Learning to Localize Objects with Limited Supervision

by

Akhil Pilakkatt Meethal

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

MONTREAL, NOVEMBER 10, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Marco Pedersoli, Thesis Supervisor
Department of Systems Engineering, École de technologie supérieure

Mr. Eric Granger, Thesis Co-Supervisor
Department of Systems Engineering, École de technologie supérieure

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

Mr. Ismail Ben Ayed, Member of the Jury
Department of Systems Engineering, École de technologie supérieure

Mr. Samuel Foucher, External Independent Examiner
Department of Applied Geomatics, University of Sherbrooke

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "OCTOBER 20, 2023"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

Though only my name appears on the cover of this thesis, the work presented in this thesis would not have been possible without the constant support and encouragement of many wonderful human beings.

First and foremost, I would like to express my sincere gratitude to my advisors Prof. Marco Pedersoli and Prof. Eric Granger. One of the major struggles for me at the beginning of my Ph.D. was the habit of constantly changing research directions. I used to monitor some global metrics over the dataset and evaluate the research directions only based on those metric values. Due to this, when the metrics were not improving, I kept on changing to new directions thinking that my direction was wrong. I am very grateful to Marco for teaching me to meticulously evaluate the research directions, going deep with rigorous scientific processes. From Eric, I learned the standardization in every stage of research including critically studying past research, identifying the research gap, designing your solutions to solve it, and clearly communicating this in the manuscript.

I am fortunate to receive immense support from friends locally and remotely during this thesis work. This includes hours-long discussions on research topics, reading groups, implementation tricks, paper criticizing, and many more. Locally from ETS, I would like to express my sincere thanks to the labmates Bala, Saypra, Soufiane, Madhu, Ziko, Karthik, Shambhavi, Sajjad, Sukesh, Praveen, Heitor, Navaneeth and Atif. During this time, I also made great connections outside the lab with other friends from ETS including Shubham, Navaneeth, Kanika, and Olivier. I would also like to express my sincere thanks to Sweta, Vaisahkh, Rohith, Sangeeth, Sobhan, and Nidhi who supported me immensely from remote. My internship at Ericsson was a great experience and it really shaped my understanding of the research to production path and the challenges. I am very grateful to my collaborators from Ericsson Francisco and Zhongwen for this. The CEO of my first company Abe, with whom I share a great bond also shaped this understanding.

Coming to a new country, managing severe illness and hospitalization alone, and surviving the isolation period during a pandemic was not easy for me without having support from family. I

am very grateful to Binu, Prasanth, Selva, and Roshna who helped me like a family during all these difficult times. Needless to say, I had a good time exploring Canada and doing many new activities for which I am thankful to Sebin, Nafseer, Bhagya, and Jyothis. Going far from home made my connection to family even stronger and I always kept the communication through daily calls. I would like to thank my father, mother, and my sister for their care, love, and support from miles away. Also, I express many thanks to my grandmother who always kept me in her prayers. I would like to take this opportunity to thank my cousins too who were always there for me. From my family, I would also like to mention my uncle Dileep who has always helped in every stage of my education starting from high school.

Finally, I'd love to thank my wife Rithu for her unconditional love and support. Her support was immense to get more focused time on my research and better communication of my research materials.

Apprendre à localiser des objets avec une supervision limitée

Akhil Pilakkatt Meethal

RÉSUMÉ

Les détecteurs d'objets profonds sont omniprésents dans les industries d'aujourd'hui, répondant à de nombreuses exigences telles que l'inspection, l'automatisation, la surveillance, la surveillance, etc. détecteurs profonds affamés. L'étiquetage des images avec des cadres de délimitation est coûteux et prend du temps. Le coût est prohibitif lorsqu'une forte expertise est nécessaire pour l'étiquetage, par exemple, un radiologue étiquetant des images médicales. De plus, les détecteurs entièrement supervisés actuels ne suffiront pas à notre demande croissante car il est pratiquement impossible d'étiqueter des milliers d'images pour chaque tâche de détection d'objet que nous voulons résoudre. Ainsi, pour atténuer ce défi d'annotation pour la formation des détecteurs profonds modernes, la communauté explore de nombreuses directions, y compris la formation faiblement supervisée, auto-supervisée, semi-supervisée, adaptative au domaine et à quelques coups.

Pour contribuer à cet effort, dans cette thèse, nous avons exploré des méthodes faiblement supervisées et semi-supervisées pour entraîner des systèmes de localisation pour la localisation mono-objet et multi-objet. Avec des méthodes faiblement supervisées, nous avons observé que les chercheurs utilisent une architecture par défaut et proposent des techniques pour améliorer sa localisation. Nous avons identifié les limites des architectures par défaut pour la localisation et la détection faiblement supervisées. Ensuite, nous nous sommes concentrés sur des architectures alternatives qui répondent à ces limitations et sont faciles à utiliser. Nos architectures proposées ont également montré des performances améliorées. Pour les méthodes semi-supervisées, nous nous concentrons sur l'amélioration de leur utilisabilité sur des applications utilisant l'imagerie aérienne. L'imagerie aérienne connaît un intérêt croissant de nos jours où de grandes collections d'images sont collectées à l'aide de drones et de satellites en mode surveillance. Il n'est pas possible d'utiliser cette collection sans méthodes semi-supervisées efficaces car les étiqueter n'est tout simplement pas une option. Différent des images naturelles, les images aériennes ont une résolution élevée en pixels et les objets sont minuscules. Une application directe des méthodes modernes de détection semi-supervisée sur ces images ne donnera pas les meilleurs résultats. Nous proposons une détection semi-supervisée sur mesure pour la détection d'objets minuscules sur des images aériennes à haute résolution.

La première contribution de cette thèse est une architecture de localisation d'objet faiblement supervisée entièrement convolutive avec un composant de localisation apprenable. Différente de l'architecture CAM par défaut, notre méthode est entièrement convolutive et possède des composants séparés pour l'apprentissage de la localisation. Nous avons utilisé des transformateurs spatiaux de manière convolutive pour apprendre la localisation sous une forme paramétrique où la paramétrisation est de transformations affines. L'un des défis majeurs des méthodes de localisation faiblement supervisées est la localisation discriminative des régions. Dans notre architecture, cela peut être facilement réduit en spécifiant une contrainte de régularisation sur

les paramètres de localisation appris. Grâce à des études empiriques approfondies, nous avons établi une localisation améliorée et un contrôle flexible de notre projet de réseau de localisation entièrement convolutif faiblement supervisé.

La deuxième contribution propose une méthode pour réutiliser les architectures de détection d'objets existantes pour la détection d'objets faiblement supervisés au lieu du choix par défaut WSDDN. Bien qu'il existe de nombreuses architectures de détection d'objets proposées pour la détection d'objets génériques, les chercheurs travaillant sur des détecteurs faiblement supervisés utilisent l'architecture WSDDN car il n'est pas possible de traduire la supervision globale au niveau de l'image fournie par les étiquettes d'image en étiquettes locales au niveau de l'instance. Nous avons proposé une méthode de construction de pseudo-étiquettes basée sur l'échantillonnage à l'aide de laquelle les étiquettes au niveau de l'image peuvent être traduites en étiquettes au niveau de l'instance, entraînant ainsi le détecteur à l'aide de détecteurs prêts à l'emploi. Nous avons également montré que les performances du détecteur basé sur l'échantillonnage peuvent être améliorées de manière significative en utilisant des images annotées.

La troisième contribution concerne l'adaptation des détecteurs semi-supervisés grand public pour s'entraîner sur des images aériennes à haute résolution. La détection d'objets d'images aériennes peut bénéficier de manière significative si des détecteurs semi-supervisés efficaces peuvent être conçus car de nombreuses images sont collectées en mode surveillance à l'aide de drones et de satellites. Ils ne sont tout simplement pas utilisés pendant la formation en raison du manque d'annotations. Nous avons observé que les minuscules objets dans les images aériennes à haute résolution ne peuvent pas être pseudo-étiquetés efficacement pour une formation semi-supervisée. Pour résoudre ce problème, nous avons utilisé des cultures de densité où les régions à forte concentration de petits objets sont identifiées et rognées. Ces régions sont ensuite traitées par mise à l'échelle pour une meilleure détection des petits objets. Cette formation basée sur la densité des cultures est mise en œuvre dans le détecteur, ce qui donne plus de pseudo-étiquettes pour les objets minuscules, ce qui se traduit par une détection améliorée des objets semi-supervisés sur les images aériennes.

Mots-clés: détection d'objets, localisation, apprentissage faiblement supervisé, apprentissage semi-supervisé

Learning to Localize Objects with Limited Supervision

Akhil Pilakkatt Meethal

ABSTRACT

Deep object detectors are omnipresent in today's industries meeting many requirements like inspection, automation, surveillance, monitoring, etc. One of the important bottlenecks in developing today's object detection systems is the need for a huge collection of labeled data to train the data-hungry deep detectors. Labeling images with bounding boxes is expensive and time-consuming. The cost is prohibitively high when strong expertise is needed for labeling, for example, a radiologist labeling medical images. Also, the current successful fully supervised detectors won't scale for our growing demand as it is practically impossible to label thousands of images for every object detection task we want to solve. Thus to mitigate this annotation challenge for training modern deep detectors, the community is exploring many directions including weakly supervised, self-supervised, semi-supervised, domain adaptation, and few-shot training. To contribute to this effort, in this thesis, we explored weakly supervised and semi-supervised methods for training localization systems for single-object and multi-object localization. We identified the limitations and the training difficulties of the current main-stream weakly supervised and semi-supervised detection techniques. Then we proposed alternate designs and training techniques to mitigate this.

The first contribution of this thesis is a fully convolutional weakly supervised object localization architecture with a learnable localization component. Different from the default architecture CAM (Class Activation Mapping), our method is fully convolutional and has separate components for learning localization. We used spatial transformers in a convolutional fashion for learning the localization with affine transformations. One of the major challenges of weakly supervised localization methods is the localization focus on discriminative regions. In our architecture, this can be reduced easily by a regularization constraint on the learned parameters. With extensive empirical studies, we established improved localization and flexible control of our proposed fully convolutional weakly supervised localization network.

The second contribution is a method to reuse the existing fully supervised object detection architectures for weak supervision. While there are plenty of object detection architectures proposed for supervised object detection, researchers working on weakly supervised detectors use the WSDDN (Weakly Supervised Deep Detection Networks) architecture because it is straightforward to train WSDDN with weak image-level labels. WSDDN computes class probabilities of region proposals and aggregates these probabilities to produce image-level class probabilities. Given only the global image-level labels in weakly supervised settings, there is no efficient technique to label object regions using them so as to train the system using fully supervised detectors. To address this, we proposed a sampling-based pseudo-label construction method, using which region-level labels are derived from the image labels. With these pseudo-labels for regions, we can train any off-the-shelf fully supervised detection method, thus eliminating the need for customized architectures for weakly supervised object detection. We

also showed that the performance of the sampling-based detector can be improved significantly by using a few annotated images.

The third contribution is about adapting the main-stream semi-supervised detectors to train on high-resolution aerial images. Aerial image object detection can benefit significantly if effective semi-supervised detectors can be designed because plenty of images are collected in surveillance applications using drones and satellites. Those images are simply not used during training because of the lack of annotations. We observed that the tiny objects in high-resolution aerial images cannot be pseudo-labeled effectively for semi-supervised training. To fix this, our proposed zoom-in detector uses density crops where regions with high concentrations of small objects are identified and cropped out. These regions are then upscaled for better detection of small objects. This density crop-based training is implemented within the detector giving more pseudo labels for tiny objects which translates to improved semi-supervised object detection on aerial images.

Keywords: object detection, localization, weakly supervised learning, semi-supervised learning

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

LIST OF ABBREVIATIONS

ETS	École de Technologie Supérieure
ASC	Agence Spatiale Canadienne
CV	Computer Vision
OD	Object Detection
DL	Deep Learning
WSL	Weakly Supervised Learning
SSL	Semi-supervised Learning
WSOL	Weakly Supervised Object Localization
WSOD	Weakly Supervised Object Detection
SSOD	Semi-supervised Object Detection
FSOD	Fully-supervised Object Detection
Semi-WSOD	Semi-weakly Supervised Object Detection
WSDDN	Weakly Supervised Deep Detection Networks
GT	Groundtruth
UDA	Unsupervised Domain Adpatation
MIL	Multiple Instance Learning
IoU	Intersection over Union
CAM	Class Activation Mapping
RCNN	Region-Based Convolutional Neural Network

OICR	Online Instance Classifier Refinement
AP	Average Precision
mAP	Mean Average Precision
MAB	Multi-armed Bandit
STN	Spatial Transformer Networks
CSTN	Convolutional Spatial Transformer Networks
GAP	Global Average Pooling
CNN	Convolutional Neural Networks
FPN	Feature Pyramid Networks
ADL	Attention-based Dropout Layer
EMA	Exponential Moving Average
NMS	Non-Maximal Suppression
DCrop	Density Crop
CZ	Cascaded Zoom-in
FCOS	Fully Convolutional One Stage detector
FPS	Frames Per Second

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

w_k^c	Weight corresponding to class c for unit k
$M_c(x, y)$	Class activation map of class c at location (x, y)
L_{cls}	Classification loss
L_{reg}	Regression loss
T	Temperature parameter of the softmax function
p_c	probability of a class c
$\mathcal{U}$	Uniform distribution
$\mathcal{B}$	Set of all bounding boxes in an image
C	Set of all classes corresponding to boxes $\mathcal{B}$
$\mathcal{L}_{unsup}$	Unsupervised loss
$\mathcal{L}_{sup}$	Supervised loss

INTRODUCTION

0.1 Object Localization

Object localization is the task of identifying and precisely locating the position of one or more objects within an image. The localization returns the bounding box coordinates (usually in terms of top-left and bottom-right corners) that tightly enclose the object(s) of interest. This is different from the widely studied computer vision task of image classification where we only need to identify the object classes present in the image. In object localization, we need to identify the "where" the predicted objects are present in the image. Figure 0.1 shows image understanding problems in computer vision in the increasing order of their complexity. Localization is the next complex task after classification and is an important component in the subsequent more complex tasks of object detection and instance segmentation.

