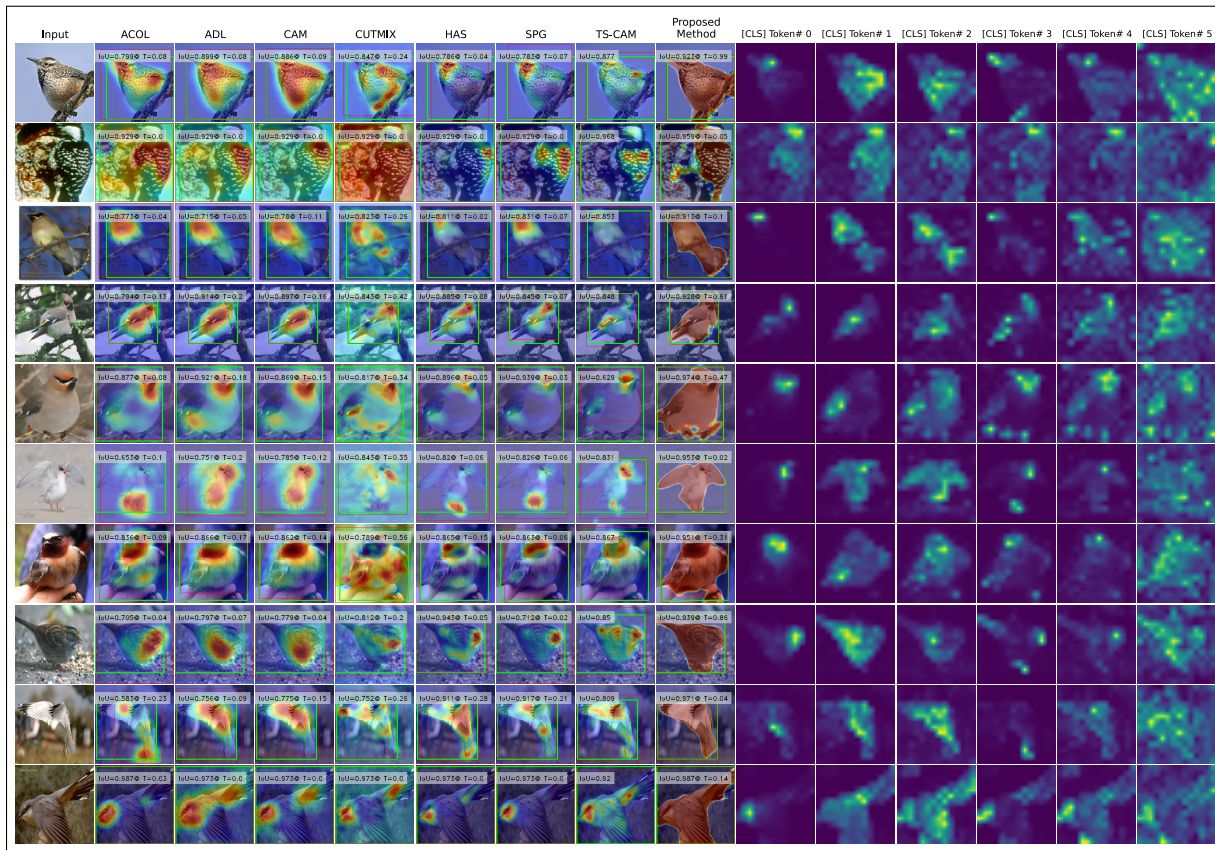


Figure-A I-7 Examples of visual results on the CUB-200-2011 dataset

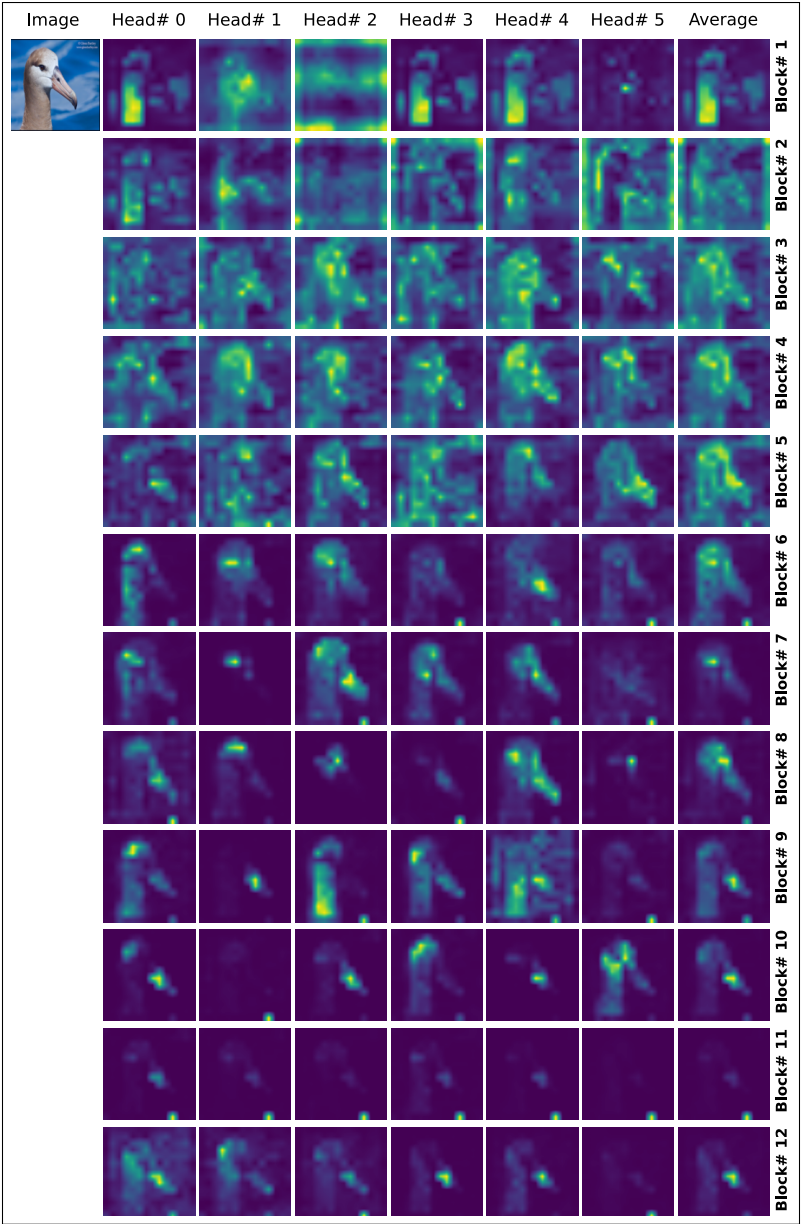


Figure-A I-8 Attention map of each transformer head (class token) learned by TS-CAM. However, different parts of the object are accumulated across the layers/blocks, and it must include semantic aware CAM to suppress noise and generate final results

APPENDIX II

LEVERAGING TRANSFORMERS FOR WEAKLY SUPERVISED OBJECT LOCALIZATION IN UNCONSTRAINED VIDEOS

Shakeeb Murtaza^a , Marco Pedersoli^a , Aydin Sarraf^b , Eric Granger^a

^aDepartment of Systems Engineering, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

^bEricsson, Global AI Accelerator,
8275 Trans Canada Route, Saint-Laurent, Québec H4S 0B6

Paper published in *IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition, Montreal, Canada, October 2024*

Abstract Weakly-Supervised Video Object Localization (WSVOL) involves localizing an object in videos using only video-level labels, also referred to as tags. State-of-the-art WSVOL methods like Temporal CAM (TCAM) rely on class activation mapping (CAM) and typically require a pre-trained CNN classifier. However, their localization accuracy is affected by their tendency to minimize the mutual information between different instances of a class and exploit temporal information during training for downstream tasks, e.g., detection and tracking. In the absence of bounding box annotation, it is challenging to exploit precise information about objects from temporal cues because the model struggles to locate objects over time. To address these issues, a novel method called *transformer based CAM for videos (TrCAM-V)*, is proposed for WSVOL. It consists of a DeiT backbone with two heads for classification and localization. The classification head is trained using standard classification loss (CL), while the localization head is trained using pseudo-labels that are extracted using a pre-trained CLIP model. From these pseudo-labels, the high and low activation values are considered to be foreground and background regions, respectively. Our TrCAM-V method allows training a localization network by sampling pseudo-pixels on the fly from these regions. Additionally, a conditional random field (CRF) loss is employed to align the object boundaries with the foreground map. During inference, the model can process individual frames for real-time localization applications. Extensive experiments on challenging YouTube-Objects unconstrained video datasets show that

our TrCAM-V method achieves new state-of-the-art performance in terms of classification and localization accuracy.

1. Introduction

The rapid development of video-sharing platforms has led to the availability of extensive video data (Shao, Chen, Shao & et. al, 2022; Tang, Sukthankar, Yagnik & Fei-Fei, 2013), escalating the need for automated tools that analyze video content. Video object localization plays a crucial role in understanding this content. It also helps in enhancing the performance of our model in different downstream tasks, such as video object detection (Chen, Cao, Hu & Wang, 2020; Han, Wang, Chang & Qiao, 2020; Shao *et al.*, 2022), visual object tracking (Bergmann, Meinhardt & Leal-Taixé, 2019; Luo *et al.*, 2021), video summarization (Zhang, Han, Jiang, Ye & Chang, 2017) and event detection (Chang, Yang, Long, Zhang & Hauptmann, 2016).

Most videos are captured in wild and unconstrained environments, exhibiting different properties due to different factors such as differences in camera lenses, multiple viewpoints, decoding distortions, moving objects, and editing effects. Leveraging these videos for training our model to perform different downstream tasks requires bounding boxes or pixel-level labels by human annotators. In contrast to images, labeling videos is a very arduous and expensive task, as each video contains a large number of frames. To reduce the annotation cost, videos are weakly labeled (Jerripathula, Cai & Yuan, 2016; Tsai, Zhong & Yang, 2016) by identifying a global tag/class for each video. These global tags represent the object of interest in the video even though some frames may not contain the object of interest, leading to inconsistent labels for different frames. Moreover, these labels don't represent spatial-temporal information about a particular object across different frames. This results in noisy or corrupted frame-level labels, as labels are attributed to an entire video, even though only a subset of its frames might contain the object of interest.

Using these global noisy labels for object localization can reduce the dependency on bounding box annotations, making this task more challenging. Different techniques have been proposed

for weakly supervised video object localization (WSVOL) (Joulin, Tang & Fei-Fei, 2014; Koh, Jang & Kim, 2016; Kwak, Cho, Laptev, Ponce & Schmid, 2015; Prest, Leistner, Civera & et. al, 2012; Rochan, Rahman, Bruce & Wang, 2016; Zhang, Han, Yang & Xu, 2020d) and weakly supervised video object segmentation (Belharbi *et al.*, 2023a; Croitoru, Bogolin & Leordeanu, 2019; Fu, Xu, Zhang & Lin, 2014; Haller & Leordeanu, 2017; Liu, Tao, Song & et. al, 2014; Tsai *et al.*, 2016; Tokmakov, Alahari & Schmid, 2016; Umer, Dawood, Yousaf & et. al, 2021; Yan, Xu, Cai & Corso, 2017; Zhang, Javed & Shah, 2014). Nevertheless, these techniques necessitate post-processing, rendering them infeasible for object localization in unconstrained videos within real-world scenarios.

State-of-the-art WSVOL methods closely adhere to standard protocols (Hartmann, Grundmann, Hoffman & et. al, 2012; Kwak *et al.*, 2015; Prest *et al.*, 2012; Tang *et al.*, 2013; Tokmakov *et al.*, 2016; Xu, Xiong & Corso, 2012; Yan *et al.*, 2017; Zhang *et al.*, 2020d). They first generate object proposals using visual cues (e.g., motion) which are then employed to identify the relevant object using different post-processing techniques. Despite their outstanding performance, these methods exhibit several limitations. Typically, these methods involve multiple steps for WSVOL that restrain us from training these models in an end-to-end fashion, thus rendering these models susceptible to sub-optimal solutions. They also necessitate a separate model for each class as they are trained to localize a particular object. These limitations deem these models ill-suited for deployment in real-world environments and restrict their scalability to a small number of classes due to various constraints (e.g., resources and inference time). Most of these methods cannot employ class-level labels for extracting initial proposals, so they cannot semantically align different object parts belonging to an object of a particular class/tag. Moreover, these methods often utilize motion cues (e.g., optical flow) to localize an object of interest, which makes them susceptible to the same alignment problem as they often ignore the semantics of a particular object. Also, relying solely on motion cues hinders the network performance as they contain noisy information in unconstrained videos due to various external factors such as camera movements.

In a recent study, a discriminative multi-class deep learning model (DL) for WSVOL has been proposed (Belharbi *et al.*, 2023a). This method leverages class activation maps (CAMs) for WSVOL, which have proven effective for weakly supervised object localization (WSOL) in static images (Choe *et al.*, 2020; Belharbi *et al.*, 2022c; Murtaza *et al.*, 2023a,b, 2022; Rony *et al.*, 2023). In WSOL, the model is trained using image-level labels to localize an object of interest corresponding to the underlying class. These CAMs highlight the areas that strongly contribute to predicting a particular class. Nevertheless, these methods are not inherently designed to utilize the temporal information in videos for WSVOL. To deal with this issue, recent studies (Belharbi *et al.*, 2023a) propose an approach to leverage spatio-temporal information by harvesting CAMs using the LayerCAM (Zhou *et al.*, 2016) method. While this approach harnesses the spatiotemporal relationships within videos, it remains susceptible to the accumulation of inaccurate activations arising due to unconstrained object motion. This prevents the model from forming dependencies between object parts, thereby hindering its ability to localize various parts of the object, as depicted in the results section of their study.

To address these issues, we propose a transformer-based class activation mapping for videos (TrCAM-V) to localize a particular object. Unlike other methods, TrCAM-V requires only video-level annotations for training and is independent of additional assumptions, such as motion and temporal cues. This is inspired by WSOL methods, which employ pseudo-labels to train models for WSOL tasks within static images (Murtaza *et al.*, 2023a; Belharbi *et al.*, 2022c; Wei *et al.*, 2021). They utilize pseudo-labels and class-level labels to train the localization and classification head in an end-to-end fashion. Moreover, these approaches scale effectively with a large number of classes, making them applicable to a wide range of applications.

This paper aims to explore the transformer models for WSVOL tasks by leveraging pseudo-label from a Contrastive Language-Image Pretraining (CLIP) model. Our proposed method, called TrCAM-V, consists of a transformer encoder (adhering to the design framework of DeiT (H. Touvron *et al.*, 2021)) with two heads – one head for classification and the other for localization. The localization head produces a map with the same resolution as the input image, which is trained using pseudo-labels. Furthermore, the pseudo-labels for each frame

are extracted from CLIP (Radford *et al.*, 2021) model using GradCAM presented in (Lin *et al.*, 2023). This model accepts an image with a text prompt (class label) to produce a pixel-level pseudo-label (activation map). In line with standard protocols (Durand *et al.*, 2017; Zhou *et al.*, 2016), the strong and weak activations are deemed as FG and BG regions, respectively. Moreover, at each stochastic gradient descent (SGD) step (Murtaza *et al.*, 2023a), foreground (FG) and background (BG) pseudo-pixels are selected to build pseudo-labels (Murtaza *et al.*, 2023a) to train localization head. This random sampling enables the network to explore FG/BG regions and fosters the emergence of activation values over different object parts in the localization map. We also employ conditional random field (CRF) loss to align the object boundaries with the boundaries of the localization map by leveraging statistical properties of an image, such as pixel color and proximity among pixels. Moreover, our model does not require exploiting temporal dependencies during either training or inference. This approach is more suitable for real-time applications compared to other state-of-the-art WSVOL methods, as TCAM does not need to process an entire video to localize an object within a particular frame.

The main contributions of this paper are summarized as follows.

- (1) A novel transformer-based CAM (TrCAM-V) method is proposed for WSVOL tasks. It is comprised of two heads – a classification head trained with class labels, and a localization head trained with pseudo pixels, which are obtained from a pre-trained CLIP model.
- (2) Unlike previous WSVOL methods (Belharbi *et al.*, 2023a), TrCAM-V does not necessitate a pre-trained classifier to harvest activation maps and does not require temporal information during training.
- (3) Following the WSVOL experimental protocol in (Belharbi *et al.*, 2023a), our empirical results shows that our method achieves state-of-the-art performance the challenging YouTube-Object v1.0 (YT0v1) (Prest *et al.*, 2012) and v2.2 (YT0v2.2) (Kalogeiton, Ferrari & Schmid, 2016) datasets.

2. Related Work

(a) Weakly Supervised Object Localization in Images. The baseline method for WSOL in still images is class activation mapping (CAM). This technique generates the localization map by weighted aggregation of different activations map from the penultimate layer of a convolutional neural network (CNN) (Zhou *et al.*, 2016). The weight coefficient for each activation map is determined by employing global average pooling (GAP) on these activations. To improve the localization performance, different pooling layers have been introduced such as WILDCAT (Durand *et al.*, 2017; Durand, Thome & Cord, 2016), Peak Response Maps (Zhou *et al.*, 2018), Log-Sum-Exp (Pinheiro & Collobert, 2015; Sun *et al.*, 2016), and multi-instance learning pooling (Ilse *et al.*, 2018). Despite the success of these methods, they tend to produce blobby maps covering only discriminative areas. This problem arises because these methods seek to minimize the mutual information between different instances of the same class, resulting in models that only highlight discriminate regions of a particular object (Choe *et al.*, 2020). To deal with this issue, different methods have been proposed to expand attention maps beyond discriminative regions (Belharbi *et al.*, 2022b; Choe & Shim, 2019; Li *et al.*, 2018; Mai *et al.*, 2020; Singh & Lee, 2017; Wei *et al.*, 2017; Yun *et al.*, 2019; Zhang *et al.*, 2018c; Zhu, Zhou, Ye *et al.*, 2017). Additionally, different transformer based methods have been proposed to improve the effective receptive field and cover all object parts (Chen *et al.*, 2022; Li, 2022; Gupta *et al.*, 2022; Bai *et al.*, 2022; Su *et al.*, 2022; Meng *et al.*, 2022c; Gao *et al.*, 2021).

Instead of just minimizing the classification loss, different methods proposed to employ pseudo-labels for directly minimizing loss over generated maps to generate robust localization maps. (Murtaza *et al.*, 2022; Belharbi *et al.*, 2022c,a; Meethal, Pedersoli, Belharbi & Granger, 2021; Murtaza *et al.*, 2023a; Wei *et al.*, 2021; Zhang, Cao & Wu, 2020b; Zhang *et al.*, 2018d). For instance, (Murtaza *et al.*, 2022, 2023a) efficiently sample different FG and BG pixels at each stochastic gradient descent step to reduce the effect of cluttered BG. These methods rely solely on a forward pass to compute localization maps. However, different methods have been proposed to compute maps using both forward and backward passes (Cao *et al.*, 2015; Zhang *et al.*, 2018a; Selvaraju *et al.*, 2017; Chattopadhyay *et al.*, 2018; Fu *et al.*, 2020; Jiang *et al.*, 2021;

Adebayo *et al.*, 2018; Kindermans, Hooker, Adebayo & *et. al.*, 2019; Ramaswamy *et al.*, 2020; Wang *et al.*, 2020a,b; Naidu *et al.*, 2020). For instance, (Chattopadhyay *et al.*, 2018; Fu *et al.*, 2020; Jiang *et al.*, 2021) rely on gradient information while (Adebayo *et al.*, 2018; Kindermans *et al.*, 2019; Ramaswamy *et al.*, 2020; Wang *et al.*, 2020a,b; Naidu *et al.*, 2020) employ confident aggregation for avoiding gradient saturation. Similarly, (Chattopadhyay *et al.*, 2018) and (Fu *et al.*, 2020) employ employee feedback layer and Excitation-backprop to improve localization performance of the network.

(b) Weakly Supervised Video Object Segmentation. Several methods have been proposed for video segmentation, each requiring various post-processing steps to produce final activation maps. Most of these methods process single videos or clusters of videos to localize a prominent object without relying on discriminative information.

Baseline methods for WSVOL (Hartmann *et al.*, 2012; Tang *et al.*, 2013; Xu *et al.*, 2012; Yan *et al.*, 2017) begin by extracting spatial-temporal segmentation either using pre-trained detectors (Zhang, Chen, Li & *et. al.*, 2015) or some unsupervised methods (Banica, Agape, Ion & Sminchisescu, 2013; Xu *et al.*, 2012). These methods employ different graph-based techniques such as GrabCut (Rother, Kolmogorov & Blake, 2004) and conditional random fields (CRF) (Krahenbuhl & Koltun, 2011; Tang *et al.*, 2018). The extracted segments are then used to localize an object by imposing motion cues and the visual appearance of a particular object. Similarly, (Liu *et al.*, 2014) proposed a method for multi-class segmentation by employing nearest neighbor-based label transfer methods between videos belonging to the same subclass. Initially, spatial-temporal supervoxels (Xu *et al.*, 2012) are identified that are projected into a high-dimensional space using different cues e.g., color, texture, and motion. Using these supervoxels, a graph is constructed to enforce label smoothness among spatial-temporally contiguous supervoxels in a particular video, as well as among supervoxels exhibiting similar visual appearances across various videos. Additionally, M-CNN (Tokmakov *et al.*, 2016) employs a CNN network to estimate FG regions using motion cues and the Gaussian mixture model. These FG regions are then integrated using a fully convolutional network fine-tuned on a few images.

Several methods have also been proposed for co-segmentation, where a model is trained to identify similar objects presented in different images. A common approach is to utilize inter and intra-video visual and motion cues to discover common segments by modeling the relations between different segments using graph-based techniques (Chen, Chen & Chang, 2012; Fu *et al.*, 2014; Tsai *et al.*, 2016; Zhang *et al.*, 2014). For instance, (Zhang *et al.*, 2014) employed a regulated maximum weight clique to sample object proposal for co-segmentation. Similarly, (Fu, Xu, Zhang, Lin & Rabab, 2015) employs a method for multiple-object segmentation in a scene by using intra-video coherence of different object parts, as well as the consistency of the FG objects across various videos. Moreover, in (Tsai *et al.*, 2016) the authors extracted object tracklets for each video using a pre-trained FCN and linked them to the object category using a graph. Using this graph, relationships between different tracklets were formulated using various cues (e.g., shape, motion, appearance) to discover the prominent objects in different videos.

The methods described above optimize using global class labels for each video. However, different methods have been proposed to learn object segmentation without relying on video level labels (Croitoru *et al.*, 2019; Haller & Leordeanu, 2017; Papazoglou & Ferrari, 2013; Umer *et al.*, 2021). These methods first estimate potential FG regions using different cues (motion, appearance) and then use these regions to produce a segmentation map. For example, (Croitoru *et al.*, 2019) employed a DL model to process initial FG regions as pseudo-labels to train CNN for segmentation.

(c) Weakly Supervised Video Object Localization. Despite the success of deep learning methods for WSOL in still images, limited methods have been proposed for WSVOL (Belharbi *et al.*, 2023a; Belharbi *et al.*, 2023b; Koh *et al.*, 2016; Kwak *et al.*, 2015; Prest *et al.*, 2012; Rochan *et al.*, 2016; Zhang *et al.*, 2020d). Most of these models typically initialize and refine prominent proposals while incorporating spatiotemporal consistency constraints by exploiting information from visual appearance and motion cues of an object. Some methods have employed proposals as supervision to train localizers (Prest *et al.*, 2012; Zhang *et al.*, 2020d), whereas others have leaned towards segmentation-based approaches (Rochan *et al.*, 2016), further refining

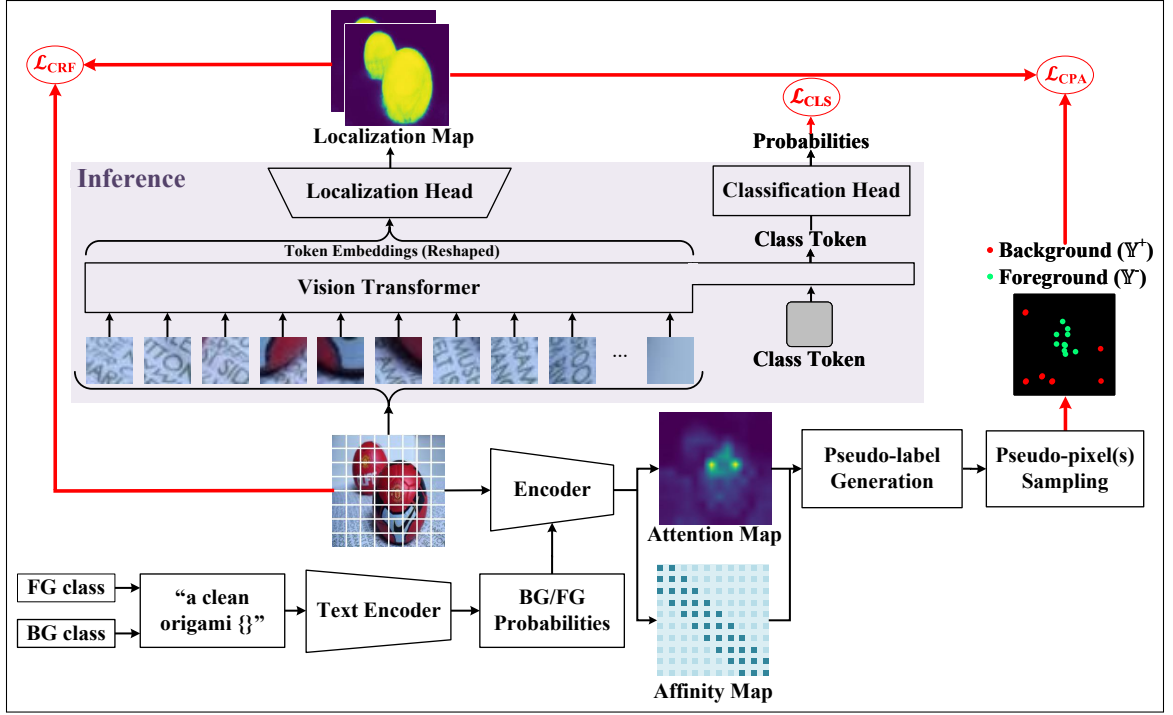


Figure-A II-1 Illustration of the proposed TrCAM-V training architecture. It consists of DeiT backbone a classification and a localization head that are trained using class labels and pseudo labels, respectively. A pre-trained CLIP model is employed to generate pseudo-labels by utilizing a sharpness-based prompt along with the input image, as suggested in (Lin *et al.*, 2023). These pseudo-labels are then used to sample pseudo-pixels for training the localization head. For inference, we only retain DeiT with both heads

the results using GrabCut (Rother *et al.*, 2004). Moreover, these methods often only select one or more videos of a particular class to optimize an underlying model.