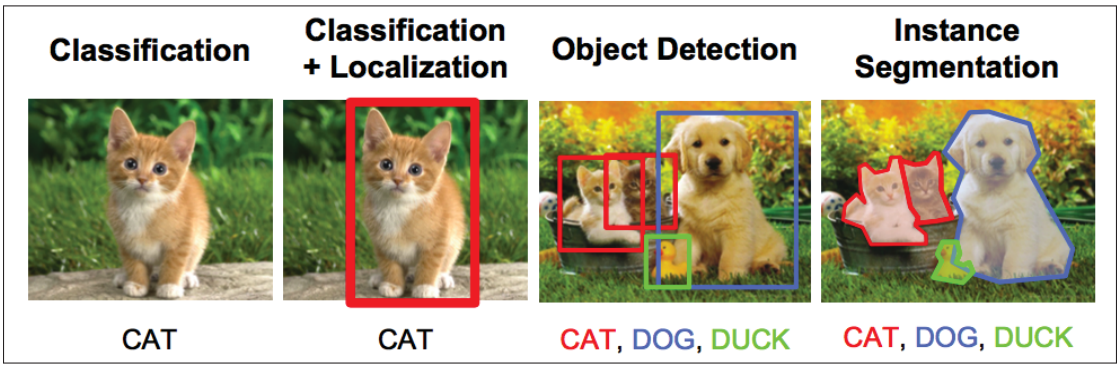


Figure 0.1 Computer vision problems in the increasing order of complexity. Taken from cs231n standford 2015

Generally, instance-level recognition tasks from an image are referred to as dense prediction problems. Both object detection and instance segmentation are dense prediction problems where localization is an integral component. In object detection tasks, localization is used to identify the location of multiple objects in the image. The detector uses a classification and localization head to predict multiple objects in the image. The classification head predicts the object category.

The localization head is the one that gives the required object localization. It is achieved by regressing the bounding box coordinates of the ground truth box given by a human annotator.

In addition to the classification and localization of bounding box regions, instance segmentation aims to predict pixel-level labels for each object inside the bounding box. This is also called pixel-level localization. The combination of accurate localization and pixel-wise segmentation enables instance segmentation models to provide detailed and fine-grained information about objects within an image. In addition to classification and localization heads, instance segmentation also uses a mask head to predict the pixel masks for each object. Given the localization from the localization head, the mask head makes pixel-level predictions for all pixels inside the box.

As localization is a vital component for the majority of computer vision tasks, we need better localization for a deeper understanding of the objects present in an image or scene. For example, in applications like autonomous driving, accurate object localization is vital for making safe and reliable decisions. In medical imaging, precise object localization is critical for identifying abnormalities, such as tumors or lesions. In retail environments, accurate localization of products on shelves or in storage helps in managing inventory. Next, we will discuss the challenges in learning better localization.

0.2 Challenges in Object Localization

Getting better localization is correlated with getting better representation or features from the image/object regions. The feature learning for localization has gone through a significant evolution, starting from using traditional features like SIFT (Lowe, 1999) and HOG (Dalal & Triggs, 2005a) to modern deep features with transformers (Vaswani *et al.*, 2017). A closely related evolution can be seen in the localization ability of object detectors from Dalal & Triggs (2005b); Felzenszwalb, Girshick, McAllester & Ramanan (2010) to Carion *et al.* (2020). When deep learning (DL models) entered the main-stream (Krizhevsky, Sutskever & Hinton, 2012), object

localization initially took a new shape where the deep features are used for feature engineering but the localization is predicted separately in a second stage after feature learning (Sermanet *et al.*, 2014; Girshick, Donahue, Darrell & Malik, 2014). Later end-to-end prediction of the localization became more popular with improved speed and accuracy (Ren, He, Girshick & Sun, 2015; Liu *et al.*, 2016). As the popular localization from 2015 onwards relies on DL models, the research in this thesis (started in 2018) is focused on deep object detectors. First, we will present the main challenges in the modern deep object localization methods which justifies this research.

0.2.1 Appearance variation

There is significant variance in appearance for objects of the same class in images collected from the real world. The appearance variation can be due to lighting (e.g., day and night images), camera angle, occlusion, clutter, and many others (Liu *et al.*, 2019). Variations can also arise due to temporal evolution, for example, wear and tear of the components in a machine. These variations in appearance also undermine the standard assumption of the deep models which states that the training and test data comes from the same distribution. This appearance variation also creates significant confusion between classes, especially in fine-grained detection problems like (Welinder *et al.*, 2010). Figure 0.2 shows a localization problem where two bird species are visually similar, in the CUB-200 dataset. More about appearance variation and the issues it creates when learning with inexact annotations is presented in chapter 2.

0.2.2 Scale variation

Variation in object scale is ubiquitous in localization. Object instances can exhibit significant variation in size. For example, the same objects from the aerial view appear smaller in size than the front view (e.g.: the difference in scale for the person and vehicle classes between the aerial dataset VisDrone (Zhu *et al.*, 2018) and natural images of Pascal VOC dataset (Everingham, Gool, Williams, Winn & Zisserman, 2010)). This can create confusion among classes, missing

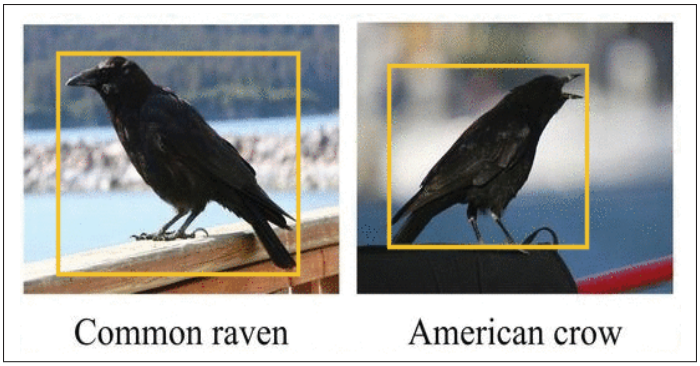


Figure 0.2 Fine-grained localization of bird species in the CUB-200-2011 dataset

detection due to low appearance information, and reduced confidence in detected objects (Oksuz, Cam, Kalkan & Akbas, 2021). Figure 0.3 shows an instance of scale variation where a reduction in object scale resulted in low confidence for the "kite" category and missing prediction for the "snowboard" category.

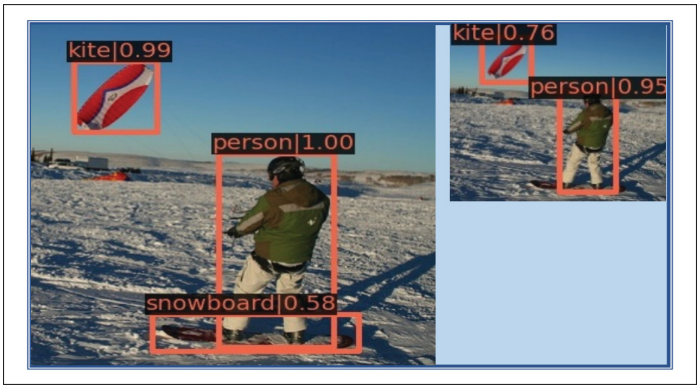


Figure 0.3 Scale inconsistent detection. Taken from Guo *et al.* (2022)

0.2.3 Annotation of labeled data

To train the localization head, we need the location information of all categories of interest present in the image either in the form of bounding boxes or pixel-wise masks. Unlike the image-level class labels, getting location annotation is costly and time-consuming (Liu *et al.*, 2021a; Xu *et al.*,

2021; Tang, Chen, Luo & Zhang, 2021; Li, Yuan & Li, 2022a; Bilen & Vedaldi, 2016; Tang, Wang, Bai & Liu, 2017; Li *et al.*, 2022b; Antonelli *et al.*, 2022; Köhler, Eisenbach & Gross, 2021). See figure 0.4 for a comparison of labeling time for different types of annotations. Thus getting sufficient annotated images to train the localization head is costly in practical applications.

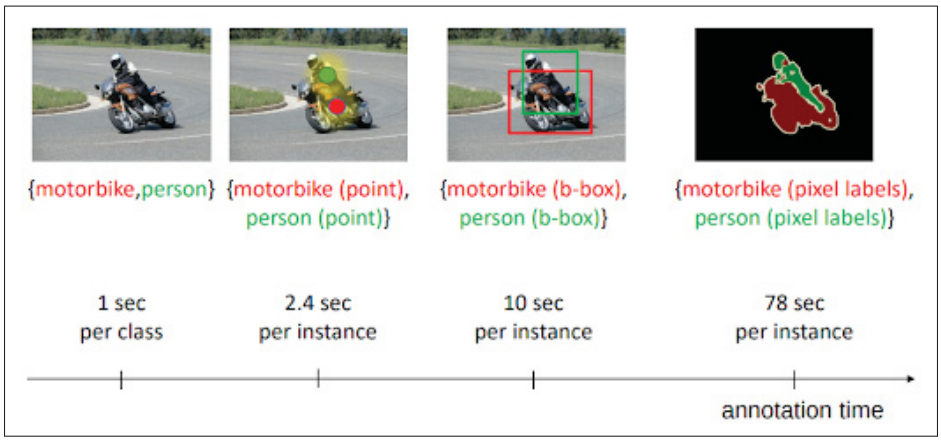


Figure 0.4 Annotation time for different types of annotations. Taken from Bearman *et al.* (2015)

0.2.4 Long-tail distribution of objects

Many object classes in the real world exhibit a long-tailed distribution. This creates more bias towards the prediction of the dominant classes (Oksuz *et al.*, 2021). Recently LVIS (Gupta, Dollar & Girshick, 2019) dataset was released to study the long-tailed object detection problem more effectively. The performance of state-of-the-art methods on balanced datasets showed significant deterioration in this dataset. The long-tail problem is undermining the classification abilities during recognition. Their impact is similar in classification and localization as well. The most visible impact of this challenge is in low confidence or no detection of the tail classes. Figure 0.5 shows the confidence distribution of object classes in the long-tail context.

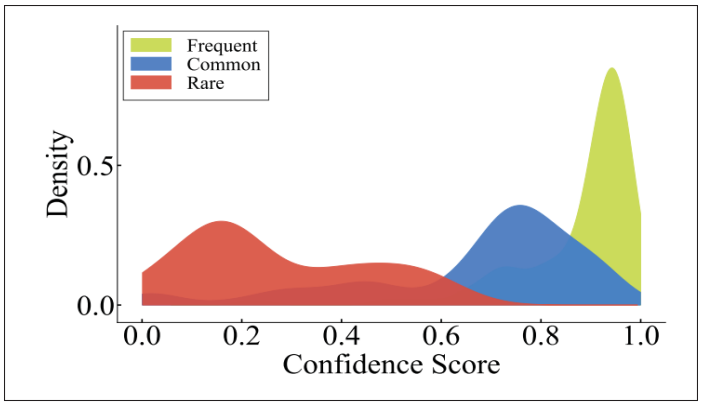


Figure 0.5 Long-tail distribution of objects classes show significant variations in prediction confidence of head and tail classes. Taken from Zang *et al.* (2023)

0.3 Annotation Challenge in Deep Localization Methods

Among the many challenges we discussed so far, the focus of this thesis is on the annotation challenge of modern localization methods. Deep detectors and localizers are inherently label-hungry. Their success recipe is in training with large annotated datasets like MS-COCO (Lin *et al.*, 2014) and OpenImages (Kuznetsova *et al.*, 2020). As deep learning-based methods are winning the margin significantly over traditional methods, the value of reducing annotation costs is increasing. Practical applications can collect thousands of raw images from their deployment, but the bottleneck in using these images is the lack of annotations. It is important to reduce the impact of this bottleneck through research for the increased adoption of deep learning-based localization techniques.

0.4 Research Gap

In the past years, we have seen immense progress in the localization accuracy of the methods learning to localize objects with reduced supervised data. Among the many methods for learning with reduced supervised data, we focus our attention on weakly supervised and semi-supervised

methods in this thesis. In weakly supervised methods, both WSOL (Weakly Supervised Object Localization) and WSOD (Weakly Supervised Object Detection) have benefited immensely from deep features to improve localization. In semi-supervised methods, better localization helped in pseudo-labeling the unlabeled data more accurately, thereby improving the detection performance. However, some important issues remain to be addressed.

WSOL focuses on single-object localization using image-level supervision. The default component of modern WSOL approaches in the recent literature is an architecture called Class Activation Maps (CAMs) (Zhou *et al.*, 2016). Other methods in the literature propose techniques on top of the CAM architecture to further improve the localization (Zhang *et al.*, 2018a; Singh & Lee, 2017; Choe & Shim, 2019; Choe. *et al.*, 2020). CAM is repurposing a classification network to do localization with restrictive architectural choices (a global average pooling and fully connected layers must be required at the end). The localization is derived from classification activation, not explicitly learned. During our studies in 2018, fully convolutional models were popular design choices due to increased computational efficiency and reduction in model parameters (Tian *et al.*, 2019; Long, Shelhamer & Darrell, 2015; Dai, Li, He & Sun, 2016). So the natural question was how can we design a WSOL system with a fully convolutional architecture where localization-specific parameters can be explicitly learned. We addressed this in our first contribution.

WSOD aims to detect multiple objects (even from different classes) in an image using image-level supervision. The first deep weakly supervised object detection method is proposed by Bilen & Vedaldi (2016) called Weakly Supervised Deep Detection Networks (WSDDN). Other methods in the literature are built on top of the WSDDN architecture to further improve the localization accuracy by solving mainly the discriminative region localization problem of weakly supervised methods (Diba, Sharma, Pazandeh, Pirsiavash & Gool, 2017; Tang *et al.*, 2017; Li, Huang, Li, Wang & Yang, 2016; Wan, Wei, Jiao, Han & Ye, 2018). Fundamentally, the WSDDN

architecture and its training process (using MIL pooling) significantly deviate from the fully supervised detectors. Due to this, advances in fully supervised detection research are not easy to adapt to WSODs. Can we solve the WSOD problem with existing fully supervised detectors? Bringing them in a common platform in terms of architecture and loss functions could make the techniques in one useful for the other. This is an important aspect that the community hasn't paid much attention to. Existing WSODs are mainly focusing on improving the WSDDN model in a race to close the performance gap with fully supervised detectors.

Semi-Supervised Object Detection (SSOD) aims to train a detector with few annotated images and a large collection of unlabeled images. Our focus on SSOD is to use them for tiny object detection in aerial images. Typically aerial image collection mechanisms are deployed in surveillance mode with satellites and drones frequently visiting the given location. Due to this, many images are being collected, but only a fraction of them is utilized for the standard fully supervised training as it is not possible to label this vast collection of images with tiny objects. Thus the potential of SSOD is significant in this area to better utilize the whole images collected. The most widely used SSOD strategy in recent literature is based on the mean-teacher method (Tarvainen & Valpola, 2018). While the vanilla mean-teacher-based STAC (Sohn *et al.*, 2020) detector has undergone significant changes in recent years, using them on tiny object localization in its current form is challenging. Aerial images are usually very high in pixel resolution and objects are tiny and often appear clustered. Due to these differences, even the popular fully supervised detectors cannot be used standalone to get good detection performance (Yang, Fan, Chu, Blasch & Ling, 2019; Duan, Wei, Zhang, Qu & Wang, 2021; Deng *et al.*, 2020; Yang, Huang & Wang, 2022). Additional modules to deal with the scale imbalance are often used and the images are processed as tiles. Using the standard mean-teacher-based detector alone, in this case, is not optimal. One could add additional modules to aid small object detection, but training them in the mean-teacher settings is not straightforward. First, the mean-teacher method needs pseudo labels to train. How can one compute pseudo-labels for the added modules?

The added modules are often trained before the detector training, but how can we incorporate this multi-stage training in mean-teacher settings which is also inherently multi-stage (with a supervised burn-up stage followed by mean-teacher training)? We hypothesize these challenges might be the reason why semi-supervised detectors are not popular in aerial images.

0.5 Contributions

The contributions of this thesis are focused on addressing the research gap discussed above. For the weakly supervised methods, the contributions are more on the architectural side innovations addressing some of the fundamental questions. This is orthogonal to existing research which is mostly focused on performance improvement. For the semi-supervised methods, our contribution is mainly focused on adapting the popular mean-teacher system to aerial images for localizing tiny objects in high-resolution images.

The main contributions are

- A fully convolutional design for weakly supervised object localization. Different from CAM-based WSOLs, localization is explicitly learned by applying spatial transform in a convolutional fashion. Additional regularizations are proposed to reduce the discriminative region localization problem and improve scale-specific localization in the right feature pyramid level.

  Related publication:

  Convolutional STN for weakly supervised object localization, in International Conference on Pattern Recognition (ICPR), 2020.
- A weakly supervised and semi-weakly supervised detector that can be trained with off-the-shelf detection architectures. No change in the architecture or loss function is needed, unlike existing weakly supervised detectors. It learns to use images with weak labels by sampling region proposals from right object locations as pseudo-ground-truth boxes.

Related publication:

Semi-Weakly Supervised Object Detection by Sampling Pseudo Ground-Truth Boxes, in International Joint Conference on Neural Networks (IJCNN), 2022.

- A density-guided cropping and semi-supervised detection method for aerial images. Different from existing approaches (that use additional density extraction modules), object regions with clustered small objects are cropped out from high-resolution images reusing the detector itself. Inference is performed on them after upscaling for better small object detection. This design is easy to adapt to a semi-supervised mean-teacher detector where we identify density crops on labeled and unlabeled images, and use them to augment the training set to further boost the performance.

  Related publication:

  Cascaded Zoom-in Detector for High-Resolution Aerial Images, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPRw), 2023.

  Density Crop-guided Semi-supervised Object Detection in Aerial Images, in IEEE Transactions on Geoscience and Remote Sensing (TGRS), 2023. Under review.

0.6 Thesis Organization

The organization of this thesis is as follows. In chapter 1, we present the settings of weakly supervised object localization, weakly supervised object detection, and semi-supervised object detection in detail along with other approaches for reducing the annotation cost. We will then dive into the challenges of learning better localization in these settings, and review the existing research works, and the limitations of mainstream methods. This discussion will shape the focus of our contribution. Then, we also discuss the core components of our proposed solutions and present their values in solving the localization challenges with reduced annotations.

In chapter 2, we present the first contribution, Convolutional STN for WSOL. First, we present the localization abilities of STN when used in classification settings. However, these results are

on synthetic images of MNIST with clear foreground-background separation. We present the issues when using STN for localization on natural images. With our Convolutional STN design, we addressed these issues and obtained impressive results on localization in CUB-200-2011, and ImageNet datasets.

In chapter 3, the sampling-based WSOD is presented which uses the existing fully supervised architectures and trains them in a weakly supervised fashion by sampling region proposals as pseudo-GT boxes. Their performance gap with WSODs based on classical WSDDN architecture is studied empirically on the Pascal VOC dataset. To address this, we focus on our method using fully supervised detectors for WSOD permits training with accurate ground-truth (GT) labels as well. So, we propose to use a small fraction of images with GT annotation which resulted in a significant performance boost beating existing WSODs and the vanilla mean-teacher detector STAC. We also present the difficulty of the object proposal sampling on localizing small objects.

In chapter 4, our focus is on tiny object detection with less supervised data. We present the challenges in designing semi-supervised detectors for tiny object detection. We will focus on using density crops for accelerating tiny object detection which identifies the clusters of small objects and performs a focus and detect operation on those cluster regions. We then introduce a method to do density crop-based detection for tiny images where density crops are identified by the detector itself. As we can do "focus and detect" with our detector itself, we use the resulting architecture and perform semi-supervised learning on them using the mean-teacher method.

Finally, we conclude the thesis with a discussion of the key findings and recommendations for future research in this direction.

CHAPTER 1

BACKGROUND

In this chapter, we will present the technical details of existing methods for weakly and semi-supervised object detection and localization. We will try to understand the challenges in the respective problem settings as well as with the existing solutions.

1.1 Weakly Supervised Object Localization

Weakly supervised object localization (WSOL) is a technique in computer vision that aims to train object localization models using a limited amount of labeled data, often with only image-level annotations instead of precise bounding box annotations. Unlike traditional object localization, where models are trained on images with accurate bounding box annotations, WSOL with weaker forms of supervision avoids the laborious process of creating bounding box annotations. Figure 1.1 shows a high-level view of a WSOL system.

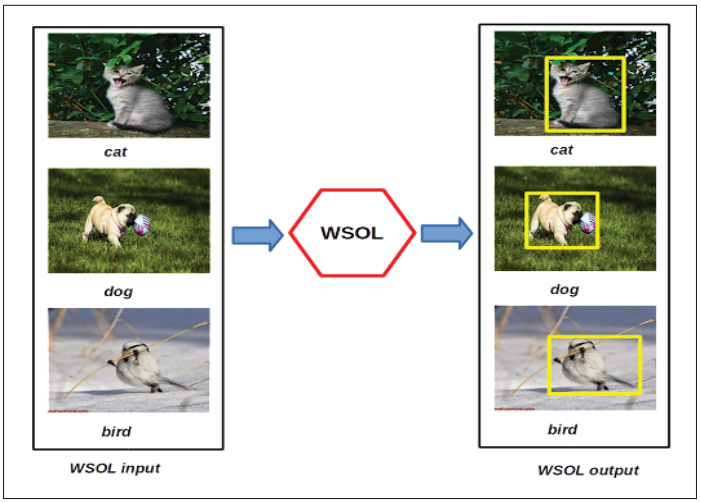


Figure 1.1 A high-level overview of the WSOL system using weak image-level labels for localization

Localizing objects with such a weaker form of supervision comes with several challenges. We will review the main challenges in WSOL next.

1.2 Challenges in Weakly Supervised Object Localization

In this section, we will discuss the challenges in weakly supervised object localization systems. As these challenges are fundamentally affecting the localization ability of the WSOL systems, recent research in WSOL is trying to address one or more of them.

1.2.1 Context misunderstood with object

This is one of the important problems faced by weakly supervised systems. As we don't have a bounding box to clearly specify the object boundaries, WSOL mostly misunderstands the object and its context. For example, consider the images of the bike class shown in figure 1.2. Bike images usually have a person as the rider, a road in the background, etc. Since we only give the supervision as the presence of a bike in the image, WSOL may incorrectly localize a "person+bike" as the bike category as both are present together in most training images.

Figure 1.2 Context of Motorbike: Road, Person

But, context information is not always detrimental to building weakly supervised detection systems. Kantorov, Oquab, Cho & Laptev (2016) utilized context information in a constructive manner to improve localization. They proposed to use additive and contrastive context descriptors to the region features to achieve this. The additive model encourages the predicted object region to be supported by its context region. The contrastive model encourages the predicted regions to be outstanding from their context region.

1.2.2 Selection of the most discriminative object regions

This is another common issue with weakly supervised systems. Since we optimize a classification objective, the weights learned will be strongly correlated to object parts to facilitate easier discrimination of the classes. As the training objective has no constraint regarding localization, the classifier will simplify its job by tuning its weights to recognize the discriminative object parts shown in figure 1.3. In WSOL this will result in the obtained segmentation mask around discriminative object parts. Whereas in WSOD, object proposals from discriminative regions will be returned as the detection output.

Figure 1.3 WSOL localizing discriminative object parts of the objects instead of the correct localization

Methods like ACOL (Zhang *et al.*, 2018a) and ADL (Choe & Shim, 2019) are addressing this limitation of the CAM method (Zhou *et al.*, 2016) widely used in the community. The studies from these papers observed that adapting the basic CAM with localization-specific learnable components improves the discriminative region localization of WSOL. We take inspiration from this in our CSTN.

1.2.3 Intra-class variation challenge in localization

Intra-class variations are a problem in both fully supervised and weakly supervised settings. However, the impact is more severe in the case of weakly supervised detectors, since we are not giving exact object location to learn the common features. For example, consider the variations

of the class "motorbike" shown in figure 1.4. Regardless of the scale, appearance, occlusion, aspect ratio, background clutter, etc., WSOL methods are expected to localize them correctly. In practice, it is observed that they localize very poorly under such variations.

Figure 1.4 Intra-class variations for the category motorbike

When enclosing object instances that contain a lot of background regions in a tightly fitting box, the localization goes to discriminative regions as the appearance model is consistent in such regions regardless of the variations of the object as a whole. This might be the case with poor localization on the "person" class in most of WSOL methodss. The "person" class has so many variations in terms of clothing, color, context, etc. The face is relatively easy to localize as the appearance model of a face will have fewer variations.

1.3 Standard Weakly Supervised Object Localization Approach

Class activation Mapping (CAM) is the popular method for single object localization with weak image labels. Proposed by Zhou *et al.* (2016) in 2016, CAM offers a simple method to extract activation maps from the classifiers. Other methods in the literature like ACoL (Zhang *et al.*, 2018a), ADL (Choe & Shim, 2019), HaS (Singh & Lee, 2017) and SPG (Zhang, Wei, Kang, Yang & Huang, 2018b) are improvements on the basic CAM to produce better localization. Figure 1.5 shows the standard CAM architecture.