Similarly, methods have been proposed to localize a prominent object in a video by identifying similar FG regions based on visual appearance and motion cues Koh *et al.* (2016); Rochan *et al.* (2016); Kwak *et al.* (2015). For instance, (Rochan *et al.*, 2016) generate bounding box proposals within a video and retain relevant ones to build an object appearance detection model by enforcing temporal appearance consistency using maximum a posteriori inference. Similarly, (Kwak *et al.*, 2015) learn to localize an object by discovering similar objects in different videos and tracking prominent regions in individual videos. (Manen, Guillaumin & Gool, 2013) employed region proposals along with appearance and motion consistency to discover FG

objects (Brox & Malik, 2010) thereby maintaining temporal relationships between consecutive frames. In addition to conventional methods, deep learning (DL) models have been proposed for WSVOL. For instance, (Zhang *et al.*, 2020d) proposed SPFTN to jointly learn object segment and localization tasks by employing an optical flow technique (Lee, Kim & Grauman, 2011). Recently (Belharbi *et al.*, 2023a,b) proposed an approach for WSVOL that exploits mutual information between different consecutive frames. However, these methods require temporal information during training and also rely on pre-trained classifiers to harvest pseudo-labels and low-level representations that hinder their localization abilities as they tend to minimize the mutual information between instances of the same class. Compared to these methods, TrCAM-V does not require temporal information during training, yet is capable of achieving state-of-the-art performance because it does not rely on discriminative pseudo-labels, harvested from a pre-trained classifier, for the training of the localization head.

3. Proposed Approach

Notation. Consider a training set $\mathbb{T} = \{v_i, y_i\}$ where each $v_i = \{x_j \in \mathbb{R}^{H \times W \times 3}\}_{j=1}^f$ represent an input video composed of f frames and each $y_i \in \{1, \dots, C\}$ represent the global class label i -th video, where $i = 1, 2, \dots, n$ and C is the number of classes in our dataset. It's further presumed that each frame within a video encompasses a particular object corresponding to the underlying class y_i . Furthermore, our model is capable of accepting a frame f_i and predicting class category y_i along with a localization map $\hat{M}$ encompassing a corresponding object.

3.1 Background on Transformers

Vision Transformer (ViTs) have recently achieved state-of-the art accuracy in different tasks such as classification (Dosovitskiy *et al.*, 2020; H. Touvron *et al.*, 2021) and WSOL (Gao *et al.*, 2021; Murtaza *et al.*, 2023a) tasks. A ViT is composed of N cascaded encoder blocks that capture long-range relationships between different object parts within a scene. Each block consists of different layers, including a multi-headed attention layer, followed by a fully connected layer. To pass an image through the transformer, a frame f_i is divided into M patches of size

$(H/S) \times (W/S)$. Subsequently, positional embeddings and a class (`class`) token are added to the P patches and projected into a fixed embedding dimension of size d . The computed embeddings are then passed through the N cascaded encoder blocks of the transformer. From the output of the last encoder layer, the `class` token is extracted and passed through a fully connected layer to produce classification scores for the relevant classes.

Self-Supervised Transformers (SSTs) can learn object representations within an image without relying on explicit supervision. In this context, self-distillation with no labels (DINO) has been proposed to learn meaningful representation from data without supervision (Caron *et al.*, 2021). To learn a representation in an unsupervised fashion, the author employed a student and a teacher network. The objective is to match their probability distributions corresponding to two different views of an input image. The output of these networks is normalized using temperature-softmax. Moreover, after each optimization step, the parameters of the student are transferred to the teacher network using an exponential moving average.

3.2 Transformer-Based CAM for Videos

The SST (Caron *et al.*, 2021) can identify objects in a frame without explicit supervision. However, they tend to decompose different object parts each corresponding to different attention maps as they are trained without explicit supervision. These attention maps highlight different object parts based on their attention coefficients learned during network training. For extracting relevant object parts and suppressing BG regions from these maps, we propose TrCAM-V. It decodes the patch embeddings and `class` tokens to produce a localization map that is optimized using pseudo-labels derived from the pre-trained CLIP model. Furthermore, these pseudo-labels allow us to directly optimize the loss over generated pseudo-labels instead of just minimizing the mutual information between different instances of a particular class.

More specifically, our model consist of a transformer backbone T with two heads; (i) classification head F trained using global video labels via standard cross-entropy, $\min_{\theta} -\log(\Pr(y|f_i))$. It accept `class` token for an input video frame f_i from transformer backbone and produce

classification probabilities for each class $F(T(f_i)) = \Pr(c|f_i)$ where $F(T(f_i)) \in [0, 1]_c$. **(ii)** A localization head G that accepts patch embeddings and class tokens from the last layer of the transformer to produce a high-resolution localization map M . It consists of two channels M_0 and M_1 , which represent the BG and FG maps, respectively. The localization head is trained using pseudo-labels extracted from a pre-trained CLIP model (PCM). To extract the pseudo-label, TrCAM-V first obtains a saliency map from PC, after which it determines the FG and BG regions using Otsu’s threshold (Otsu, 1979). The densest region among the different regions extracted by the Otsu method is used to sample FG and BG regions. Given that these sampled regions may be noisy and uncertain regarding our concerned object, we refrain from directly using these maps to minimize the loss over the generated map. Instead, we stochastically sample a few FG and BG pixels (Feng, Yang, Laine & Angelini, 2017) and utilize them to train our model using partial cross-entropy. This allows the network to explore relevant object parts, resulting in a more robust localization map. This sampling process can also be viewed as a *fill-in-the-gap* approach. It enables our localization head to explore similar-looking object parts while considering the statistical properties of the object to generate a robust localization map. Moreover, in contrast to baseline methods (Belharbi *et al.*, 2023a,b), we do not take into account the temporal information for producing pseudo-labels.

Initial pseudo-label generation. Following (Lin *et al.*, 2023), we employ GradCAM to harvest maps from CLIP for FG maps while suppressing regions belonging to BG categories. Initially, attention map regions are identified via a text-driven approach, leveraging a sharpness-based prompt as opposed to solely relying on class categories. To extract weights for generating GradCAM, a softmax function is applied to the computed attention between text and image embeddings. To compute the final activation maps, class-aware attention-based affinity (CAA) maps are extracted and fused with initial CAMs, as depicted in Fig.II-1. These harvested maps can be viewed as pseudo-labels that are used to harvest pseudo-pixels.

Selection of FG/BG pixels. To train the localization head G , we leverage pseudo pixels from saliency maps S obtained from pre-trained CLIP model. We produced a separate pseudo-label for each frame f_i , without considering temporal information into account. In each saliency map,

weak and strong activations are likely to represent the BG and FG regions, respectively (Belharbi *et al.*, 2023a, 2022a,c; Durand *et al.*, 2017; Murtaza *et al.*, 2023a; Zhou *et al.*, 2016). Relying on this assumption, we can estimate BG $\mathbb{C}^-$ and FG regions $\mathbb{C}^+$ using Otsu threshold (Otsu, 1979) as follows,

$$\mathbb{M}_t^+ = \mathcal{O}^+(\mathbf{M}), \quad \mathbb{M}_t^- = \mathcal{O}^-(\mathbf{M}) \quad (\text{A II-1})$$

Here, operation $\mathcal{O}^+$ is employed to select top $n\%$ pixels from the most dense connected area in activation map $\mathbf{M}$. It selects pixels that have a magnitude larger than the Otsu threshold. Conversely, the operation $\mathcal{O}^-$ is used to select the bottom $-n\%$ pixels from $\mathbf{M}$, which are ordered from the lowest to highest activation value. These selected regions are uncertain about object location and may contain incorrect labels. Considering the potential uncertainty, a few BG and FG pixels are sampled probabilistically to construct our pseudo-labels, as follows:

$$\mathbf{M}' = \mathcal{P}(\mathbb{M}^+) \cup \mathcal{P}(\mathbb{M}^-), \quad (\text{A II-2})$$

where $\mathcal{P}$ represents the sampling process of a few pixels from selected FG and BG regions. To generate the final pseudo-label, the values of the selected FG and BG pixels are replaced with 1's and 0's, respectively.

Training loss. Cross-entropy is employed to train our classification head. It relies on DeiT features for producing class category $\Pr(c|\mathbf{f}_i)$ as follows:

$$\min_{\theta_c} -\log(\Pr(y|\mathbf{f}_i)) . \quad (\text{A II-3})$$

For the localization head, we employ a loss function consisting of three terms:

(i) The pixel alignment loss ($\mathcal{L}_{PAL}$) for learning FG/BG regions¹. It aims to align the output map $\hat{\mathbf{M}}$ with the selected pixels within $\mathcal{P}(\mathbb{M}^+) \cup \mathcal{P}(\mathbb{M}^-)$ via a partial cross-entropy loss denoted as

¹ A similar approach for exploiting few pixels to localize a concerned object is also employed in (Feng *et al.*, 2017).

$\mathcal{L}_{CPA}(\mathcal{P}(\mathbb{M}^+) \cup \mathcal{P}(\mathbb{M}^-), M^r)$, where r represents the pseudo-pixels:

$$\mathcal{L}_{PAL} = -(1 - \hat{M}) \log(\mathbb{M}^+) - \hat{M}(\log(\mathbb{M}^-)) \quad (\text{A II-4})$$

(ii) The absolute size loss ($\mathcal{L}_{ASL}$) for localizing all object parts belonging to FG/BG regions, as defined by (Belharbi *et al.*, 2022c). It is formulated as a constraint for pushing FG and BG regions away and optimized using the log-barrier method (Boyd & Vandenberghe, 2004).

(iii) The conditional random field (CRF) loss is employed to align the localization map with the boundaries of the object by utilizing colour similarity and proximity of nearby pixels as $\mathcal{L}_{CRF}(\mathbf{S}, \mathbf{f}_i) = \sum_{r \in \{0,1\}} \mathbf{S}^{r\top} \mathbf{W} (\mathbf{1} - \mathbf{S}^r)$. $\mathbf{W}$ is the affinity matrix to capture colour similarity and pixels' proximity between pixels of frame $\mathbf{f}_i$ and $\mathbf{W}$. The overall loss is defined as:

$$\mathcal{L} = \min_{\theta} \lambda_{PAL} \mathcal{L}_{PAL} + \lambda_{ASL} \mathcal{L}_{ASL} + \lambda_{CRF} \mathcal{L}_{CRF}, \quad (\text{A II-5})$$

where λ_{CLS} and λ_{CPA} are hyperparameters that lie in the interval $[0, 1]$. $2e^{-9}$ and λ_{CRF} is set to $2e^{-9}$ (Tang *et al.*, 2018).

Inference. During the inference phase, we discard CLIP and its associated components utilized during training to harvest pseudo-labels. We only retain DeiT backbone T along with the classification and inference head. The backbone network T inputs a frame $\mathbf{f}_i$ and predicts a class label y_i and the corresponding localization map $\hat{M}$.

4. Results and Discussion

4.1 Experimental Methodology

Datasets. To validate the performance of our model, we conducted extensive experiments on two challenging datasets comprised of unconstrained videos from YouTube²; YouTube-Object v1.0 (YT0v1) (Prest *et al.*, 2012) and YouTube-Object v2.2 (YT0v2.2) (Kalogeiton *et al.*, 2016).

² <https://www.youtube.com/>

Table-A II-1 CorLoc performance on YT0v1 Prest *et al.* (2012) and YT0v2.2 Kalogeiton *et al.* (2016) test sets

Dataset	Method (venue)	Aero	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Avg
YT0v1	Prest <i>et al.</i> (2012) (<i>cvpr</i> ,2012)	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
	Papazoglou & Ferrari (2013) (<i>iccv</i> ,2013)	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1
	Joulin <i>et al.</i> (2014) (<i>eccv</i> ,2014)	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
	Kwak <i>et al.</i> (2015) (<i>iccv</i> ,2015)	56.5	66.4	58.0	76.8	39.9	69.3	50.4	56.3	53.0	31.0	55.7
	Rochan <i>et al.</i> (2016) (<i>ivc</i> ,2016)	60.8	54.6	34.7	57.4	19.2	42.1	35.8	30.4	11.7	11.4	35.8
	Tokmakov <i>et al.</i> (2016) (<i>eccv</i> ,2016)	71.5	74.0	44.8	72.3	52.0	46.4	71.9	54.6	45.9	32.1	56.6
	POD Koh <i>et al.</i> (2016) (<i>cvpr</i> ,2016)	64.3	63.2	73.3	68.9	44.4	62.5	71.4	52.3	78.6	23.1	60.2
	Tsai <i>et al.</i> (2016) (<i>eccv</i> ,2016)	66.1	59.8	63.1	72.5	54.0	64.9	66.2	50.6	39.3	42.5	57.9
	Haller & Leordeanu (2017) (<i>iccv</i> ,2017)	76.3	71.4	65.0	58.9	68.0	55.9	70.6	33.3	69.7	42.4	61.1
	Croitoru <i>et al.</i> (2019) (LowRes-Net _{tier1}) (<i>ijcv</i> ,2019)	77.0	67.5	77.2	68.4	54.5	68.3	72.0	56.7	44.1	34.9	62.1
	Croitoru <i>et al.</i> (2019) (LowRes-Net _{tier2}) (<i>ijcv</i> ,2019)	79.7	67.5	68.3	69.6	59.4	75.0	78.7	48.3	48.5	39.5	63.5
	Croitoru <i>et al.</i> (2019) (DilateU-Net _{tier2}) (<i>ijcv</i> ,2019)	85.1	72.7	76.2	68.4	59.4	76.7	77.3	46.7	48.5	46.5	65.8
	Croitoru <i>et al.</i> (2019) (MultiSelect-Net _{tier2}) (<i>ijcv</i> ,2019)	84.7	72.7	78.2	69.6	60.4	80.0	78.7	51.7	50.0	46.5	67.3
	SPFTN (M) Zhang <i>et al.</i> (2020d) (<i>tpami</i> ,2020)	66.4	73.8	63.3	83.4	54.5	58.9	61.3	45.4	55.5	30.1	59.3
	SPFTN (P) Zhang <i>et al.</i> (2020d) (<i>tpami</i> ,2020)	97.3	27.8	81.1	65.1	56.6	72.5	59.5	81.8	79.4	22.1	64.3
	FPPVOS Umer <i>et al.</i> (2021) (<i>optik</i> ,2021)	77.0	72.3	64.7	67.4	79.2	58.3	74.7	45.2	80.4	42.6	65.8
	CAM Zhou <i>et al.</i> (2016) (<i>cvpr</i> ,2016)	75.0	55.5	43.2	69.7	33.3	52.4	32.4	74.2	14.8	50.0	50.1
	GradCAM Selvaraju <i>et al.</i> (2017) (<i>iccv</i> ,2017)	86.9	63.0	51.3	81.8	45.4	62.0	37.8	67.7	18.5	50.0	56.4
	GradCAM++ Chattopadhyay <i>et al.</i> (2018) (<i>wacv</i> ,2018)	79.8	85.1	37.8	81.8	75.7	52.4	64.9	64.5	33.3	56.2	63.2
	Smooth-GradCAM++ Omeiza, Speakman, Cintas & Weldermariam (2019) (<i>corr</i> ,2019)	78.6	59.2	56.7	60.6	42.4	61.9	56.7	64.5	40.7	50.0	57.1
	XGradCAM Fu <i>et al.</i> (2020) (<i>bmvc</i> ,2020)	79.8	70.4	54.0	87.8	33.3	52.4	37.8	64.5	37.0	50.0	56.7
	LayerCAM Jiang <i>et al.</i> (2021) (<i>ieee</i> ,2021)	85.7	88.9	45.9	78.8	75.5	61.9	64.9	64.5	33.3	56.2	65.6
	TCAM Belharbi <i>et al.</i> (2023a) (<i>wacv</i> ,2023)	90.5	70.4	62.2	75.7	84.8	81.0	81.0	64.5	70.4	50.0	73.0
	TrCAM-V with CLIP (ours)	91.7	77.8	91.9	94.0	84.8	81.0	83.8	77.4	77.8	87.5	84.8
	TrCAM-V with CLIPSeg (ours)	94.0	74.1	94.6	90.9	87.9	81.0	89.2	77.4	74.1	75.0	83.8
YT0v2.2	Haller & Leordeanu (2017) (<i>iccv</i> ,2017)	76.3	68.5	54.5	50.4	59.8	42.4	53.5	30.0	53.5	60.7	54.9
	Croitoru <i>et al.</i> (2019) (LowRes-Net _{tier1}) (<i>ijcv</i> ,2019)	75.7	56.0	52.7	57.3	46.9	57.0	48.9	44.0	27.2	56.2	52.2
	Croitoru <i>et al.</i> (2019) (LowRes-Net _{tier2}) (<i>ijcv</i> ,2019)	78.1	51.8	49.0	60.5	44.8	62.3	52.9	48.9	30.6	54.6	53.4
	Croitoru <i>et al.</i> (2019) (DilateU-Net _{tier2}) (<i>ijcv</i> ,2019)	74.9	50.7	50.7	60.9	45.7	60.1	54.4	42.9	30.6	57.8	52.9
	Croitoru <i>et al.</i> (2019) (BasicU-Net _{tier2}) (<i>ijcv</i> ,2019)	82.2	51.8	51.5	62.0	50.9	64.8	55.5	45.7	35.3	55.9	55.6
	Croitoru <i>et al.</i> (2019) (MultiSelect-Net _{tier2}) (<i>ijcv</i> ,2019)	81.7	51.5	54.1	62.5	49.7	68.8	55.9	50.4	33.3	57.0	56.5
	CAM Zhou <i>et al.</i> (2016) (<i>cvpr</i> ,2016)	52.3	66.4	25.0	66.4	39.7	87.8	34.7	53.6	45.4	43.7	51.5
	GradCAM Selvaraju <i>et al.</i> (2017) (<i>iccv</i> ,2017)	44.1	68.4	50.0	61.1	51.8	79.3	56.0	47.0	44.8	42.4	54.5
	GradCAM++ Chattopadhyay <i>et al.</i> (2018) (<i>wacv</i> ,2018)	74.7	78.1	38.2	69.7	56.7	84.3	61.6	61.9	43.0	44.3	61.2
	Smooth-GradCAM++ Omeiza <i>et al.</i> (2019) (<i>corr</i> ,2019)	74.1	83.2	38.2	64.2	49.6	82.1	57.3	52.0	51.1	42.4	59.5
	XGradCAM Fu <i>et al.</i> (2020) (<i>bmvc</i> ,2020)	68.2	44.5	45.8	64.0	46.8	86.4	44.0	57.0	44.9	45.0	54.6
	LayerCAM Jiang <i>et al.</i> (2021) (<i>ieee</i> ,2021)	80.0	84.5	47.2	73.5	55.3	83.6	71.3	60.8	55.7	48.1	66.0
	TCAM Belharbi <i>et al.</i> (2023a) (<i>wacv</i> ,2023)	79.4	94.9	75.7	61.7	68.8	87.1	75.0	62.4	72.1	45.0	72.2
	TrCAM-V with CLIP (ours)	87.6	91.6	90.3	74.1	78.7	79.2	76.2	66.9	60.0	62.0	76.7
	TrCAM-V with CLIPSeg (ours)	84.7	95.5	92.4	79.3	78.7	87.9	84.1	66.9	68.4	62.0	80.0

For training, only global video labels are available, which correspond to the prominent object in different frames of the video. Further details of these datasets are given below:

YT0v1 consists of various videos collected from YouTube. Ten categories were considered to query different videos containing a particular object. Each class consists of a varying number of videos ranging from 9-24 frames with durations spanning from 0.5-3 minutes. The dataset comprises 155 videos that are segmented into short clips, referred to as shorts. After dividing the videos into clips, we ended up with 5507 shorts encompassing 571,089 frames. Within each short, a few frames have bounding box annotations corresponding to the underlying class. The annotated frames can be used as validation and test sets for hyper-parameter and model selection, respectively. Moreover, the authors reserved 128 videos for training and 27 for testing, which collectively encompass a total of 396 labeled bounding boxes. We reserved some frames from

the training videos for a validation set. To build the validation set, we followed (Belharbi *et al.*, 2022c) and randomly chose five videos per class, resulting in a total of 50 videos.

YT0v2.2 is an extension of YT0v1 and consist of 722,040 frames. Compared to YT0v1, this dataset has a large number of bounding box annotations for different categories. The dataset comprises 155 videos divided into 9 categories. In terms of dataset partitioning, the authors reserved 106 videos for training and 49 videos for testing. Following (Belharbi *et al.*, 2023a), we built our validation set by randomly selecting three videos per class from the training set. Compared to YT0v1, this dataset has a larger number of bounding box annotations; the test set comprises 1,781 frames with bounding box annotations, resulting in a total of 2,667 bounding boxes. The expanded test set and annotations render YT0v2.2 more challenging.

Evaluation measures. We employ two metrics for measuring the performance of our model; (i) CorLoc (Deselaers *et al.*, 2012) is used to measure the localization performance. CorLoc indicates the proportion of predicted bounding having an intersection over union (IoU) greater than 50%. (ii) Standard classification accuracy (CL) is measured to evaluate the classification performance of our model. CL is measured over frames that have bounding boxes annotations.

Implementation details. We follows the protocols of (Belharbi *et al.*, 2023a) for all of our experiments. Specifically, we train our model for 100 epochs with a mini-batch size of 32. All images are resized to 256×256 , then randomly cropped to a size of 224×224 . After cropping, we augment the image by random horizontal flipping, followed by normalization. The weight λ for CRF is set to $2e^{-9}$ as defined in (Tang *et al.*, 2018). For log-barrier optimization, we use the same hyperparameter value as suggested in (Belharbi, Ayed, McCaffrey & Granger, 2019a; Kervadec *et al.*, 2019b; Belharbi *et al.*, 2023a). We optimize our network using the Stochastic Gradient Descent (SGD) algorithm with learning rates ranging between 0.1 and 0.00001 (Murtaza *et al.*, 2023a). Moreover, we use the DeiT backbone with pre-trained weights from SST (Caron *et al.*, 2021). To account for the large number of redundant frames per video, we randomly selected a few frames for each shot for each gradient descent step. This approach enabled us to train our model over a large number of video frames within a reasonable timeframe.

Baseline Models. To validate our TrCAM-V method, its performance is compared with various state-of-the-art methods, e.g., FPPVOS (Umer *et al.*, 2021), SPFTN (Zhang *et al.*, 2020d), POD (Koh *et al.*, 2016), and (Croitoru *et al.*, 2019; Haller & Leordeanu, 2017; Joulin *et al.*, 2014; Kwak *et al.*, 2015; Papazoglou & Ferrari, 2013; Prest *et al.*, 2012; Rochan *et al.*, 2016; Tokmakov *et al.*, 2016; Tsai *et al.*, 2016). The performance of our model is also compared with different CAM methods: LayerCAM (Jiang *et al.*, 2021), XGradCAM (Fu *et al.*, 2020), GradCAM++ (Omeiza *et al.*, 2019), GradCAM (Selvaraju *et al.*, 2017) and CAM (Zhou *et al.*, 2016). Note that our method for generating pseudo-label and pseudo-pixel selection is generic, and can be integrated to train any model for WSOL.

4.2 Comparison with State-of-Art Methods

Qualitative Results. The classification and localization performance of our model is presented in Table II-2 and II-1, respectively. Regarding classification performance, our model exhibits an improvement of 4.9% and 12.7% on the YT0v1 and YT0v2.2 datasets, respectively, when compared with the baseline model. Similarly, our model is able to surpass baseline models in terms of localization performance on both datasets. In contrast to the baseline methods, our method is able to achieve state-of-the-art performance without the need for exploiting temporal information, thereby enabling a consistent pipeline at both training and inference time. In addition to this, we trained our model using pseudo-label harvested from CLIPSeg (Luddecke & Ecker, 2022) that helps in improving the performance of our model on YT0v2.2 dataset as shown in Table II-1.

Table-A II-2 Classification accuracy (CL) on YT0v1 and YT0v2.2 test sets

Methods	YT0v1	YT0v2.2
CAM (Zhou <i>et al.</i> , 2016) (<i>cvpr,2016</i>)	85.3	73.9
GradCAM (Selvaraju <i>et al.</i> , 2017) (<i>iccv,2017</i>)	85.3	71.3
GradCAM++ (Chattopadhyay <i>et al.</i> , 2018) (<i>wacv,2018</i>)	84.4	72.4
Smooth-GradCAM++ (Omeiza <i>et al.</i> , 2019) (<i>corr,2019</i>)	82.6	75.2
XGradCAM (Fu <i>et al.</i> , 2020) (<i>bmvc,2020</i>)	87.3	71.6
TCAM (Belharbi <i>et al.</i> , 2023a) (<i>wacv,2023</i>)	84.4	72.1
TrCAM-V (ours)	92.2	87.9

Quantitative Results. Fig.II-2 depicts localization predictions of TrCAM-V compared to baselines and state-of-the-art methods. Results illustrate that our model can generate robust maps that encompass both FG and BG regions, demarcated by sharp boundaries. Activation-based methods focus on discriminative areas that are common among different instances of the same class. Moreover, TCAM generates localization maps with sharp boundaries over various object parts, facilitating the prediction of bounding boxes that fully enclose the object. However, activation maps fail to encompass all parts of an object, resulting in inaccurate localization in produced maps.

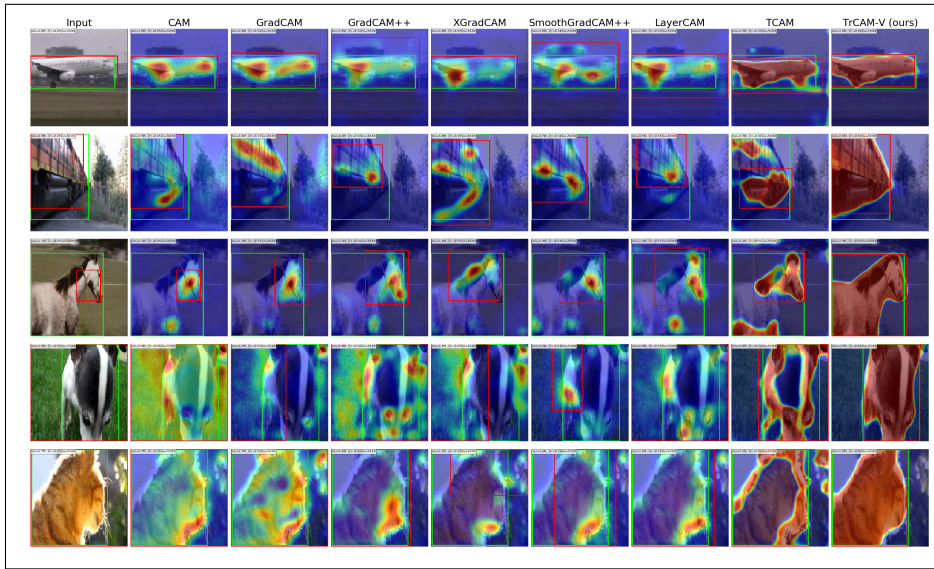


Figure-A II-2 Visualization of YTOv1 frames. Here, red and green box indicate the predicted and annotated bounding-box

4.3 Ablation Studies

Ablations presented in Table II-3 are conducted to show the effectiveness of various loss terms. This study indicates that all auxiliary terms of our loss contribute significantly to the model's performance on both datasets, YTOv1 and YTOv2 . 2. Harvesting pseudo-labels from the CLIP model yields competitive performance compared to the baseline method. Adding size terms to the loss function helps in connecting different object parts. Additionally, CRF terms help to significantly improve localization performance by aligning the boundaries of the localization

map with the boundaries of the concerned object. In contrast to the baseline methods, TrCAM-V can achieve state-of-the-art performance without exploiting temporal information.