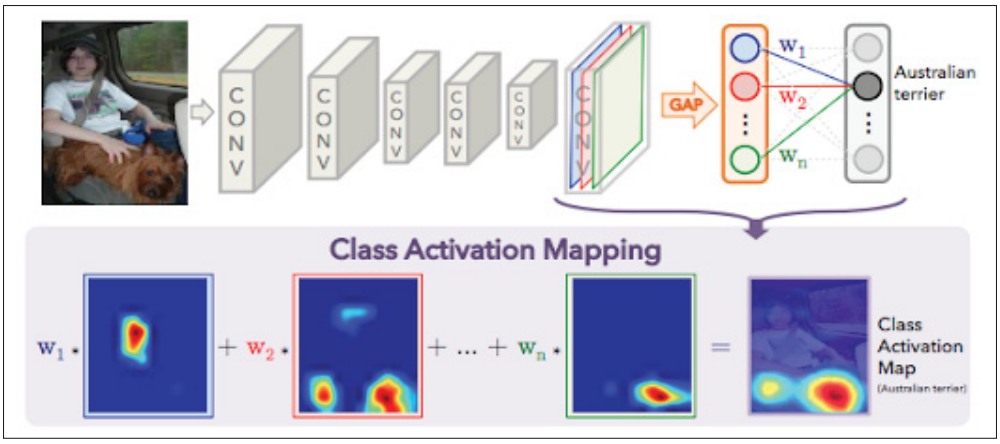


Figure 1.5 Architecture of CAM, the baseline system for WSOL. Taken from Zhou *et al.* (2016)

It consists of the feature extraction layers from a CNN backbone, followed by a Global Average Pooling (GAP) layer and a fully connected layer that gives the final class predictions. CAM uses a weighted sum of the final feature map to generate a saliency map from a standard classification network. Here the weights are basically the coefficient of the predicted class from the last fully connected layer after GAP. Let $f_k(x, y)$ represent the activation of a unit k at spatial location (x, y) in the last convolutional feature map. The result of performing GAP on unit k will be

$$F_k = \sum_{x,y} f_k(x, y) \tag{1.1}$$

The input to the softmax layer at the end for a class c is computed as

$$S_c = \sum_k w_k^c F_k \tag{1.2}$$

where w_k^c is the weight corresponding to class c for unit k. Plugging in the expression of F_k in S_c

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y) \tag{1.3}$$

Let's define $M_c(x, y)$ as the class activation map for class c at spatial location (x, y).

$$M_c(x, y) = \sum_k w_k^c f_k(x, y) \tag{1.4}$$

We get the activation map for the class c in M_c. This weighted saliency map is then up-scaled to image resolution and binarized by thresholding to obtain a mask. The box enclosing the max-connected component from the obtained mask is returned as the object location. The localization issues faced by the base CAM stem from its lack of position awareness, strong activation from small parts of the object while maximizing the classification score, and the bilinear interpolation of the saliency map.

Since CAM's training objective is primarily focused on achieving a high level of classification accuracy, its localization tends to correspond with the most discriminative object region. Figure 1.6 illustrates this problem and the solution proposed by Zhang *et al.* (2018a). Most of the recent WSOL techniques propose updated versions of the CAM that can avoid the bias towards the discriminative region (Singh & Lee, 2017; Zhang *et al.*, 2018a,b). They typically seek to erase or hide the most discriminative region during training so that the classifier will focus on other relevant object regions. To achieve this, they leverage different strategies, like using multiple classifiers to localize complementary regions(ACoL) (Zhang *et al.*, 2018a), self-produced guidance(SPG) (Zhang *et al.*, 2018b), randomly hiding patches from the input image (HaS) (Singh & Lee, 2017).

The different strategies for erase and learn are summarized and their issues are analyzed in detail by Mai *et al.* (2020). They summarized erase strategies as random erase, step-wise erase, and multi-branch erase. Figure 1.7 illustrates these strategies. While erase and learn was the fundamental strategy to deal with the discriminative localization problem in WSOL, we took a

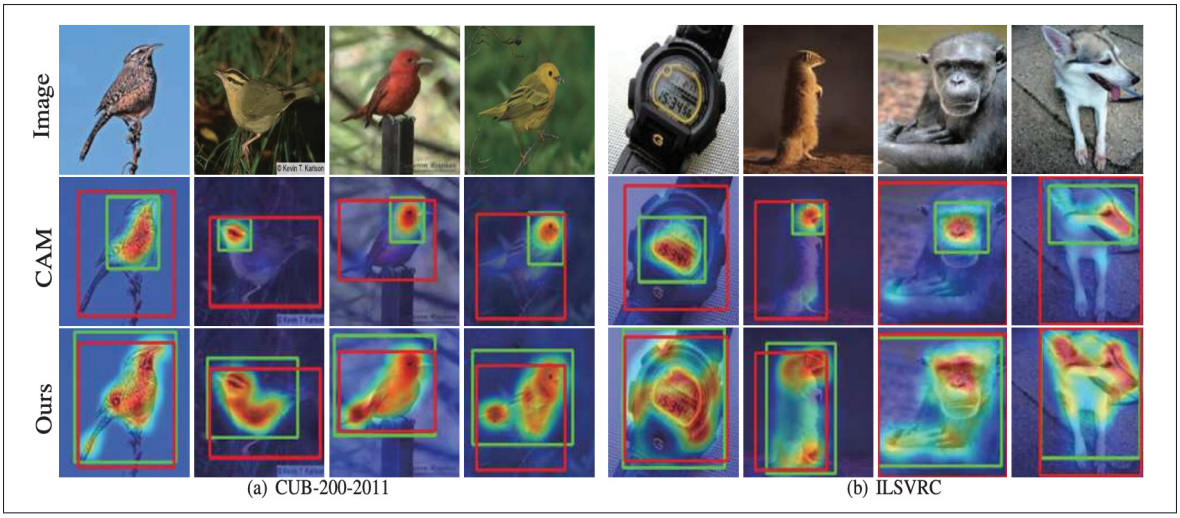


Figure 1.6 CAM localizing discriminative object parts. Taken from Zhang *et al.* (2018a)

different route and learned the localization with respect to reference boxes to solve this problem. This is in practice similar to learning with respect to anchor boxes in fully supervised detectors (Ren *et al.*, 2015). As the localized boxes are a small perturbation from the reference boxes of the matching level of the feature pyramid, we could avoid the large shrinking of the localization to discriminative areas. Basic CAM has no component for learning better localization, it is just a simple classification architecture that gives an activation map as a bi-product. Erase and learn methods are fundamentally adding more components to learn better localization. We also used learnable localization, but instead of following the erase and learn strategy, we used spatial transformers (Jaderberg *et al.*, 2015) for learning localization.

1.4 Spatial Transformer Networks for Localization

We used a Spatial Transformer Network (STN) for constructing a learnable localization component for our WSOL algorithm. An STN (Jaderberg *et al.*, 2015) is a learnable module that can be placed at any layer(s) of a CNN. It learns an affine transformation (other transformations are also possible) of its input to maximize the learning objective of the network (initially the authors proposed to maximize the classification accuracy, but later several use cases are

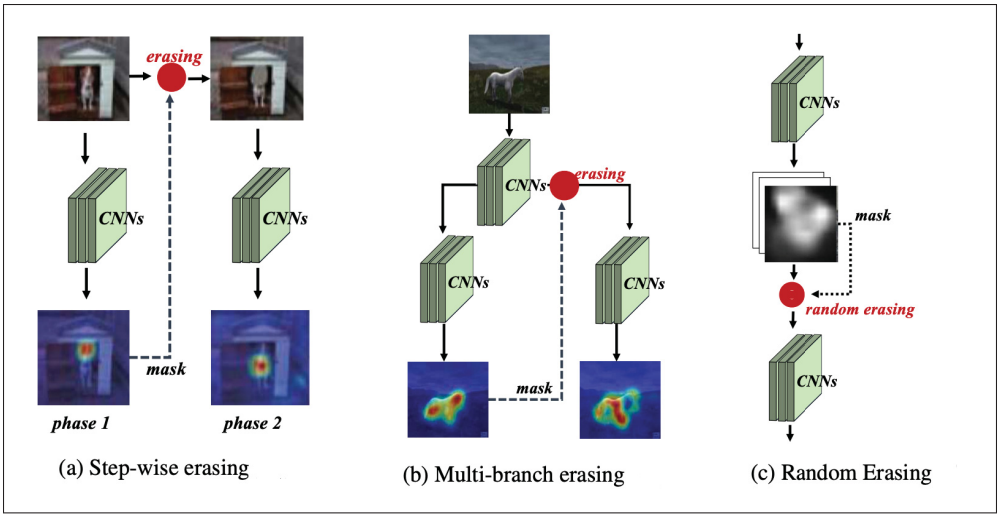


Figure 1.7 Different strategies for erase and learn. Taken from Mai *et al.* (2020)

being discovered like in image captioning (Johnson, Karpathy & Fei-Fei, 2016), disentangled representation learning (Detlefsen & Hauberg, 2019) and image composting (Lin, Yumber, Wang, Shechtman & Lucey, 2018) etc). STN applies a spatial transformation to its input feature map in the forward pass where the transformation magnitude is conditioned on the same input itself. STN is used in learning the localization within our weakly supervised localization system. Figure 1.8 illustrates the architecture of an STN block.

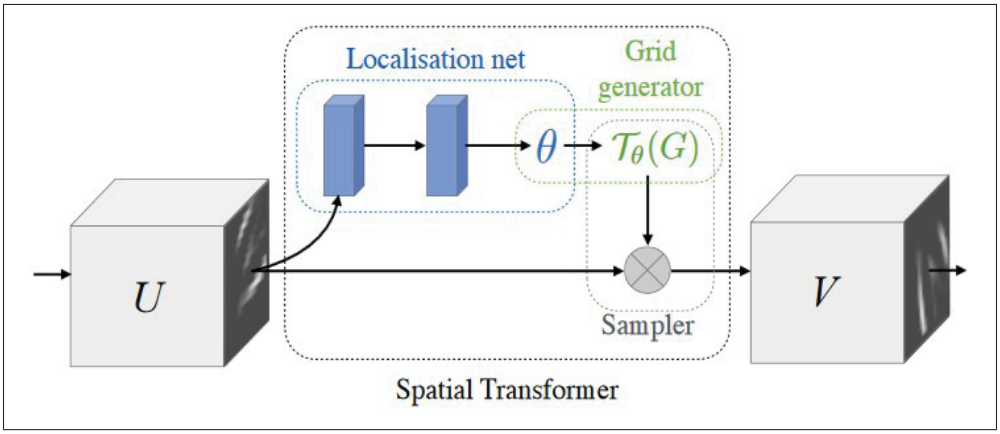


Figure 1.8 A spatial transformer module. Taken from Jaderberg *et al.* (2015)

The input feature map U is passed through the Localization net which predicts the transformation parameter θ. The transformation $T_\theta(G)$ is then applied to the regular sampling grid G. The resulting positions are sampled by the sampler (using bilinear interpolation) giving the final feature map V. The transformation typically used is an affine transformation, so the θ vector is 6-dimensional. The localization network can be fully connected or convolutional with the only restriction of predicting the right size θ at the end. Let (x_u, y_u) be a position in the input feature map U. Considering an affine transformation by T_θ the resulting sampling location will be obtained as

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} \tag{1.5}$$

The sampling kernel is then applied at each $\begin{pmatrix} x_s \\ y_s \end{pmatrix}$ to get the value at a corresponding pixel location in V. A bilinear sampling kernel is commonly used.

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c max(0, 1 - |x_i^s - m|) max(0, 1 - |y_i^s - n|) \tag{1.6}$$

where V_i^c is the output value computed for pixel (x_i, y_i) in channel c.

1.5 Supervised Object Detection

Fully supervised object detection needs bounding box annotations for each object present in the image. Fully supervised object detectors are broadly classified into two: one-stage (Redmon & Farhadi, 2017; Liu *et al.*, 2016) and two-stage (Ren *et al.*, 2015; Lin *et al.*, 2017b) detectors. Two-stage object detectors have a first stage that extracts RoIs (candidate object regions) whose reliability as a potential candidate region is quantified by their objectness score. Earlier approaches extract these RoIs using low-level image features in R-CNN (Girshick *et al.*,

2014), Fast R-CNN (Girshick, 2015), etc. Later, end-to-end two-stage models emerged as more accurate detectors, where an additional learnable head called RPN (Region Proposal Network) is used to regress candidate regions (Ren *et al.*, 2015; Lin *et al.*, 2017b; Dai *et al.*, 2016). In contrast, one-stage object detectors avoid the RoI extraction stage and classify and regress directly from each location in the feature map. They are generally fast and applicable to real-time object detection (Redmon & Farhadi, 2017; Redmon, Divvala, Girshick & Farhadi, 2016; Liu *et al.*, 2016). Though the two-stage detectors are typically slower compared to their one-stage counterparts, extracting reliable candidate regions in the first stage provides an edge in terms of localization accuracy.

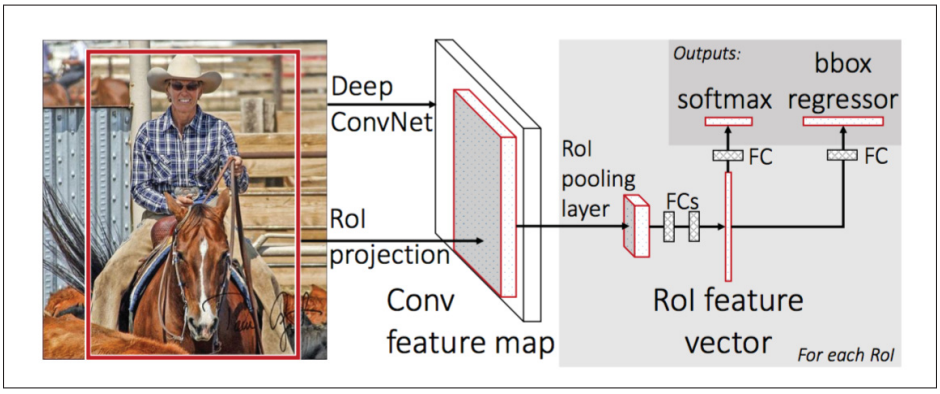


Figure 1.9 Components of a typical two-stage object detector.
Taken from Girshick (2015).