Table-A II-3 CorLoc accuracy of TrCAM-V with various losses

Methods	CorLoc	
	YT0v1	YT0v2.2
TCAM Belharbi <i>et al.</i> (2023a) (<i>wacv,2023</i>) (Jiang <i>et al.</i> , 2021)	73.0	72.2
PAL	74.8	72.3
PAL + ASL	75.5	74.2
PAL + CRF	81.4	76.2
PAL + CRF + ASL	84.8	76.7
Improvement	11.8	4.5

4.4 Conclusion

In this paper, a novel approach is proposed to train transformers for WSVOL tasks. The transformer consists of two heads for classification and localization, which are trained using video-level class labels and pixel-wise pseudo-labels, respectively. These pseudo-labels are harvested from the standard CLIP model. Additionally, a CRF loss is employed to align the boundaries of the localization map and the concerned object. Our model outperforms state-of-the-art methods in terms of both classification and localization accuracy without requiring temporal information across subsequent frames.

APPENDIX III

PUBLICATIONS DURING PH.D. STUDY

Journal Articles

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Aydin Sarraf, Eric Granger. “DiPS: Discriminative pseudo-label sampling with self-supervised transformers for weakly supervised object localization”, *Image and Vision Computing*, Volume 140, 2023.
- Soufiane Belharbi, **Shakeeb Murtaza**, Marco Pedersoli, Ismail Ben Ayed, Luke McCaffrey, Eric Granger. “CoLo-CAM: Class activation mapping for object co-localization in weakly-labeled unconstrained videos”, *Pattern Recognition*, 2025.

Conference Articles

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Aydin Sarraf, Eric Granger, “Discriminative Sampling of Proposals in Self-Supervised Transformers for Weakly Supervised Object Localization”, *IEEE/CVF Conference on Applications of Computer Vision (WACV) Workshops*, pp. 155-165, 2022.
- **Shakeeb Murtaza**, Marco Pedersoli, Aydin Sarraf, Eric Granger, “Leveraging Transformers for Weakly Supervised Object Localization” in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)*, 2024, pp. 195-207.
- Alexis Guichemerre, Soufiane Belharbi, Tsiry Mayet, **Shakeeb Murtaza**, Pourya Shamsolmoali, Luke McCaffrey, Eric Granger, “Source-Free Domain Adaptation of Weakly-Supervised Object Localization Models for Histology”, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 33-43, 2024.
- Soufiane Belharbi, Mara KM Whitford, Phuong Hoang, **Shakeeb Murtaza**, Luke McCaffrey, Eric Granger, “SR-CACO-2: A Dataset for Confocal Fluorescence Microscopy Image Super-Resolution”, *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Eric Granger, “A Realistic Protocol for Evaluation of Weakly Supervised Object Localization”, *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

- **Shakeeb Murtaza**, Soufiane Belharbi, Marco Pedersoli, Eric Granger, “TeD-Loc: Text Distillation for Weakly Supervised Object Localization”, (Submitted to IEEE/CVF International Conference on Computer Vision (ICCV), 2025).

BIBLIOGRAPHY

- Abnar, S. & Zuidema, W. (2020). Quantifying attention flow in transformers. *ACL*.
- Adams, A., Baek, J. & Davis, M. (2010). Fast high-dimensional filtering using the permutohedral lattice. *Computer graphics forum*.
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M. & Kim, B. (2018). Sanity checks for saliency maps. *NeurIPS*.
- Agrawal, P., Girshick, R. & Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. *ECCV*.
- Alexe, B., Deselaers, T. & Ferrari, V. (2012). Measuring the Objectness of Image Windows. *TPAMI*.
- Allred, C. (2022, Sep). 5 major benefits of drone cell tower inspections.
- Anderson, P., He, X., Buehler, C. & et. al. (2018). Bottom-up and top-down attention for image captioning and visual question answering. *CVPR*.
- Andreopoulos, A. & Tsotsos, J. K. (2013). 50 years of object recognition: Directions forward. *Computer vision and image understanding*.
- Arefin, M., Michalski, V., St-Charles, P. & et. al. (2020). Multi-image super-resolution for remote sensing using deep recurrent networks. *CVPRw*.
- Ba, J., Mnih, V. & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *CORR*, abs/arXiv:1412.7755.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*.
- Bae, W., Noh, J. & Kim, G. (2020). Rethinking class activation mapping for weakly supervised object localization. *ECCV*.
- Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Bai, H., Zhang, R., Wang, J. & Wan, X. (2022). Weakly Supervised Object Localization via Transformer with Implicit Spatial Calibration. *ECCV*.
- Banica, D., Agape, A., Ion, A. & Sminchisescu, C. (2013). Video Object Segmentation by Salient Segment Chain Composition. *ICCVw*.
- Bau, D., Zhou, B., Khosla, A. & et. al. (2017). Network dissection: Quantifying interpretability of deep visual representations. *CVPR*.

- Bau, D., Zhu, J., Strobel, H. & et. al. (2018). GAN dissection: Visualizing and understanding generative adversarial networks. *CoRR*, abs/arXiv:1811.10597.
- Beal, J., Kim, E., Tzeng, E. & et. al. (2020). Toward transformer-based object detection. *CoRR*, abs/arXiv:2012.09958.
- Bearman, A. L., Russakovsky, O., Ferrari, V. & Fei-Fei, L. (2016). What's the Point: Semantic Segmentation with Point Supervision. *ECCV*.
- Behl, A., Jafari, O. H., Mustikovela, S. K. & et. al. (2017). Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? *ICCV*.
- Bejnordi, B., Timofeeva, N., Otte-Holler, I. & et. al. (2014). Quantitative analysis of stain variability in histology slides and an algorithm for standardization. *MI'14:DP*.
- Bejnordi, B., Litjens, G., Timofeeva, N. & et. al. (2015). Stain specific standardization of whole-slide histopathological images. *TMI*.
- Belharbi, S., Ayed, I., McCaffrey, L. & Granger, E. (2019a). Deep Ordinal Classification with Inequality Constraints. *CoRR*, abs/1911.10720.
- Belharbi, S., Rony, J., Dolz, J., Ayed, I., McCaffrey, L. & Granger, E. (2019b). Min-max entropy for weakly supervised pointwise localization. *CoRR*.
- Belharbi, S., Pedersoli, M., Ayed, I., McCaffrey, L. & Granger, E. (2022a). Negative evidence matters in interpretable histology image classification. *MIDL*.
- Belharbi, S., Rony, J., Dolz, J., Ayed, I., McCaffrey, L. & Granger, E. (2022b). Deep Interpretable Classification and Weakly-Supervised Segmentation of Histology Images via Max-Min Uncertainty. *TMI*.
- Belharbi, S., Sarraf, A., Pedersoli, M., Ayed, I. B., McCaffrey, L. & Granger, E. (2022c). F-CAM: Full Resolution Class Activation Maps via Guided Parametric Upscaling. *WACV*.
- Belharbi, S., Ayed, I., McCaffrey, L. & Granger, E. (2023a). TCAM: Temporal Class Activation Maps for Object Localization in Weakly-Labeled Unconstrained Videos. *WACV*.
- Belharbi, S., Murtaza, S., Pedersoli, M., Ayed, I. B., McCaffrey, L. & Granger, E. (2023b). CoLo-CAM: Class Activation Mapping for Object Co-Localization in Weakly-Labeled Unconstrained Videos. *CoRR*, abs/2303.09044.
- Belharbi, S., Murtaza, S., Pedersoli, M. & et. al. (2023c). CoLo-CAM: Class Activation Mapping for Object Co-Localization in Weakly-Labeled Unconstrained Videos. *CoRR*, abs/2303.09044.
- Benenson, R., Popov, S. & Ferrari, V. (2019). Large-scale interactive object segmentation with human annotators. *CVPR*.

- Bergmann, P., Meinhardt, T. & Leal-Taixé, L. (2019). Tracking Without Bells and Whistles. *ICCV*.
- Bonet, D., Ortega, A., Hidalgo, J. R. & Shekkizhar, S. (2021). Channel-Wise Early Stopping without a Validation Set via NNK Polytope Interpolation. *APSIPA*.
- Borji, A. (2019). Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*.
- Boyd, S. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Breiman, L. (2001). Random Forests. *ML*.
- Brox, T. & Malik, J. (2010). Object Segmentation by Long Term Analysis of Point Trajectories. *ECCV*.
- Cao, C., Liu, X., Yang, Y. & et. al. (2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *ICCV*.
- Cao, X., Zheng, X., Shen, Y. & et. al. (2023). LocLoc: low-level cues and local-area guides for weakly supervised object localization. *ACM MM*.
- Carbonneau, M., Cheplygina, V., Granger, E. & Gagnon, G. (2018). Multiple instance learning: A survey of problem characteristics and applications. *PR*.
- Carion, N., Massa, F., Synnaeve, G. & et. al. (2020). End-to-end object detection with transformers. *ECCV*.
- Caron, M., Misra, I., Mairal, J. & et. al. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *CORR*, abs/arXiv:2006.09882.
- Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P. & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *ICCV*.
- Chang, X., Yang, Y., Long, G., Zhang, C. & Hauptmann, A. G. (2016). Dynamic Concept Composition for Zero-Example Event Detection. *AAAI*.
- Chattopadhyay, A., Sarkar, A., Howlader, P. & Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *WACV*.
- Chefer, H., Gur, S. & Wolf, L. (2021). Transformer interpretability beyond attention visualization. *CVPR*.
- Chen, C., Li, O., Tao, C., Barnett, A., Su, J. & Rudin, C. (2018a). This looks like that: deep learning for interpretable image recognition. *abs/arXiv:1806.10574*.
- Chen, C., Li, O., Tao, D., Rudin, C. & Su, J. (2019). This looks like that: deep learning for interpretable image recognition. *NeurIPS*.

- Chen, D., Pan, X., Tang, F. & et. al. (2023a). Spa2net: Structure-preserved attention activated network for weakly supervised object localization. *TIP*.
- Chen, D., Chen, H. & Chang, L. (2012). Video object cosegmentation. *MC*.
- Chen, H., Gu, J. & Zhang, Z. (2021a). Attention in Attention Network for Image Super-Resolution. *CoRR*, abs/arXiv:2104.09497.
- Chen, L. C., Papandreou, G., Kokkinos, I. & et. al. (2018b). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI*, 40(4), 834–848.
- Chen, W. & Cummings, M. L. (2023). Subjectivity in Unsupervised Machine Learning Model Selection. *CoRR*, abs/2309.00201.
- Chen, X., Hsieh, C. & Gong, B. (2021b). When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations. *CoRR*, abs/arXiv:2106.01548.
- Chen, X., Xie, S. & He, K. (2021c). An Empirical Study of Training Self-Supervised Vision Transformers. *ICCV*.
- Chen, Y., Cao, Y., Hu, H. & Wang, L. (2020). Memory Enhanced Global-Local Aggregation for Video Object Detection. *CVPR*.
- Chen, Z., Wang, C., Wang, Y. & et. al. (2022). LCTR: On Awakening the Local Continuity of Transformer for Weakly Supervised Object Localization. *AAAI*.
- Chen, Z., Ding, J., Cao, L. & et. al. (2023b). Category-aware Allocation Transformer for Weakly Supervised Object Localization. *ICCV*.
- Cheng, Y., Wang, D., Zhou, P. & Zhang, T. (2018). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *SPM*.
- Cheplygina, V., de Bruijne, M. & Pluim, J. (2019). Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning. *MIA*.
- Choe, J. & Shim, H. (2019). Attention-based dropout layer for weakly supervised object localization. *CVPR*.
- Choe, J., Oh, S., Lee, S. & et. al. (2020). Evaluating weakly supervised object localization methods right. *CVPR*.
- Choe, J., Han, D., Yun, S. & et. al. (2021a). Region-based dropout with attention prior for weakly supervised object localization. *PR*.
- Choe, J., Lee, S. & Shim, H. (2021b). Attention-Based Dropout Layer for Weakly Supervised Single Object Localization and Semantic Segmentation. *TPAMI*.

- Choudhary, T., Mishra, V., Goswami, A. & Sarangapani, J. (2020). A comprehensive survey on model compression and acceleration. *AIR*.
- Cinbis, R. G., Verbeek, J. & Schmid, C. (2014). Multi-fold mil training for weakly supervised object localization. *CVPR*.
- Cortes, C. (1995). Support-Vector Networks. *ML*.
- Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*.
- Croitoru, I., Bogolin, S. & Leordeanu, M. (2019). Unsupervised learning of foreground object segmentation. *IJCV*.
- Deng, J., Dong, W., Socher, R. & et. al. (2009). ImageNet: A Large-Scale Hierarchical Image Database.
- Deselaers, T., Alexe, B. & Ferrari, V. (2010). Localizing objects while learning their appearance. *ECCV*.
- Deselaers, T., Alexe, B. & Ferrari, V. (2012). Weakly supervised localization and learning with generic knowledge. *IJCV*.
- Deudon, M., Kalaitzis, A., Goytom, I., Arefin, M. R., Lin, Z., Sankaran, K., Michalski, V., Kahou, S. E., Cornebise, J. & Bengio, Y. (2020). Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *CoRR*, abs/arXiv:2002.06460.
- Dinu, M., Holzleitner, M., Beck, M. & et. al. (2023). Addressing Parameter Choice Issues in Unsupervised Domain Adaptation by Aggregation. *ICLR*.
- Dolz, J., Desrosiers, C. & Ayed, I. (2018). 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*.
- Dong, C., Loy, C., He, K. & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. *ECCV*.
- Dong, C., Loy, C. & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. *ECCV*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A. & et. al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/arXiv:2010.11929.
- Durand, T., Thome, N. & Cord, M. (2016). Weldon: Weakly supervised learning of deep convolutional neural networks. *CVPR*.
- Durand, T., Mordan, T., Thome, N. & Cord, M. (2017). WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. *CVPR*.

- Duvenaud, D., Maclaurin, D. & Adams, R. (2016). Early stopping as nonparametric variational inference. *AIS*.
- Endres, I. & Hoiem, D. (2010). Category Independent Object Proposals. *ECCV*.
- Ericsson, L., Li, D. & Hospedales, T. (2023). Better Practices for Domain Adaptation. *AutoML*.
- Fan, H., Xiong, B., Mangalam, K. & et. al. (2021). Multiscale vision transformers. *CoRR*, abs/arXiv:2104.11227.
- Fang, Y., Wang, W., Xie, B. & et. al. (2023a). EVA: Exploring the limits of masked visual representation learning at scale. *CVPR*.
- Fang, Y., Yap, P., Lin, W., Zhu, H. & Liu, M. (2023b). Source-Free Unsupervised Domain Adaptation: A Survey. *CoRR*, abs/2301.00265.
- Felzenszwalb, P., Girshick, R., McAllester, D. & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *TPAMI*, 32(9), 1627–1645.
- Feng, X., Yang, J., Laine, A. F. & Angelini, E. D. (2017). Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules. *MICCAI*.
- Finnoff, W., Hergert, F. & Zimmermann, H. (1993). Improving model selection by nonconvergent methods. *Neural Networks*.
- Fong, R., Patrick, M. & Vedaldi, A. (2019). Understanding Deep Networks via Extremal Perturbations and Smooth Masks. *ICCV*.
- Fong, R. & Vedaldi, A. (2018). Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. *CVPR*.
- Frank, M. & Wolfe, P. (1956). An algorithm for quadratic programming. *NRLQ*.
- Freund, Y. & Schapire, R. (1996). Experiments with a New Boosting Algorithm. *ICML*.
- Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *AS*.
- Fu, H., Xu, D., Zhang, B. & Lin, S. (2014). Object-Based Multiple Foreground Video Co-segmentation. *CVPR*.
- Fu, H., Xu, D., Zhang, B., Lin, S. & Rabab, R. (2015). Object-based multiple foreground video co-segmentation via multi-state selection graph. *TIP*.
- Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y. & Li, B. (2020). Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312*.
- Fulkerson, B., Vedaldi, A. & Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. *ICCV*.

- GANDER, W. (1980). Algorithms for the QR-Decomposition.
- Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., Zhou, B. & Ye, Q. (2021). TS-CAM: Token Semantic Coupled Attention Map for Weakly Supervised Object Localization. *arXiv preprint arXiv:2103.14862*.
- Ghiasi, A., Shafahi, A. & Ardekani, R. (2022). Adaptive Weight Decay: On The Fly Weight Decay Tuning for Improving Robustness. *CoRR*, abs/2210.00094.
- Ghorbani, A., Wexler, J., Zou, J. Y. & Kim, B. (2019). Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*.
- Gondal, W., Kohler, J., Grzeszick, R. & et. al. (2017). Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. *ICIP*.
- Gonzalez-Garcia, A., Modolo, D. & Ferrari, V. (2018). Do semantic parts emerge in convolutional neural networks? *IJCV*.
- Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016). *Deep learning*. MIT press Cambridge.
- Grauman, K. & Leibe, B. (2011). Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning*.
- Gretton, A., Fukumizu, K., Teo, C. & et. al. (2007). A kernel statistical test of independence. *NeurIPS*.
- Guo, G., Han, J., Wan, F. & Zhang, D. (2021). Strengthen learning tolerance for weakly supervised object localization. *CVPR*.
- Guo, H., Xu, M., Chi, Y., Zhang, L. & Hua, X. (2020). Weakly supervised organ localization with attention maps regularized by local area reconstruction. *MICCAI*.
- Gupta, S., Lakhotia, S., Rawat, A. & Tallamraju, R. (2022). VITOL: Vision transformer for weakly supervised object localization. *CVPR*.
- H. Touvron, H., Cord, M., Douze, M. & et. al. (2021). Training data-efficient image transformers & distillation through attention. *ICCV*.
- Hafiz, A. M., Parah, S. A. & Bhat, R. U. A. (2021). Attention mechanisms and deep learning for machine vision: A survey of the state of the art. *CoRR*, abs/arXiv:2106.07550.
- Hajimiri, S., Ayed, I.-B. & Dolz, J. (2024). Pay Attention to Your Neighbours: Training-Free Open-Vocabulary Semantic Segmentation. *arXiv preprint arXiv:2404.08181*.
- Haller, E. & Leordeanu, M. (2017). Unsupervised Object Segmentation in Video by Efficient Selection of Highly Probable Positive Features. *ICCV*.

- Han, M., Wang, Y., Chang, X. & Qiao, Y. (2020). Mining Inter-Video Proposal Relations for Video Object Detection. *ECCV*.
- Han, S., Mao, H. & Dally, W. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *CoRR*, abs/arXiv:1510.00149.
- Haris, M., Shakhnarovich, G. & Ukita, N. (2018). Deep back-projection networks for super-resolution. *CVPR*.
- Hartmann, G., Grundmann, M., Hoffman, J. & et. al. (2012). Weakly Supervised Learning of Object Segmentations from Web-Scale Video. *ECCV*.
- He, J., Chen, J., Liu, S. & et. al. (2022). Transfg: A transformer architecture for fine-grained recognition. *AAAI*.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *CVPR*.
- Heusel, M., Ramsauer, H., Unterthiner, T. & et. al. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *NeurIPS*.
- Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.
- Hosang, J., Benenson, R. & Schiele, B. (2014). How good are detection proposals, really? *BMVC*.
- Howard, A., Zhu, M., Chen, B. & et. al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.
- Hu, J., Cao, L., Lu, Y. & et. al. (2021a). Istr: End-to-end instance segmentation with transformers. *CoRR*, abs/arXiv:2105.00637.
- Hu, J., Ji, R., Ye, Q. & et. al. (2021b). Architecture Disentanglement for Deep Neural Networks. *ICCV*.
- Huang, Z., Zou, Y., Bhagavatula, V. & Huang, D. (2020). Comprehensive attention self-distillation for weakly-supervised object detection. *CoRR*, abs/arXiv:2010.12023.
- Ilse, M., Tomczak, J. & Welling, M. (2018). Attention-based deep multiple instance learning. *ICML*.
- Izadyazdanabadi, M., Belykh, E., Cavallo, C. & et. al. (2018). Weakly-supervised learning-based feature localization for confocal laser endomicroscopy glioma images. *International Conference on Medical Image Computing and Computer-Assisted Intervention*.
- Jain, S. & Grauman, K. (2014). Supervoxel-consistent foreground propagation in video. *ECCV*.
- Jalwana, M., Akhtar, N., Bennamoun, M. & Mian, A. (2021). CAMERAS: Enhanced Resolution And Sanity preserving Class Activation Mapping for image saliency. *CVPR*.

- James, H. (2019). More Than 500 Hours Of Content Are Now Being Uploaded To YouTube Every Minute, Website: <https://www.tubefilter.com/2019/05/07/number-hours-video-uploaded-to-youtube-per-minute/> (Accessed May 2022).
- James, S., Wohlhart, P., Kalakrishnan, M. & et. al. (2019). Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. *CVPR*.
- Jerripothula, K. R., Cai, J. & Yuan, J. (2016). CATS: Co-saliency Activated Tracklet Selection for Video Co-localization. *ECCV*.
- Jiang, P.-T., Hou, Q., Cao, Y., Cheng, M.-M., Wei, Y. & Xiong, H.-K. (2019). Integral object mining via online attention accumulation. *ICCV*.
- Jiang, P., Zhang, C., Hou, Q., Cheng, M. & Wei, Y. (2021). LayerCAM: Exploring Hierarchical Class Activation Maps for Localization. *TIP*.
- Joulin, A., Tang, K. & Fei-Fei, L. (2014). Efficient image and video co-localization with frank-wolfe algorithm. *ECCV*.
- Kalogeiton, V., Ferrari, V. & Schmid, C. (2016). Analysing Domain Shift Factors between Videos and Images for Object Detection. *TPAMI*.
- Kawulok, M., Benecki, P., Hrynczenko, K. & et. al. (2019). Deep learning for fast super-resolution reconstruction from multiple images. *RTIP&DL*.
- Kervadec, H., Dolz, J., Tang, M., Granger, E., Boykov, Y. & Ayed, I. B. (2019a). Constrained-CNN losses for weakly supervised segmentation. *MIA*.
- Kervadec, H., Dolz, J., Yuan, J., Desrosiers, C., Granger, E. & IB.Ayed. (2019b). Constrained Deep Networks: Lagrangian Optimization via Log-Barrier Extensions. *CoRR*, abs/1904.04205.
- Khuran, A., Lohani, B., Bibhu, V. & Kushwaha, P. (2021). An AI Integrated Face Detection System for Biometric Attendance Management. *ICIEM*.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., F. Viegas, F. et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *ICCV*, pp. 2668–2677.
- Kim, E., Kim, S., Lee, J. & et. al. (2022). Bridging the Gap between Classification and Localization for Weakly Supervised Object Localization. *CVPR*.
- Kim, J., Lee, J. & Lee, K. (2016a). Accurate image super-resolution using very deep convolutional networks. *CVPR*.
- Kim, J., Choe, J., Yun, S. & Kwak, N. (2021). Normalization Matters in Weakly Supervised Object Localization. *ICCV*.

- Kim, J., Choo, W. & Song, H. (2020). Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup. *ICML*.
- Kim, J., Lee, J. K. & Lee, K. M. (2016b). Deeply-recursive convolutional network for image super-resolution. *CVPR*.
- Kindermans, P., Hooker, S., Adebayo, J. & et. al. (2019). The (Un)reliability of Saliency Methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*.
- Kirillov, A., Mintun, E., Ravi, N. & et. al. (2023). Segment anything. *ICCV*.
- Koh, Y., Jang, W. & Kim, C. (2016). POD: Discovering Primary Objects in Videos Based on Evolutionary Refinement of Object Recurrence, Background, and Primary Object Models. *CVPR*.
- Kolesnikov, A. & Lampert, C. (2016). Seed, Expand and Constrain: Three Principles for Weakly Supervised Image Segmentation. *ECCV*.
- Krahenbuhl, P. & Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*.
- Krizhevsky, A., Sutskever, I. & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NeurIPS*.
- Kvist, M. (2020). Telecom Equipment Segmentation and Detection Using Drone Images.
- Kwak, S., Cho, M., Laptev, I., Ponce, J. & Schmid, C. (2015). Unsupervised Object Discovery and Tracking in Video Collections. *ICCV*.
- LeCun, Y., Denker, J. & Solla, S. (1990). Optimal brain damage. *NeurIPS*.
- Ledig, C., Theis, L., Huszar, F. & et. al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*.
- Lee, J., Kim, E., Lee, S. & et. al. (2019). Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. *CVPR*.
- Lee, S., Mun, C., Uh, Y. & et. al. (2024). Discovering an inference recipe for weakly-supervised object localization. *PR*.
- Lee, Y., Kim, J. & Grauman, K. (2011). Key-segments for video object segmentation. *ICCV*.
- Li, C., Yang, J., Zhang, P. & et. al. (2022). Efficient self-supervised vision transformers for representation learning. *ICLR*.
- Li, K., Wu, Z., Peng, K. et al. (2018). Tell Me Where to Look: Guided Attention Inference Network. *CVPR*.