We will now review both types of architecture in detail. Figure 1.9 shows a high-level abstraction of a two-stage fully supervised object detector. It consists of a feature extraction module that extracts whole image features, RoI feature extraction which gives region-level features, the heads for classification, and bounding box localization. In the above example, a Deep ConvNet is giving the image-level features. Region-level features are obtained by projecting RoIs from the image space to the feature map and then performing RoI pooling. The classification head is implemented by a softmax layer; the localization head is implemented by a bounding box regressor. The regressor will predict the four corners of the bounding box enclosing the object.

Let the predictions for each RoI r be (p_r, b_r) where p_r is a softmax vector over the number of classes and b_r represents the top left and bottom right coordinates of the predicted box. During training, the predictions are matched with the given ground truth, and regions are labeled as foreground or background based on their overlap with any of the ground-truth boxes. If the overlap is above a threshold τ the region is designated as foreground. Its label is assigned to be the label of the corresponding ground-truth box. Let the GT be denoted as (u, v) where u contains the list of labels and v contains the list of GT box coordinates. The multi-task loss function is then computed as

$$L(p, b, u, v) = L_{cls}(p, u) + \lambda[o \geq \tau]L_{reg}(b, v) \tag{1.7}$$

Where $L_{cls}(p, u) = -log p_u$ can be, for example, the log loss for true class u. $L_{reg}(b, v)$ can be L_1 loss, L_2 loss or a combination of them (Girshick, 2015). There are more variants for L_{cls} and L_{reg} in modern object detectors (Lin, Goyal, Girshick, He & Dollar, 2017a; Tian *et al.*, 2019).

In one-stage detectors, rather than using RoIs, predictions are made from every point or grid of points in a feature map. Both the classification and localization heads make a prediction at each point. During training, they are matched with the available GT, and classification and localization losses are computed as explained before. Instead of a curated set of predictions from confident RoIs, dense predictions are made at every point in one-stage detectors, thus creating many noisy predictions. Post-processing (like NMS) then removes the noisy predictions giving the final output. Figure 1.10 illustrates a typical one-stage detector architecture.

Among the one-stage detectors, a popular category at the current time is anchor-free detection. Popularized by the Faster RCNN paper (Ren *et al.*, 2015), anchor boxes are pre-defined template boxes of various scale and aspect ratios. They are placed at every point on the feature map. The predictions are made as offset to these anchor boxes as shown in figure 1.11 left so that the gradient-based methods can easily learn these small offset values instead of direct box coordinate prediction. During training, anchors are matched to the GT boxes and they are labeled

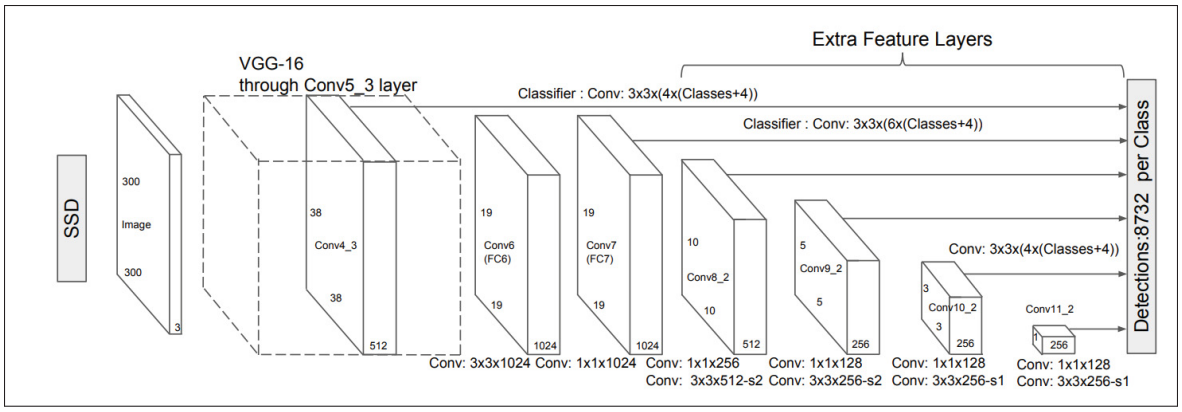


Figure 1.10 Components of a typical one-stage object detector. Taken from Liu *et al.* (2016)

as foreground and background. The problem with using anchor boxes is their size must be hand-picked carefully to match the average object size range in the dataset (Redmon & Farhadi, 2017) for the best results. Anchor-free detectors like FCOS (Tian *et al.*, 2019) and CenterNet (Duan *et al.*, 2019) are avoiding this manual process and making the training pipeline more automated. FCOS for example, is predicting the offset to top, bottom, left, and right ends from each pixel position in the feature map as shown in figure 1.11 right.

Another category of detectors getting popular these days is one that models object detection as a set prediction problem. DETR (Carion *et al.*, 2020) is the pioneer in this line of work with a transformer (Vaswani *et al.*, 2017) based backbone. An encoder-decoder architecture makes set predictions, they are then matched with available GT using the Hungarian bipartite matching algorithm. One of the main attractions of the set prediction design is they don't need post-processing like NMS. So the predictions can be directly used without additional processing.

1.6 Reducing the Annotation Cost in Object Detection

To reduce the annotation burden, researchers explored multiple research directions including semi-supervised detection (Sohn *et al.*, 2020; Liu *et al.*, 2021a), weakly supervised object detection (Cinbis *et al.*, 2016; Bilen & Vedaldi, 2016), semi-weakly supervised object detection (Fang *et al.*, 2021; Chen, Yang, Zhang, Zhang & Sun, 2021), Few-shot Object Detection

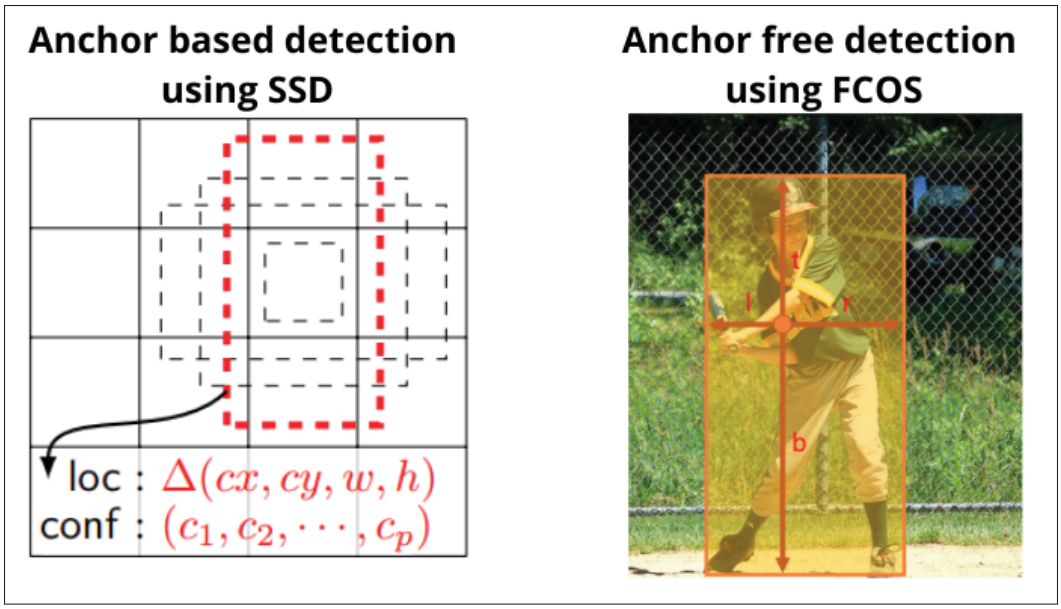


Figure 1.11 Anchor-based vs anchor-free prediction from a point in the feature map. Taken from Tian *et al.* (2019)

(Antonelli *et al.*, 2022), Domain adaptive object Detection (Li *et al.*, 2022b) among many others. We will now briefly review these research directions.

1.6.1 Semi-supervised object detection

Semi-supervised object detection (SSOD) aims to train a detector with limited labeled data and a large collection of unlabeled data (Liu *et al.*, 2021a; Guo *et al.*, 2022; Tang *et al.*, 2021). Let D_l and D_u denote the labeled and unlabeled data respectively. In a semi-supervised setting, we have $D_u >> D_l$ and there is a closed set assumption that states that the unlabeled data contains the same object classes as labeled. This is assumed because the data collection process is the same for all data points and annotations are created only for a small fraction of the total data collected. To learn the detector in a semi-supervised setting, the most popular approaches are consistency regularization and/or pseudo labels (Sohn *et al.*, 2020). Consistency regularization works by enforcing consistent predictions between differently augmented versions of unlabeled input images (Jeong, Lee, Kim & Kwak, 2019). The pseudo-labeling approach works by computing pseudo labels for the unlabeled images (Xu *et al.*, 2021).

1.6.2 Weakly supervised object detection

Weakly supervised object detection (WSOD) aims to train an object detector with classification data where image-level annotations are only available. As we observed the annotation time in figure 0.4, image-level labels are easier to provide than instance-level labels. Thus in the real world, we see large annotated datasets for classification (e.g., ImageNet (Russakovsky *et al.*, 2015)) than for detection ((Lin *et al.*, 2014; Everingham *et al.*, 2010)). What if we can learn object detectors from this wealth of classification data? WSOD tries to address this research question. Typically a weakly supervised detector is trained following the Multiple Instance Learning (MIL) approach (Zhang, Han, Cheng & Yang, 2021). To do that, we first obtain object/region proposals (probable candidate boxes) from the input image. Then features are extracted from these region proposals and their softmax class probabilities are computed. A MIL pooling operation then aggregates these region-level class probabilities and produces image-level class probabilities. It is then used to train the detector using the weak image labels available.

1.6.3 Semi-weakly supervised object detection

Semi-weakly supervised detection(Semi-WSOD) aims to combine the best of both worlds. Using the limited available labeled data, it can learn better localization cues over a weakly supervised detector. Whereas, by having weak image-level annotations on the unlabeled data, candidate object regions can be pseudo-labeled more accurately, consistent with the image-level labels (Chen *et al.*, 2021; Meethal, Pedersoli, Zhu, Romero & Granger, 2022; Fang *et al.*, 2021). Some works considered point-level annotations in addition to the image-level labels semi-weakly supervised setting (Chen *et al.*, 2021).

1.6.4 Few-shot object detection

Few-shot object detection aims to detect novel object categories with only a few annotated instances of them (Köhler *et al.*, 2021). It assumes access to abundant labeled source data with bounding box annotations. This source data is used to learn generic object characteristics and

when a novel class is presented with a few annotations, the classification heads learn to identify them. In this setting, when we say K-shot object detection, we have exactly K annotated instances per novel category. Recently, with powerful large vision-language models like CLIP (Radford *et al.*, 2021), localizing novel objects with zero instances for training (Zero-shot Detection) is gaining attention (Zhong *et al.*, 2022).

1.6.5 Domain adaptive object detection

Domain adaptive object detection aims to learn an object detector with labeled source domain and unlabeled target domain (Li *et al.*, 2022b). This is the most popular type of domain adaptation, also referred to as unsupervised domain adaptation (UDA). There are more variations where weak labels or full labels are available for the target domain, but UDA is widely used as we have zero annotation cost for the target domain in that setting. As we have seen before, appearance variation results affect the generalization ability of deep detectors. Domain adaptation methods explicitly train the detector to improve the generalization to a new domain. In such cases, the knowledge about object classes learned from the source domain can be transferred to facilitate better detection of the same objects in the target domain. Recently, the mean-teacher combined with adversarial training has emerged as a successful recipe in domain adaptive object detection (Li *et al.*, 2022b).

Among the many possible research directions discussed so far to reduce the annotation cost, the focus of this thesis is on semi-supervised and weakly supervised learning techniques. We chose this setting keeping the reusability of the existing data collections in mind. To train weakly supervised detectors, the classification data available in plenty can be reused. To train in semi-supervised detectors, the raw data collected from the practical deployment can be reused. Generally from the data collected, only a subset is annotated and the remaining is ignored. However, the semi-supervised detectors can use both sets.

1.7 Weakly Supervised Object Detection

Figure 1.12 illustrates the problem settings of Weakly Supervised Object Detection (WSOD). The data is provided with weak image-level labels as supervision. If we can train the detector with image-level labels, the vast collection of classification datasets available will be useful for building object detectors. Apart from the image labels, point annotations and scribble annotations are also considered as weak annotations (Chen *et al.*, 2021).

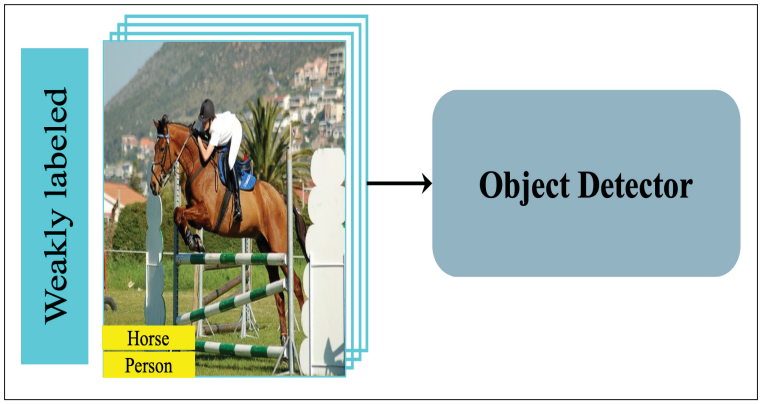


Figure 1.12 Weakly supervised object detection problem settings

While it is appealing to train object detectors with classification data, using this inexact supervision comes with many challenges. Next, we will discuss those challenges.