- Li, M. (2022). CaFT: Clustering and Filter on Tokens of Transformer for Weakly Supervised Object Localization. *CoRR*, abs/arXiv:2201.00475.
- Li, W., Geng, C. & Chen, S. (2020). Leave Zero Out: Towards a No-Cross-Validation Approach for Model Selection. *CoRR*, abs/2012.13309.
- Li, Y., Ji, R., Lin, S. & et. al. (2019). Interpretable Neural Network Decoupling. *ECCV*.
- Lim, B., Son, S., Kim, H., Nah, S. & Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. *CVPRw*.
- Lin, M., Chen, Q. & Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.
- Lin, Y., Chen, M., Wang, W. & et. al. (2023). CLIP is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. *CVPR*.
- Liu, T., Yuan, Z., Sun, J. & et. al. (2010). Learning to detect a salient object. *TPAMI*.
- Liu, X., Tao, D., Song, M. & et. al. (2014). Weakly Supervised Multiclass Video Segmentation. *CVPR*.
- Liu, Z., Wang, L., Li, C. & et. al. (2019). Image Super-Resolution via Attention Based Back Projection Networks. *ICCVW*.
- Lodwich, A., Rangoni, Y. & Breuel, T. M. (2009). Evaluation of robustness and performance of Early Stopping Rules with Multi Layer Perceptrons. *IJCNN*.
- Lu, W., Jia, X., Xie, W. & et. al. (2020a). Geometry constrained weakly supervised object localization. *ECCV*.
- Lu, X., Wang, W., Shen, J., Tai, Y., Crandall, D. & Hoi, S. (2020b). Learning video object segmentation from unlabeled videos. *CVPR*.
- Luddecke, T. & Ecker, A. (2022). Image segmentation using text and image prompts. *CVPR*.
- Lundberg, S. & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. *NeurIPS*.
- Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W. & Kim, T. (2021). Multiple object tracking: A literature review. *AI*.
- Macenko, M., Niethammer, M., Marron, J. & et. al. (2009). A method for normalizing histology slides for quantitative analysis. *SBI*.
- Magee, D., Treanor, D., Crellin, D. & et. al. (2009). Colour normalisation in digital histopathology images. *MICCAIw*.
- Mahsereci, M., Balles, L., Lassner, C. & Hennig, P. (2017). Early Stopping without a Validation Set. *CoRR*, abs/1703.09580.

- Mai, J., Yang, M. & Luo, W. (2020). Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. *CVPR*.
- Manen, S., Guillaumin, M. & Gool, L. V. (2013). Prime Object Proposals with Randomized Prim's Algorithm. *ICCV*.
- Martens, M., Izzo, D., Krzic, A. & Cox, D. (2019). Super-resolution of PROBA-V images using convolutional neural networks. *Astrodynamics*.
- Meethal, A., Pedersoli, M., Belharbi, S. & Granger, E. (2021). Convolutional STN for Weakly Supervised Object Localization. *ICPR*.
- Meethal, A., Pedersoli, M., Zhu, Z., Romero, F. & Granger, E. (2022). Semi-Weakly Supervised Object Detection by Sampling Pseudo Ground-Truth Boxes. *IJCNN*.
- Meng, M., Zhang, T., Tian, Q. & et. al. (2021). Foreground activation maps for weakly supervised object localization. *ICCV*.
- Meng, M., Zhang, T., Yang, W. & et. al. (2022a). Diverse Complementary Part Mining for Weakly Supervised Object Localization. *TIP*.
- Meng, M., Zhang, T., Zhang, Z. & et. al. (2022b). Task-aware weakly supervised object localization with transformer. *TPAMI*.
- Meng, M., Zhang, T., Zhang, Z., Zhang, Y. & Wu, F. (2022c). Adversarial Transformers for Weakly Supervised Object Localization. *IEEE Transactions on Image Processing*, 31, 7130–7143.
- Minsong, K., Youngjung, U., Wonyoung, L. & Hyeran, B. (2020). In-sample Contrastive Learning and Consistent Attention for Weakly Supervised Object Localization. *CoRR*, abs/arXiv:2009.12063.
- Montavon, G., Lapuschkin, S., Binder, A., Samek, W. & Muller, K. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *PR*.
- Morgan, N. & Bourlard, H. (1989). Generalization and parameter estimation in feedforward nets: Some experiments. *NeurIPS*.
- Murtaza, S., Belharbi, S., Pedersoli, M., Sarraf, A. & Granger, E. (2022). Constrained Sampling for Class-Agnostic Weakly Supervised Object Localization. *Montreal AI Symposium*.
- Murtaza, S., Belharbi, S., Pedersoli, M., Sarraf, A. & Granger, E. (2023a, January). Discriminative Sampling of Proposals in Self-Supervised Transformers for Weakly Supervised Object Localization. *WACVw*.
- Murtaza, S., Belharbi, S., Pedersoli, M., Sarraf, A. & Granger, E. (2023b). DiPS: Discriminative pseudo-label sampling with self-supervised transformers for weakly supervised object localization. *Image and Vision Computing*.

- Murtaza, S., Belharbi, S., Pedersoli, M. & Granger, E. (2025). A Realistic Protocol for Evaluation of Weakly Supervised Object Localization. *WACV*.
- Naidu, R., Ghosh, A., Maurya, Y. & et. al. (2020). IS-CAM: Integrated Score-CAM for axiomatic-based explanations. *CoRR*, abs/arXiv:2010.03023.
- Nguyen, N., Anger, J., Davy, A. & et. al. (2021). Self-Supervised Multi-Image Super-Resolution for Push-Frame Satellite Images. *CVPR*.
- Nguyen, T., Raghu, M. & Kornblith, S. (2020). Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *CoRR*, abs/arXiv:2010.15327.
- Niethammer, M., Borland, D., Marron, J. & et. al. (2010). Appearance normalization of histology slides. *MLMIw*.
- Oliver, A., Odena, A., Raffel, C. & et. al. (2018). Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. *NeurIPS*.
- Omeiza, ., Speakman, S., Cintas, C. & Weldermariam, K. (2019). Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *CoRR*, abs/arXiv:1908.01224.
- Oquab, M., Bottou, L., Laptev, I. & Sivic, J. (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks. *CVPR*.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.
- Pan, X., Gao, Y., Lin, Z. & et. al. (2021). Unveiling the potential of structure preserving for weakly supervised object localization. *CVPR*.
- Pan, Y., Yao, Y., Cao, Y. & et. al. (2023). Coarse2fine: local consistency aware re-prediction for weakly supervised object localization. *AAAI*.
- Papazoglou, A. & Ferrari, V. (2013). Fast Object Segmentation in Unconstrained Video. *ICCV*.
- Park, D., Hendricks, L., Akata, Z. & et. al. (2018). Multimodal explanations: Justifying decisions and pointing to the evidence. *CVPR*.
- Pathak, D., Krahenbuhl, R. & Darrell, T. (2015). Constrained convolutional neural networks for weakly supervised segmentation. *ICCV*.
- Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L. V., Gross, M. & Sorkine-Hornung, A. (2016). A benchmark dataset and evaluation methodology for video object segmentation. *CVPR*.
- Pinheiro, P. & Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. *CVPR*.

- Prest, A., Leistner, C., Civera, J. & et. al. (2012). Learning object class detectors from weakly annotated video. *CVPR*.
- Quiroga, R., Reddy, L., Kreiman, G., Koch, C. & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*.
- Radford, A., Kim, J., Hallacy, C. & et. al. (2021). Learning transferable visual models from natural language supervision. *ICML*.
- Rahimi, A., Shaban, A., Ajanthan, T. & et. al. (2020). Pairwise similarity knowledge transfer for weakly supervised object localization. *ECCV*.
- Ramaswamy, H. G. et al. (2020). Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. *WACV*.
- Real, E., Shlens, J., Mazzocchi, S., Pan, X. & Vanhoucke, V. (2017). Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *CVPE*.
- Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? *ICML*.
- Redmon, J., Divvala, S., Girshick, R. et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. *CVPR*.
- Reinhard, E., Adhikhmin, M., Gooch, B. & Shirley, P. (2001). Color transfer between images. *CGA*.
- Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*.
- Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier.
- Rochan, M., Rahman, S., Bruce, N. D. B. & Wang, Y. (2016). Weakly supervised object localization and segmentation in videos. *Image and Vision Computing*, 56, 1–12.
- Ronneberger, O., Fischer, P. & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *MICCAI*.
- Rony, J., Belharbi, S., Dolz, J. & et. al. (2023). Deep Weakly-Supervised Learning Methods for Classification and Localization in Histology Images: A Survey. *MLBI*.
- Rother, C., Kolmogorov, V. & Blake, A. (2004). "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM TG*, 23(3), 309–314.
- Ruifrok, A. & Johnston, D. (2001). Quantification of histochemical staining by color deconvolution. *AQCH*.

- Russakovsky, O., Deng, J., Su, H. & et. al. (2015a). ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3), 211-252.
- Russakovsky, O., Li, L. & Fei-Fei, L. (2015b). Best of both worlds: Human-machine collaboration for object annotation. *CVPR*.
- S, K., Norouzi, M., Lee, H. & Hinton, G. (2019). Similarity of neural network representations revisited. *ICCV*.
- Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. & Norouzi, M. (2022). Image super-resolution via iterative refinement. *PAMI*.
- Saito, K., Kim, D., Teterwak, P. & et. al. (2021). Tune it the Right Way: Unsupervised Validation of Domain Adaptation via Soft Neighborhood Density. *ICCV*.
- Saleh, F., Aliakbarian, M., Salzmann, M. & et. al. (2017). Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. *ICCV*.
- Salimans, T., Goodfellow, I., Zaremba, W. & et. al. (2016). Improved Techniques for Training GANs. *NeurIPS*.
- Salvador, T., FATRAS, K., Mitliagkas, I. & Oberman, A. (2022). A Reproducible and Realistic Evaluation of Partial Domain Adaptation Methods. *NeurIPS*.
- Sedai, S., Mahapatra, D., Ge, Z. & et. al. (2018). Deep multiscale convolutional feature learning for weakly supervised localization of chest pathologies in x-ray images. *International Workshop on Machine Learning in Medical Imaging*.
- Selvaraju, R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *ICCV*.
- Settles, B. (2009). *Active learning literature survey*.
- Shao, F., Chen, L., Shao, J. & et. al. (2022). Deep Learning for Weakly-Supervised Object Detection and Localization: A Survey. *Neurocomputing*.
- Sharir, G., Noy, A. & Zelnik-Manor, L. (2021). An image is worth 16x16 words, what is a video worth? *ICLR*.
- Shekkizhar, S. & Ortega, A. (2020). DeepNNK: Explaining deep models and their generalization using polytope interpolation. *CoRR*, abs/2007.10505.
- Shi, M. & Ferrari, V. (2016). Weakly supervised object localization using size estimates. *ECCV*.
- Shi, Y. & Sha, F. (2012). Information-Theoretical Learning of Discriminative Clusters for Unsupervised Domain Adaptation. *ICLM*.

- Simeoni, O., Puy, G., Vo, H. & et. al. (2021). Localizing Objects with Self-Supervised Transformers and no Labels. *CoRR*, abs/arXiv:2109.14279.
- Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/arXiv:1409.1556.
- Simonyan, K., Vedaldi, A. & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Singh, K. & Lee, Y. (2017). Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. *ICCV*.
- Sirinukunwattana, K., Snead, D. & Rajpoot, N. (2015). A Stochastic Polygons Model for Glandular Structures in Colon Histology Images. *TMI*.
- Sirinukunwattana, K., Pluim, J., Chen, H. & et. al. (2017). Gland segmentation in colon histology images: The glas challenge contest.
- Song, H., Kim, M., Park, D. & et. al. (2022). Learning from noisy labels with deep neural networks: A survey. *TNNLS*.
- Sovrasov, V. (2021). Flops counter for convolutional networks in pytorch framework. Retrieved from: github.com/sovrasov/flops-counter.pytorch.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *JMLR*.
- Stretcu, O. & Leordeanu, M. (2015). Multiple Frames Matching for Object Discovery in Video. *BMVC*.
- Su, H. & Yang, M. (2024). A consistency and integration model with adaptive thresholds for weakly supervised object localization. *IJCAI*.
- Su, H., Ye, Y., Chen, Z., Song, M. & Cheng, L. (2022). Re-Attention Transformer for Weakly Supervised Object Localization. *BMVC*.
- Su, J., Cheng, Z. & Maji, S. (2021). A Realistic Evaluation of Semi-Supervised Learning for Fine-Grained Classification. *CVPR*.
- Sun, C., Paluri, M., Collobert, R., Nevatia, R. & Bourdev, L. (2016). Pronet: Learning to propose object-specific boxes for cascaded neural networks. *CVPR*.
- Sun, G., Probst, T., Paudel, D. & et. al. (2021). Task Switching Network for Multi-task Learning. *ICCV*.
- Sun, Y., Ravi, S. N. & Singh, V. (2019). Adaptive activation thresholding: Dynamic routing type behavior for interpretability in convolutional neural networks. *ICCV*, pp. 4938–4947.

- Sundaram, N., Brox, T. & Keutzer, K. (2010). Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. *ECCV*.
- Sundararajan, M., Taly, A. & Yan, Q. (2017). Axiomatic attribution for deep networks. *ICML*.
- Szegedy, C., Liu, W., Jia, Y. & et. al. (2015). Going deeper with convolutions. *CVPR*.
- Tagaris, T., Sdraka, M. & Stafylopatis, A. (2019). High-Resolution Class Activation Mapping. *2019 IEEE International Conference on Image Processing (ICIP)*.
- Tai, Y., Yang, J., Liu, X. & Xu, C. (2017a). Memnet: A persistent memory network for image restoration. *ICCV*.
- Tai, Y., Yang, J. & X. Liu, X. (2017b). Image super-resolution via deep recursive residual network. *CVPR*.
- Tan, C., Gu, G., Ruan, T. & et. al. (2022). Dual-gradients localization framework with skip-layer connections for weakly supervised object localization. *TMM*.
- Tang, K., Sukthankar, R., Yagnik, J. & Fei-Fei, L. (2013). Discriminative segment annotation in weakly labeled video. *CVPR*.
- Tang, K. D., Joulin, A., Li, L. & Fei-Fei, L. (2014). Co-localization in Real-World Images. *CVPR*.
- Tang, M., Perazzi, F., Djelouah, A. & et. al. (2018). On regularized losses for weakly-supervised cnn segmentation. *ECCV*.
- Teh, E., Rochan, M. & Wang, Y. (2016). Attention Networks for Weakly Supervised Object Localization. *BMVC*.
- Tokmakov, P., Alahari, K. & Schmid, C. (2016). Weakly-Supervised Semantic Segmentation Using Motion Cues. *ECCV*.
- Touvron, H., Cord, M., Sablayrolles, A. & et. al. (2021). Going deeper with image transformers. *CoRR*, abs/arXiv:2103.17239.
- Tsai, Y., Zhong, G. & Yang, M. (2016). Semantic co-segmentation in videos. *ECCV*.
- Uddin, A. S., Monira, M., Shin, W., Chung, T. & Bae, S. (2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization. *ICLR*.
- Uijlings, J., Sande, K. V. D., Gevers, T. & Smeulders, A. (2013). Selective search for object recognition. *IJCV*.
- Umer, S., Dawood, H., Yousaf, M. & et. al. (2021). Efficient foreground object segmentation from video by Probability Weighted Moments. *Optik*.