1.8 Challenges in Weakly Supervised Object Detection

The fundamental challenges of discriminative region selection and context misunderstanding due to inexact supervision remain the same here as well. This is similar to WSOL, it stems from using only image-level labels for object localization. In addition to this, there are additional challenges in WSOD compared to WSOL.

1.8.1 Problems with multiple instances

When there are multiple instances of the same class in close proximity, WSD tends to localize all such instances together as one instance of that class. See figure 1.13 for some examples. This is due to a strong feature response from that region in the presence of multiple objects. In a fully supervised case, this can be avoided reasonably well due to the separate ground truth provided for each instance.

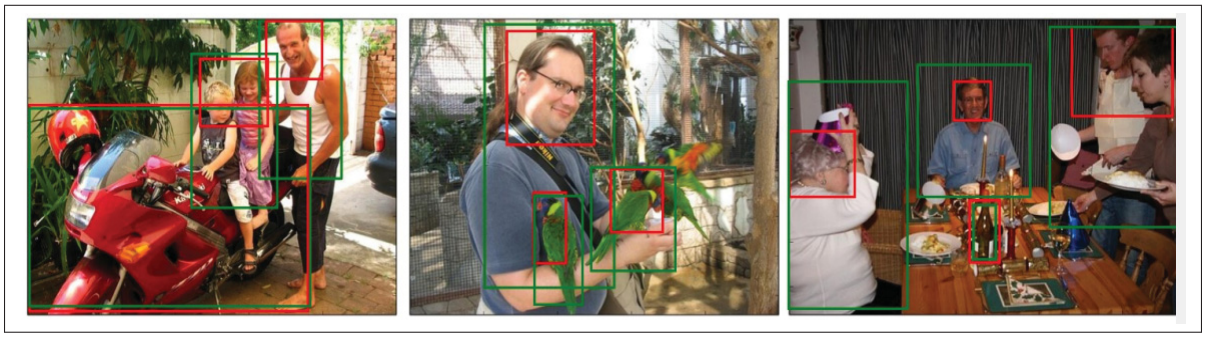


Figure 1.13 Multiple instances of "person" and "bird" class are localized as a single object. Taken from Zhang *et al.* (2018c). Here the red boxes are generated from WSDDN. The green ones are obtained after the accurate proposal mining techniques proposed by them

1.8.2 Slow inference

The detectors in the weakly supervised category are much slower compared to their fully supervised counterparts. The fast fully supervised detectors are single-shot ones which don't use object proposals. The region proposal stage consumes at least 250 ms (Shen, Ji, Zhang, Zuo & Wang, 2018). Since a majority of the weakly supervised detectors are based on region proposals, this delay is unavoidable. The only method which reported the inference time is (Shen *et al.*, 2018). Their framework obtained an inference speed of 118 FPS based on SSD300 (Liu *et al.*, 2016) architecture and 50 FPS on SSD500. This is simply by retraining an SSD detector with WSDDN detection results as pseudo ground truth. The detection speed of other systems as measured by Shen *et al.* (2018) under the same settings is given below in table 1.1.

Table 1.1 Detection speed of different weakly supervised detectors. Here, except W2F, all other methods are reported from the experiments on the same settings (GTX 1080Ti GPU with cuDNN v6 on Intel i7-6900K@3.20GHz). The result of W2F is from an independent experiment (on Pascal TITAN X)

Method	FPS
WSDDN (Bilen & Vedaldi, 2016)	0.27
WSD with progressive domain adaptation (Li *et al.*, 2016)	2.11
ContextLocNet (Kantorov *et al.*, 2016)	0.38
OICR (Tang *et al.*, 2017)	0.28
Self-taught WSD (Jie, Wei, Jin, Feng & Liu, 2017)	1.73
W2F (Zhang *et al.*, 2018c)	1.25
GAL FWSD (Shen *et al.*, 2018)	118

1.8.3 Localization getting stuck in poor local optima

This is a problem faced by methods popular in the pre-deep learning era. The standard training process of a weakly supervised detector then starts with an initial set of proposed object regions. Then a detector is trained on these proposals which can be used to score regions. With that, we'll choose better region proposals in the next re-location stage and re-train the detector. During the learning process, this alternating re-localization and re-training process continues. There are several issues we face with such a learning process. One case is that the detector may get stuck at the initial regions proposed and hardly move during the subsequent re-localization steps (this problem is generally called degenerate re-localization (Cinbis *et al.*, 2016)). Figure 1.14 shows a case in which multiple objects of the bicycle category are localized due to bad initialization. We can see that the localization got stuck in the initial window and hardly moved from there in successive iterations.

The problem of localization getting stuck in the initialization was observed by Cinbis *et al.* (2016). The high dimensionality of the feature vectors and alternating optimization are identified as the main factors responsible for this. They proposed a multi-fold training strategy to overcome this. The object detectors are trained on all folds except a held-out one and re-localized on the held-out fold. This helps to avoid the bias introduced when training and re-localizing on the

Figure 1.14 Re-localization gets stuck in the initial proposal. Taken from Cinbis *et al.* (2016)

same set of images. This doesn't occur in modern WSOD like WSDDN (Bilen & Vedaldi, 2016) and its several variants since they don't perform this alternating re-localization and re-training.

1.9 Standard Weakly Supervised Object Detection Approach

Now we will try to understand the design of existing weakly supervised detectors and their shortcomings. Figure 1.15 presents an abstract architecture of the popular WSOD systems.

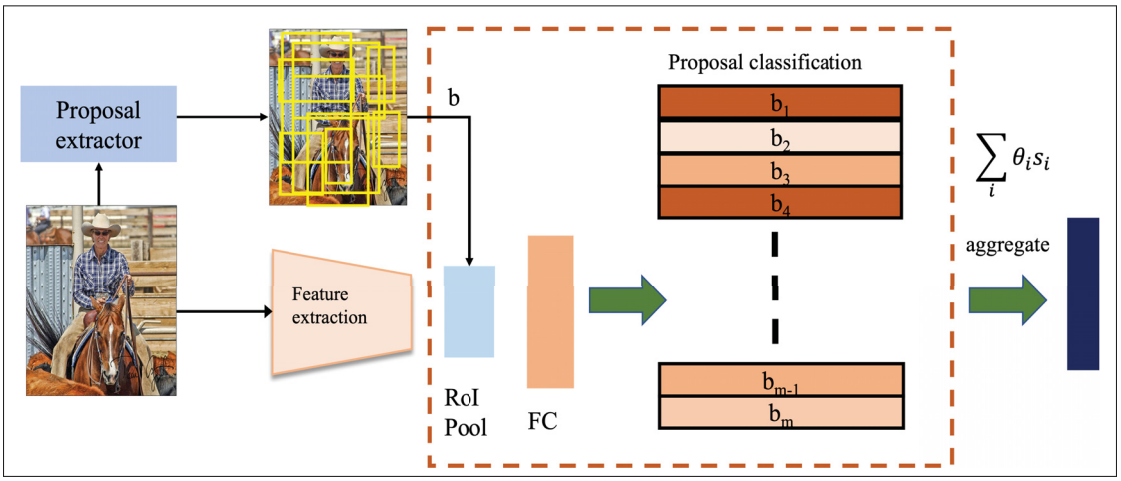


Figure 1.15 Typical WSOD architecture

It consists of three main components. A proposal extractor, a feature extractor, and an aggregation operation to combine proposal classification scores to image classification scores. Object proposals are candidate regions in an image with a high chance of an object being present. It has distinctive characteristics from background regions including a well-defined

closed boundary in space, a different appearance from its surroundings, and sometimes it is unique within the image and stands out as salient (Alexe, Deselaers & Ferrari, 2010). Typically in weakly supervised detectors, we compute the object proposals in the training images apriori (Bilen & Vedaldi, 2016; Diba *et al.*, 2017). Commonly used techniques for proposal extraction are selective search (van de Sande, Uijlings, Gevers & Smeulders, 2011) and edge box (Zitnick & Dollar, 2014).

The feature extractor extracts features from the whole image. Typically a ConvNet is used in modern deep weakly supervised detectors for this (Zhang *et al.*, 2021; Bilen & Vedaldi, 2016). Next, we need to compute region-level features to understand the class of objects present in a region. RoI pooling operation does this task by pooling features from each object proposal extracted from the image. The advantage of this architecture is that the expensive image-level feature computation step needs to be performed only once. The features for each candidate proposal can be extracted by the RoI Pooler with an inexpensive computation (Girshick, 2015). The region-level computations are highlighted by a dashed box in the figure. Once we have the RoI features, it is passed through subsequent fully connected layers giving region-level classification scores.

Next, we need to perform an aggregation operation to compute the image-level classification score from the region-level scores. This is typically done by a MIL (Multiple Instance Learning) pooling operation. Suppose there are N regions proposals in an image. Let us denote by s_i the score of the i^{th} region proposal which is a vector of length C where C is the number of classes in the dataset. The MIL pooling operation computes the weights θ_i for each region proposal and performs the aggregation to get the image-level score as I_s

$$I_s = \sum_{i=1}^{N} \theta_i s_i \tag{1.8}$$

I_s is also a vector of length C. It can be then used to compute the image classification loss as we have class labels for the image. Typically a binary cross-entropy function is used to compute

the loss. Let the scores of the class c in the i^{th} region proposal p_i is $s_i^c = f_c(p_i, I)$ where f is the RoI pooling operation. The weights θ_i associated with the region proposal are computed as normalized scores:

$$\theta_i = \frac{\exp\{\frac{s_i^c}{T}\}}{\sum_j \exp\{\frac{s_j^c}{T}\}} \tag{1.9}$$

with T being the temperature parameter that defines the sharpness of the weight distribution and is a hyper-parameter of the learning approach. In this way, boxes with higher scores will have more impact on the learning and the learning will focus more and more on the locations of the image that are more likely to contain the object of interest. At inference, those regions can be extracted to get the bounding box locations of the corresponding object. While this formulation works well, it is computationally expensive because it has to evaluate at each training iteration and for each image all box locations p_i. This architecture is first proposed by Bilen & Vedaldi (2016) (referred to as WSDDN) and further popularized with additional components for better instance localization by Diba *et al.* (2017); Tang *et al.* (2017); Zhang *et al.* (2018c) etc. These methods are in general finding solutions for the challenges faced by the WSDDN model. Next, we will try to understand the main challenges in the WSOD.

1.10 Importance Sampling for Weakly Supervised Object Detection

We used importance sampling in the design of our weakly supervised object detection. Important sampling gives us a way to approximate quantities from a distribution when we cannot sample from that particular distribution and compute a Monte Carlo estimate of the quantity. We used importance sampling in our sampling-based pseudo-label mining for weakly supervised object detection. With importance sampling, we computed the expected score of regions by sampling from an alternate distribution of scores. Suppose we have a distribution $p(x)$ and we are interested in calculating the expected value of the random variable x from this distribution. If we can generate samples from $p(x)$, we can simply use the Monte Carlo estimation to get an

estimate of the expectation as follows:

$$E(x) = \int xp(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.10}$$

where n is the number of samples generated. The difficulty is when sampling from $p(x)$ is difficult. In this case, we can use importance sampling and compute the expected value by using an alternate distribution from which sampling is easy. Suppose $q(x)$ is the alternate distribution, the expectation then is equivalent to

$$\begin{aligned} E(x) &= \int xp(x)dx \\ &= \int x\frac{p(x)}{q(x)}q(x)dx \\ &\approx \frac{1}{n}\sum_{i\sim q}\beta_i x_i \end{aligned} \tag{1.11}$$

where $\beta_i = \frac{p(x_i)}{q(x_i)}$ is the ratio of densities. The important thing is the x's are now sampled from the q distribution and we only need the ratio of these densities to offset for the correction due to sampling from a different distribution. The q distribution is usually a simple one from which sampling is easy. Figure 1.16 illustrates a simple example where importance sampling helps in estimation. We have a Gaussian distribution (red curve) with parameters $N(\mu = 0, \sigma = 1.5)$. Suppose we are interested in estimating the tail probability at the shaded regions. Sampling from the red distribution $p(x)$ in this case hardly fetches samples from the tail region, so our direct Monte Carlo estimation will be incorrect. Otherwise, we need a prohibitively huge number of samples to estimate it correctly. Now let's consider another Gaussian distribution (green curve) with parameters $N(\mu = 2.5, \sigma = 1.5)$. Sampling from this distribution q has a higher chance of fetching samples from the required tail region of p. Thus with the help of importance sampling using an alternate distribution, we can estimate the tail probability much easily.

In WSOD, we compute image-level scores computed using instance-level scores with MIL pooling as in equation 1.8. The θ_i's in this case are the weight of a proposal which is equivalent to $p(x)$. θ_i's are obtained by evaluating the expensive RoI pooling operation at every iteration

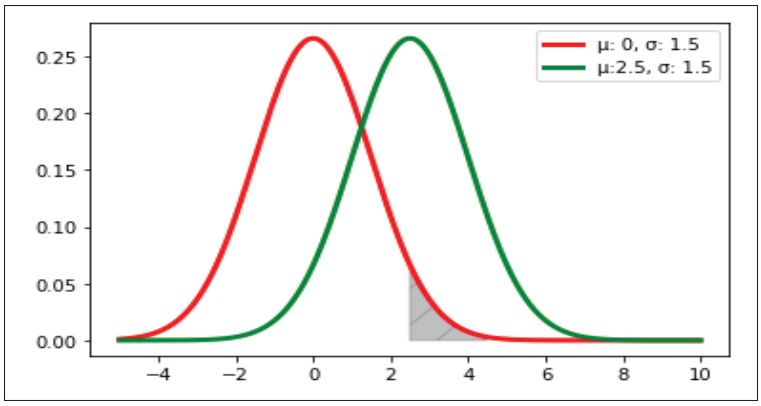


Figure 1.16 Importance sampling for estimating tail probability of a distribution

for each proposal. In our formulation, we construct an alternate distribution $q(x)$ where the probabilities are obtained by a score propagation process that evaluates only a few sampled proposals instead of the whole set at every iteration per image. As the accumulated scores nearly approximates the original semantics of a region the ratio of densities β_i's in this case is close to 1. Thus I_s^c in this case for a class c will be approximated as:

$$I_s^c \approx \sum_{k \sim \mathcal{M}(\beta_\parallel)} s_k^c \tag{1.12}$$

Where $\mathcal{M}$ is a multinomial distribution over the score vector β. In this way, we limit the expensive evaluation of the RoI pooling operation to k times.

1.11 Semi-supervised Object Detection

Semi-supervised object detection involves training object detection models using a combination of both labeled and unlabeled data. In the semi-supervised learning settings, we have a small set of labeled images $D_s = \{x_i, y_i\}_{i=1}^{N_s}$ and a large collection of unlabeled images $D_u = \{x_i\}_{i=1}^{N_u}$. Here N_s and N_u are the number of images in the labeled and unlabeled sets respectively. x_i's are the observed datapoints and y_i's are the corresponding labels. Note that labels are only available for

the supervised set. Figure 1.17 presents an illustration of the semi-supervised object detection settings.

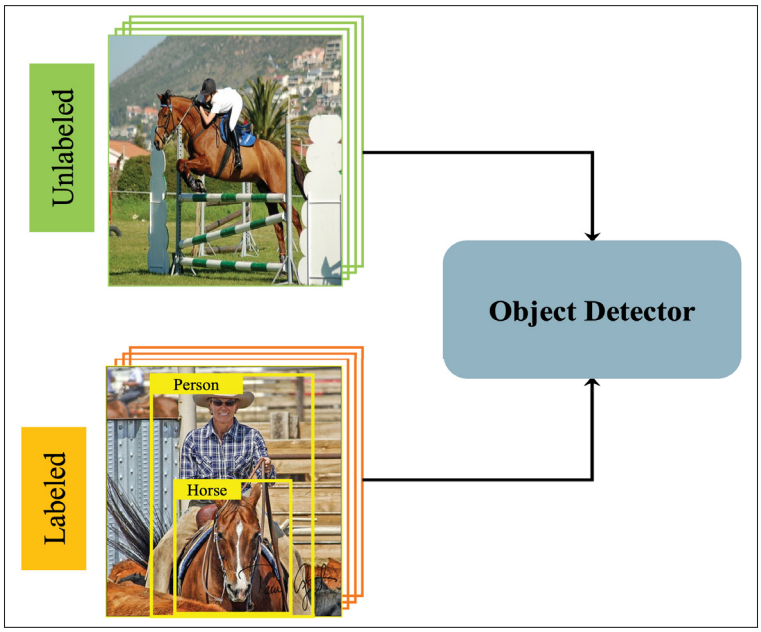


Figure 1.17 Semi-supervised object detection problem settings

Let's denote the underlying distribution from which x_i's are sampled as $p(x)$. A necessary condition for semi-supervised learning to be possible is that $p(x)$ should contain information about the posterior distribution $p(y|x)$ (van Engelen & Hoos, 2019). In that case, one might be able to use unlabelled data to gain information about $p(x)$, and thereby about $p(y|x)$. Otherwise, the unlabeled data is not useful for improving the label prediction (Zhu, 2008). Figure 1.18 shows an example of this with one of the most commonly used assumptions in semi-supervised learning, called the "low density" assumption. The low-density assumption states that the decision boundary should not pass through high-density areas in the input space. In the figure, a binary classification setting is shown. The labeled data points are shown with + and ∇ signs. If only the limited labeled data points are considered, the decision boundary obtained will not separate the two classes well as they have limited information (solid line). Using the large collection of unlabeled data, we could model the density function better and place the decision boundary at regions with less data density. As we can observe from the figure, this gives the

optimal decision boundary so our model will generalize better. This is how the semi-supervised methods help in improving the predictive accuracy of models.

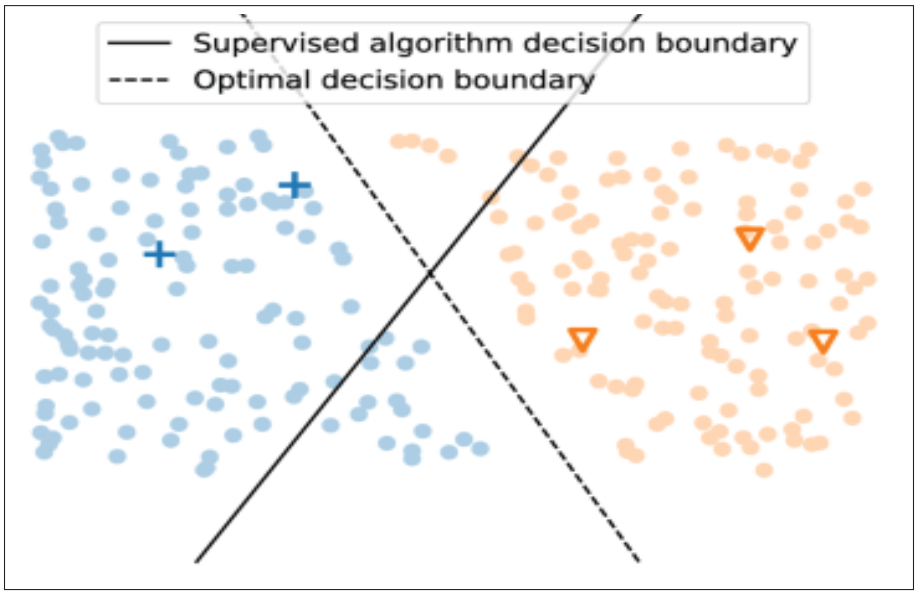


Figure 1.18 Impact of semi-supervised learning in the decision boundary. Taken from van Engelen & Hoos (2019)

While the majority of the semi-supervised papers are in the classification literature, in recent days they have equally become popular in object detection. The Mean-teacher based methods ((Guo *et al.*, 2022; Li *et al.*, 2022a; Liu *et al.*, 2021a; Xu *et al.*, 2021; Tang *et al.*, 2021; Chen *et al.*, 2022)) emerged as a clear winner with impressive results on the MS-COCO detection benchmark (Lin *et al.*, 2014) using very few annotated images. We will also use the mean-teacher framework for our small object detection problem settings. Next, we will review the major challenges in learning object detectors in semi-supervised settings.

1.12 Challenges in Semi-supervised Object Detection

We will discuss the main challenges in semi-supervised object detection in the following subsections. These challenges are mainly coming from how the unlabeled data is collected and how they are utilized in the learning process.

1.12.1 Unknown classes in the unlabeled data

The unlabeled data used in practical settings for semi-supervised object detection is not well curated. There might be unknown objects in it and we might be unaware of it. The majority of the semi-supervised methods use unlabeled data with pseudo-labels. When there are unknown classes, it might get pseudo-labeled as one of the known classes creating confusion in the detector learning process. Due to this unknown aware semi-supervised detection is getting popular nowadays (Du, Wang, Gozum & Li, 2022). Sometimes there will be no objects in the unlabeled images. This can also result in spurious pseudo-labeling that negatively impacts the model's performance. In some cases, domain shift is also observed in the unlabeled images.

1.12.2 Balancing the ratio of labeled and unlabeled samples

A good balance of labeled and unlabeled samples is crucial for semi-supervised object detection. Having so many unlabeled samples results in the model learning more with noisy pseudo labels. This is detrimental to the performance. Using mainly the samples from the labeled set results in inefficient utilization of the unlabeled data. Modern semi-supervised detectors achieve this balance at the minibatch level. Particularly, in a minibatch, they take an equal number of labeled and unlabeled samples (Liu *et al.*, 2021a; Tang *et al.*, 2021; Xu *et al.*, 2021). While practically this works to some extent, we have no theoretical justification of what is the right strategy. Assigning low weights to the unsupervised loss is also pursued in some papers. But this results in an extra hyperparameter which is difficult to tune.

1.12.3 Choosing the threshold for Pseudo-labeling

Pseudo-labeling is the most popular strategy for using unlabeled data in the semi-supervised learning process. Typically this is accomplished by using a confidence threshold so as to avoid noisy labels (Guo *et al.*, 2022; Liu *et al.*, 2021a). Predictions from unlabeled images with a confidence score above a threshold are designated as pseudo-labels. Choosing this threshold value is tricky. Most of the methods used higher threshold values to avoid noise, but this results

in the dominant classes getting more pseudo labels. If the dataset is imbalanced, this can seriously damage the performance compared to using supervised-only training. Recently, it has also observed that for one-stage detectors with severe foreground-background imbalance, using a single threshold is detrimental (Chen *et al.*, 2022).

1.13 Mean-teacher Framework for Semi-supervised Object Detection

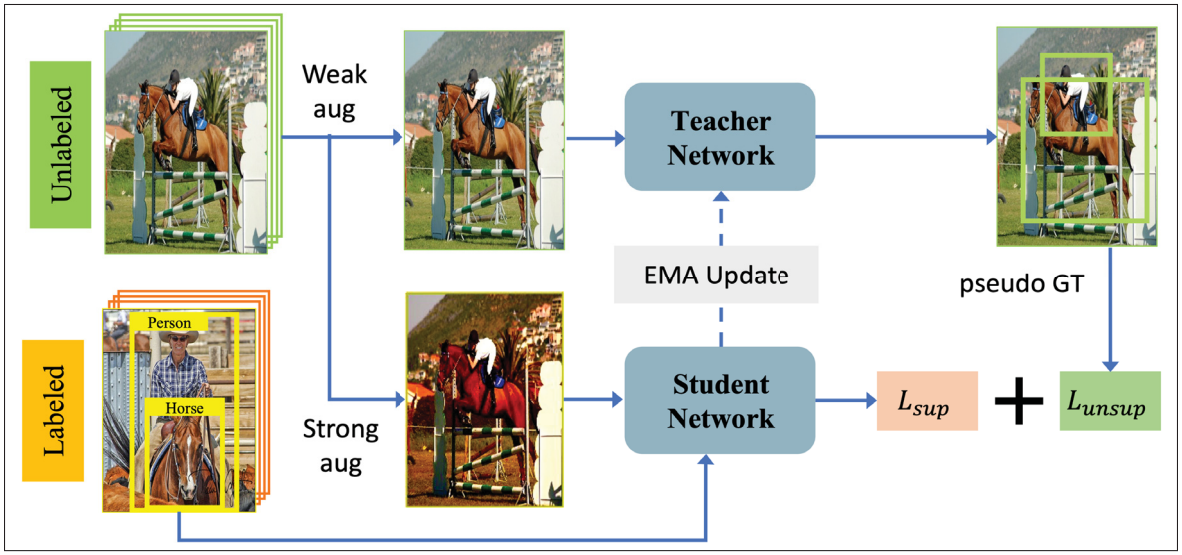


Figure 1.19 Mean-teacher framework for semi-supervised object detection

Recent advances in semi-supervised object detection are following the mean-teacher framework for semi-supervised learning (Tang *et al.*, 2021; Sohn *et al.*, 2020; Xu *et al.*, 2021; Liu *et al.*, 2021a; Chen *et al.*, 2022; Guo *et al.*, 2022). It combines consistency regularization and pseudo-label-based learning to give the best of the main-stream semi-supervised learning strategies. Figure 1.19 illustrates the mean-teacher framework for semi-supervised object detection. It consists of two networks; a teacher and a student network. The student network is learned by backpropagation optimizing the combined supervised and unsupervised loss. The teacher network is a temporal ensemble of the student weights updated through the Exponential Moving Average (EMA) during the course of training. Let θ_t and θ_s denote the teacher and student networks weights respectively. Let $\alpha \in [0, 1]$ be a hyperparameter controlling the rate of update.

The teacher network weight update using EMA is performed as follows:

$$\theta_t = \alpha * \theta_t + (1 - \alpha)\theta_s \tag{1.13}$$

The purpose of the teacher network is to provide pseudo-labels for unlabeled images. The teacher network receives a weakly augmented version of unlabeled images in a batch. The weak augmentation typically used are resizing and horizontal flipping (randomly applied). It then predicts the detection outputs which are post-processed to synthesize pseudo labels. The post-processing commonly used are confidence thresholding and Non-Maximal Suppression (NMS). As the teacher network accumulates the weights of a changing student network at a slow pace (controlled by a momentum parameter in the EMA), it is generally more accurate and so we use the teacher predictions for pseudo labeling (Tarvainen & Valpola, 2018).