- Vaithyanathan, S. & Dom, B. (1999). Generalized model selection for unsupervised learning in high dimensions. *NeurIPS*.
- Van Engelen, J. E. & Hoos, H. H. (2020). A survey on semi-supervised learning. *ML*.
- Vaswani, A., Shazeer, N., Parmar, N. & et. al. (2017). Attention is all you need. *NeurIPS*.
- Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P. (2008). Extracting and composing robust features with denoising autoencoders. *ICCV*.
- Viroli, C. & McLachlan, G. (2019). Deep Gaussian mixture models. *SC*.
- Wah, C., Branson, S., Steve, W. & et. al. (2011). *The Caltech-UCSD Birds-200-2011 Dataset* (Report n°CNS-TR-2011-001).
- Wan, F., Wei, P., Jiao, J., Han, Z. & Ye, Q. (2018). Min-entropy latent model for weakly supervised object detection. *CVPR*.
- Wang, C., Xu, R., Xu, S. & et. al. (2024). Exploring intrinsic discrimination and consistency for weakly supervised object localization. *TIP*.
- Wang, F., Mei, J. & Yuille, A. (2023). SCLIP: Rethinking Self-Attention for Dense Vision-Language Inference. *CoRR*, abs/arXiv:2312.01597.
- Wang, H., Z.Wang, Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P. & Hu, X. (2020a). Score-CAM: Score-weighted visual explanations for convolutional neural networks. *CVPRw*.
- Wang, H., Naidu, R., Michael, J. & Kundu, S. S. (2020b). SS-CAM: Smoothed Score-CAM for sharper visual feature localization. *CoRR*, abs/arXiv:2006.14255.
- Wang, J. & Kang, G. (2024). Learn to Rectify the Bias of CLIP for Unsupervised Semantic Segmentation. *CVPR*.
- Wang, M., Zhang, Q., Yang, J. & et. al. (2018a). Graph-adaptive pruning for efficient inference of convolutional neural networks. *CoRR*, abs/arXiv:1811.08589.
- Wang, X., Yan, Y., Tang, P., Bai, X. & Liu, W. (2018b). Revisiting multiple instance neural networks. *PR*.
- Wang, X., You, S., Li, X. & Ma, H. (2018c). Weakly-supervised semantic segmentation by iteratively mining common object features. *CVPR*.
- Wang, X., Yu, K., Wu, S. & et. al. (2018d). Esrgan: Enhanced super-resolution generative adversarial networks. *ECCVw*.
- Wang, Y., Perazzi, F., McWilliams, B. & et. al. (2018e). A fully progressive approach to single-image super-resolution. *CVPRw*.

- Wang, Y., Su, H., Zhang, B. & Hu, X. (2018f). Interpret neural networks by identifying critical data routing paths. *CVPR*, pp. 8906–8914.
- Wang, Y., Zhang, J., Kan, M. & et. al. (2020c). Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. *CVPR*.
- Wei, J., Wang, Q., Li, Z. & et. al. (2021). Shallow Feature Matters for Weakly Supervised Object Localization. *CVPR*.
- Wei, J., Wang, S., Zhou, S. & et. al. (2022). Weakly supervised object localization through inter-class feature similarity and intra-class appearance consistency. *ECCV*, pp. 195–210.
- Wei, Y., Feng, J., Liang, X. & et. al. (2017). Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. *CVPR*.
- Wei, Y., Xiao, H., Shi, H. & et. al. (2018). Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. *CVPR*.
- Welinder, P., Branson, S., Mita, T. & et. al. (2010). *Caltech-UCSD Birds 200* (Report n°CNS-TR-2010-001).
- Wu, H., Judd, P., Zhang, X. & et. al. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *CoRR*, abs/arXiv:2004.09602.
- Wu, H., Xiao, B., Codella, N. & et. al. (2021). Cvt: Introducing convolutions to vision transformers. *CoRR*, abs/arXiv:2103.15808.
- Wu, P., Zhai, W. & Cao, Y. (2022). Background activation suppression for weakly supervised object localization. *CVPR*.
- Wu, P., Zhai, W., Cao, Y., Luo, J. & Zha, Z. (2023). Spatial-Aware Token for Weakly Supervised Object Localization. *ICCV*.
- Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollar, P. & Girshick, R. (2021). Early convolutions help transformers see better.
- Xie, J., Luo, C., Zhu, X. & et. al. (2021). Online refinement of low-level feature based activation map for weakly supervised object localization. *ICCV*.
- Xie, J., Hou, X., Ye, K. & Shen, L. (2022a). CLIMS: Cross language image matching for weakly supervised semantic segmentation. *CVPR*.
- Xie, J., Xiang, J., Chen, J. & et. al. (2022b). C2AM: Contrastive learning of Class-agnostic Activation Map for Weakly Supervised Object Localization and Semantic Segmentation. *CVPR*.
- Xie, J., Luo, Z., Li, Y. & et. al. (2023). Open-World Weakly-Supervised Object Localization. *CoRR*, abs/:2304.08271.

- Xie, Z., Zhang, Z., Cao, Y. & et. al. (2022c). SimMIM: A simple framework for masked image modeling. *CVPR*.
- Xu, C., Xiong, C. & Corso, J. J. (2012). Streaming Hierarchical Video Segmentation. *ECCV*.
- Xu, J., Hou, J., Zhang, Y. & et. al. (2022). CREAM: Weakly Supervised Object Localization via Class RE-Activation Mapping. *CVPR*.
- Xu, K., Ba, J., Kiros, R. & et. al. (2015). Show, attend and tell: Neural image caption generation with visual attention. *ICCV*.
- Xu, R., Luo, Y., Hu, H. & et. al. (2023). Rethinking the localization in weakly supervised object localization. *ACM MM*.
- Xue, F., Tan, Z., Zhu, Y. & et. al. (2022). Coarse-to-Fine Cascaded Networks with Smooth Predicting for Video Facial Expression Recognition. *CVPRW*.
- Xue, H., Liu, C., Wan, F. & et. al. (2019). Danet: Divergent activation for weakly supervised object localization. *ICCV*.
- Yan, Y., Xu, C., Cai, D. & Corso, J. J. (2017). Weakly Supervised Actor-Action Segmentation via Robust Multi-task Ranking. *CVPR*.
- Yang, J., Qian, H., Xu, Y. & Xie, L. (2023). Can We Evaluate Domain Adaptation Models Without Target-Domain Labels? *CoRR*, abs/2305.18712.
- Yang, S., Kim, Y., Kim, Y. & Kim, C. (2020). Combinational class activation maps for weakly supervised object localization. *WACV*.
- Yang, X. & Gong, X. (2024). Foundation model assisted weakly supervised semantic segmentation. *WACV*.
- Yasuki, S. & Taki, M. (2024). Cam Back Again: Large kernel cnns from a weakly supervised object localization perspective. *CVPR*.
- You, K., Wang, X., Long, M. & Jordan, M. I. (2019). Towards Accurate Model Selection in Deep Unsupervised Domain Adaptation. *ICML*.
- Yu, F., Koltun, V. & Funkhouser, T. (2017a). Dilated residual networks. *CVPR*.
- Yu, Y., Choi, J., Kim, Y. & et. al. (2017b). Supervising neural attention models for video captioning by human gaze data. *CVPR*.
- Yuan, S., Feng, L. & Liu, T. (2024). Early Stopping Against Label Noise Without Validation Data. *ICLR*.
- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J. & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. *ICCV*.

- Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*.
- Zhang, A., Ling, Z. & Wang, Y. (2023). Multi-Layer Decoupling Attention Network for Weakly Supervised Object Localization. *TMM*.
- Zhang, B., Xiao, J., Wei, Y., Sun, M. & Huang, K. (2020a). Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. *AAAI*.
- Zhang, C., Cao, Y. & Wu, J. (2020b). Rethinking the Route Towards Weakly Supervised Object Localization. *CVPR*.
- Zhang, C., Cao, Y. & Wu, J. (2020c). Rethinking the route towards weakly supervised object localization. *CVPR*.
- Zhang, D., Javed, O. & Shah, M. (2014). Video Object Co-segmentation by Regulated Maximum Weight Cliques. *ECCV*.
- Zhang, D., Han, J., Jiang, L., Ye, S. & Chang, X. (2017). Revealing Event Saliency in Unconstrained Video Collection. *TIP*, 26(4), 1746–1758.
- Zhang, D., Han, J., Yang, L. & Xu, D. (2020d). SPFTN: A Joint Learning Framework for Localizing and Segmenting Objects in Weakly Labeled Videos. *TPAMI*.
- Zhang, D., Guo, G., Zeng, W. & et. al. (2024). Generalized weakly supervised object localization. *TNNLS*.
- Zhang, H. (2017). mixup: Beyond empirical risk minimization. *CoRR*.
- Zhang, J., Bargal, S., Lin, Z. & et. al. (2018a). Top-down neural attention by excitation backprop. *IJCV*.
- Zhang, R., Isola, P., Efros, A. A. & et. al. (2018b). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *CVPR*.
- Zhang, W., Qiu, F., Wang, S. & et. al. (2022). Transformer-based Multimodal Information Fusion for Facial Expression Analysis. *CVPRW*.
- Zhang, X., Wei, Y., Feng, J., Yang, Y. & Huang, T. (2018c). Adversarial complementary learning for weakly supervised object localization. *CVPR*.
- Zhang, X., Wei, Y., Kang, G. & et. al. (2018d). Self-produced guidance for weakly-supervised object localization. *ECCV*.
- Zhang, X., Wei, Y. & Yang, Y. (2020e). Inter-image communication for weakly supervised localization. *ECCV*.
- Zhang, X., Wei, Y., Yang, Y. & Wu, F. (2020f). Rethinking localization map: Towards accurate object perception with self-enhancement maps. *CoRR*, abs/arXiv:2006.05220.

- Zhang, Y., Chen, X., Li, J. & et. al. (2015). Semantic object segmentation via detection in weakly labeled video. *CVPR*.
- Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B. & Fu, Y. (2018e). Image super-resolution using very deep residual channel attention networks. *ECCV*.
- Zhang, Y., Tian, Y., Kong, Y., Zhong, B. & Fu, Y. (2018f). Residual dense network for image super-resolution. *ICCV*.
- Zhao, W., Cui, C., Jun, J. & Long, X. (2021). Accurate multi-image super-resolution using deep residual networks. *OIMT*.
- Zhao, Y., Ye, Q., Wu, W. & et. al. (2023). Generative prompt model for weakly supervised object localization. *CVPR*.
- Zheng, S., Lu, J., Zhao, H. & et. al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *CVPR*.
- Zhou, B., Khosla, A., Lapedriza, A. & et. al. (2016). Learning deep features for discriminative localization. *CVPR*.
- Zhou, D., Kang, B., Jin, X. & et. al. (2021). Deepvit: Towards deeper vision transformer. *CoRR*, abs/arXiv:2103.11886.
- Zhou, K. & Kainz, B. (2018). Efficient image evidence analysis of cnn classification results. *CoRR*, abs/arXiv:1801.01693.
- Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q. & Jiao, J. (2018). Weakly supervised instance segmentation using class peak response. *CVPR*.
- Zhou, Z. (2004). Multi-instance learning: A survey. *Tech.Rep NU*.
- Zhou, Z. (2018). A brief introduction to weakly supervised learning. *NSR*.
- Zhu, L., Chen, Q., Jin, L. & et. al. (2022a). Bagging Regional Classification Activation Maps for Weakly Supervised Object Localization. *ECCV*.
- Zhu, L., She, Q., Chen, Q. & et. al. (2022b). Weakly Supervised Object Localization as Domain Adaption. *CVPR*.
- Zhu, L., She, Q., Chen, Q. & et. al. (2024). Boosting Weakly Supervised Object Localization and Segmentation With Domain Adaption. *TPAMI*.
- Zhu, L., She, Q., Chen, Q., Meng, X., Geng, M., Jin, L., Jiang, Z., Qiu, B., You, Y., Zhang, Y. et al. (2021). Background-aware Classification Activation Map for Weakly Supervised Object Localization. *CoRR*, abs/arXiv:2112.14379.
- Zhu, Y., Zhou, Y., Ye, Q. et al. (2017). Soft proposal networks for weakly supervised object localization. *ICCV*.

Zintgraf, L., Cohen, T., Adel, T. & Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis. *CoRR*, abs/arXiv:1702.04595.