The student network receives a strongly augmented version of the unlabeled image. The strong augmentation typically used includes color jittering, grayscale, Gaussian blur, and cutout patches which perform only pixel-level transforms, thus the bounding box labels need not be transformed. The student makes its predictions and a loss is computed between the pseudo labels provided by the teacher network. This ensures a consistent prediction between the strong and weak augmented versions of the same image. Note that consistency regularization is a widely used technique in semi-supervised learning (van Engelen & Hoos, 2019). The labeled data points in a minibatch are treated by the student network the same way as in the fully supervised setting where a supervised loss can be computed with the real ground truth provided. For the unlabeled data points, the same loss is computed but with pseudo-ground truth this time. The combined loss is then backpropagated to update the student network. Let $\mathcal{L}_{sup}$ and $\mathcal{L}_{unsup}$ denote the supervised and unsupervised loss respectively. Let λ be a hyperparameter controlling the trade-off between these losses. The combined loss is estimated as follows:

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda \mathcal{L}_{unsup} \tag{1.14}$$

In this framework, consistency regularization and pseudo labels - the two most widely used strategies in semi-supervised learning - are combined in an efficient way to get the best of both. However, existing methods mostly use one of these strategies in their SSL formulation. We hypothesize that the reason why the mean-teacher framework is better than all existing semi-supervised frameworks is due to this reason.

1.14 Density Crops for Small Object Detection

As we used semi-supervised object detection for detecting small objects, additional techniques were needed to boost the performance of small object detection. In our work, we relied on density crops for this, which perform small object detection in a focus and detect manner. With density crops, the regions clustered with small objects are "focused" and cropped out, then "detection" is performed on these crops as well as the original images. Density cropping is a widely used strategy in aerial image object detection. As the objects are tiny (and usually appear clustered) and the images are very high in pixel resolution, crops from clustered regions are extracted and processed by upsampling to facilitate better small object detection. A detector trained in the standard way in this imagery detects only the bigger objects and small objects will be mostly missed by the detector. Figure 1.20 shows an example of a Faster RCNN detector trained in the conventional way not producing any detection of the tiny objects.

Density crops as shown in figure 1.21 identify the regions where a cluster of small objects is present. These regions are then cropped out and processed in higher resolution after up-scaling for improved small object detection. The detection results from the crops are then merged with the detection on the input image Duan *et al.* (2021); Li *et al.* (2020); Yang *et al.* (2019). Density crops in effect are doing a focus-and-detect process.

The idea of focus and detection using density crops is exploited by many researchers (Duan *et al.*, 2021; Yang *et al.*, 2019; Li *et al.*, 2020; Deng *et al.*, 2020). Figure 1.22 shows an example proposed by Li *et al.* (2020). It basically identifies the cluster of small objects using an external density crop extraction module. The detection of the original image and upscaled

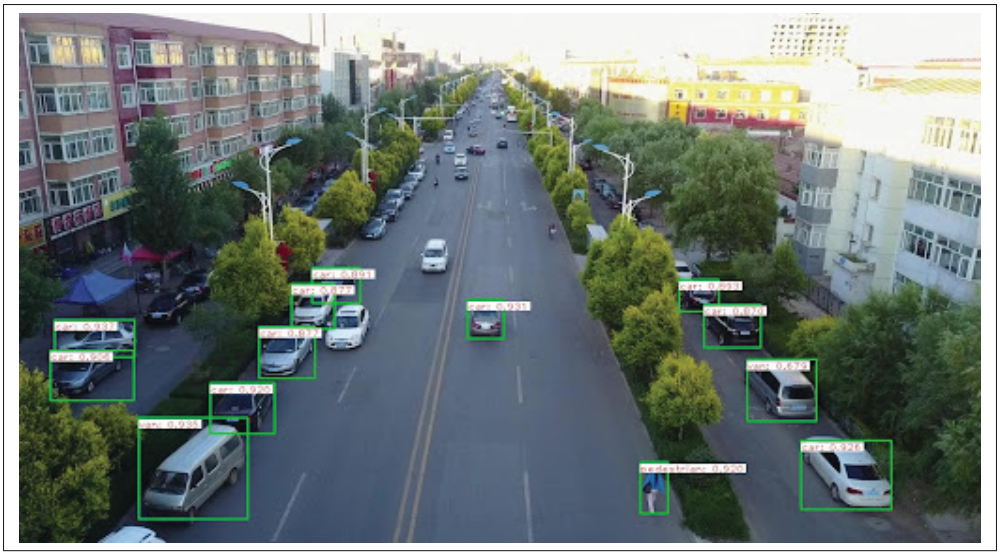


Figure 1.20 Detection results from a Faster RCNN detector trained in the conventional way

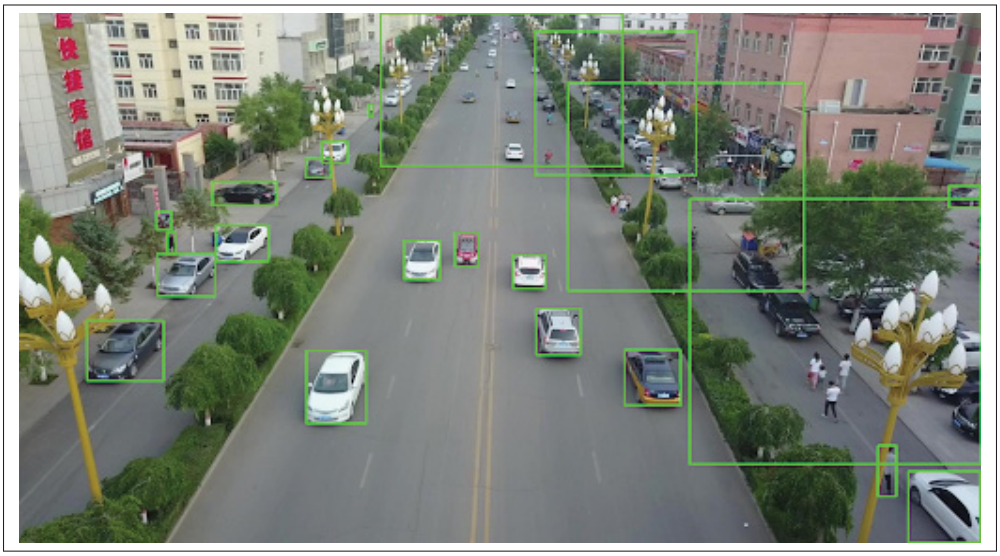


Figure 1.21 Density crops identified from an image. Note that the bigger objects are already detected from the original image. We identify regions with clusters of small objects using the density crop extraction process

crops are implemented by a standard detector. This way of using external modules for crop identification comes with challenges. They add additional parameters to train, training can become a multi-stage process (where the crop module is trained first for reliable crops, then the

detector is trained) and changes also happen in the loss function. Due to this, the detector used here cannot be easily trained in a semi-supervised fashion to reduce annotation costs. Because it is very difficult to adapt the semi-supervised training process to the density extraction module. We address this problem in our proposed cascaded zoom-in detector.

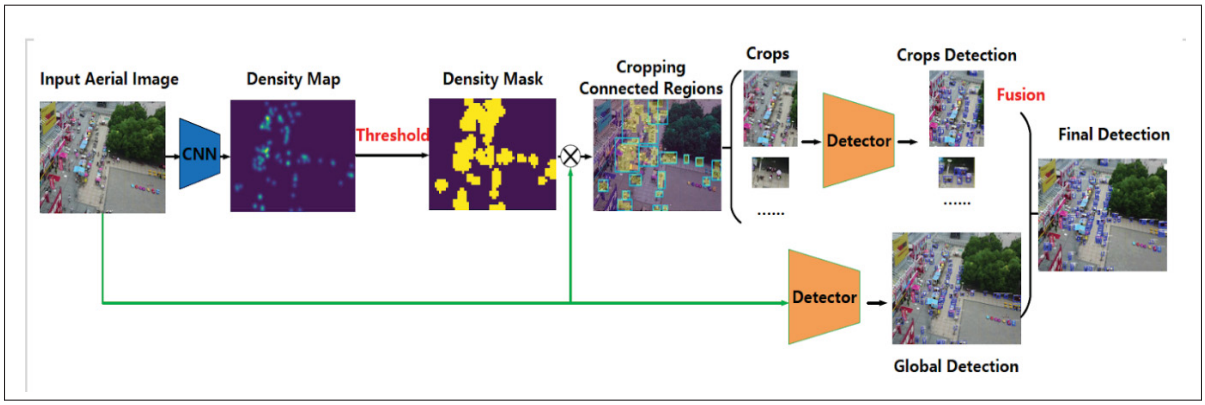


Figure 1.22 An example architecture of a density crop-based detection system. Taken from Li *et al.* (2020)

CHAPTER 2

CONVOLUTIONAL STN FOR WEAKLY SUPERVISED OBJECT LOCALIZATION

To localize the objects in images, the standard approach is to train the network with bounding box annotations. However, as we have seen, annotating object instances in an image is a time-consuming and costly affair. And it also won't scale to every detection problem we want to solve. Thus it is important to reduce this annotation burden with algorithmic solutions. As we have seen, classification labels are cheap to obtain and they are large in quantity compared to detection labels (e.g., 1.2 million in ImageNet (Russakovsky *et al.*, 2015) vs 11k in Pascal VOC 2012). A natural question then is, can we train the detection network with this incomplete image-level supervision and obtain object location as a bi-product? If this is feasible, one could obtain millions of images with class labels scrapped from the web-scale data and train a detector without the expensive box annotations. Though the web-scale data will be inherently noisy, can we beat the network trained using a small amount of precise location information with a network trained on a large quantity of weak noisy labels? Weakly supervised learning on images is trying to address this question.

In general, an image will contain multiple objects. The target of the detection method is to locate each object present in the image with a bounding box. Weakly supervised detectors (WSOD) are trained with image labels for this task. In order to simplify the problem settings, in this chapter, we will focus on localizing a single dominant object present in the image using the image label. This problem setting is called Weakly Supervised Object Localization (WSOL). We proposed a technique called Convolutional Spatial Transformers (CSTN) for learning to localize objects from image labels. The CSTN is designed with multiple location prediction, so as such it can be adapted to detect multiple objects from an image.

A closely related research direction to WSOL in the computer vision community is the explainable AI for images (Vilone & Longo, 2020; Linardatos P,Papastefanopoulos V, 2020; van der Velden, Kuijf, Gilhuijs & Viergever, 2022). Explainable AI in images cares about generating visual explanations for a classifier's prediction. This will help to explain why an image classifier

is making a particular prediction to the non-deep learning community. For example, it is important to produce visual explanations of the predictions made in areas like medical images to have trust in the deep learning algorithms (van der Velden *et al.*, 2022). Methods like CAM (Zhou *et al.*, 2016), gradCAM (Selvaraju *et al.*, 2017), gradCAM++ (Chattopadhay, Sarkar, Howlader & Balasubramanian, 2018), LIME (Ribeiro, Singh & Guestrin, 2016), TCAM (Belharbi, Ben Ayed, McCaffrey & Granger, 2023) and FCAM (Belharbi *et al.*, 2022) are focussing on extracting visual explanations from the classifier prediction. A bounding box localization can be produced from the resulting segmentation mask by thresholding it and finding the maximally connected component. The techniques from both research directions are used to localize objects using image-level supervision.

The core component of our proposed WSOL method is a Spatial Transformer Network (STN) (Jaderberg *et al.*, 2015). Preliminary studies have observed the localization ability of STN in classification settings on synthetic datasets like MNIST. We will present how the basic STN is learning to localize the objects in classification settings. The difficulties when using STN on natural images will be discussed thereafter, which will pave the way for the CSTN design.

2.1 Why STN for Localization?

Deep convolutional neural networks (CNNs) with Class Activation Maps (CAM) (Zhou *et al.*, 2016) are a prominent solution in the literature for WSOL problems (Singh & Lee, 2017; Zhang *et al.*, 2018a; Zhou *et al.*, 2016). They use spatial class-specific localization maps where high activations indicate the location of the corresponding object of the class. CAMs are obtained through standard convolution, and as such, are limited in their ability to accommodate large and unknown transformations, and variations in object scale, orientation, and pose (Dai *et al.*, 2017; Jaderberg *et al.*, 2015). Learning a transformation-invariant operation that can simultaneously handle different transformations is desirable for visual recognition systems. Spatial Transformer Networks (STNs) (Jaderberg *et al.*, 2015) have been proposed as a differentiable module that allows for spatial transformation of data within a CNN without manual intervention. This provides the network with flexibility in terms of adaption to the input image variations. Since

the location of the activation in CAMs is intrinsically dependent on the convolution operation, *flexible* convolution operation that *adapts* to scale, orientation, and other possible variations are preferable.

We believe that explicit components for learning localization should be present in a WSOL system. To this end, we propose to adapt a spatial transformed network to learn better localization. When used in a classification setting, STN crops out the relevant regions of the image so as to maximize the classification accuracy. See figure 2.1 for the transformation produced by STN on MNIST (Deng, 2012) dataset. However, naively adapting this global transformation to natural images doesn't do good with localization accuracy. When used on natural images, STN tends to focus on some discriminative object regions as shown in figure 2.2. Even with multiple STNs in parallel, each tends to specialize in certain object parts as shown in the figure. This is probably due to the large variation in the background on natural images, unlike MNIST digits. Also, the global transform producing a single bounding box is not suitable for localizing multiple objects if one wants to use them for detection. So we propose a novel adaptation of STN that addresses these limitations. We called it convolutional STN because it applies the spatial transform in a convolutional fashion. Next, we will illustrate the difference between the regular convolution vs STN convolution.

2.2 Regular Convolution vs STN Convolution

In this work, we investigate the use of STN (Jaderberg *et al.*, 2015) as an *adaptive convolution operation* to replace standard convolution. We refer to this operation as Convolutional STN (CSTN). This adaptation is achieved through the application of an STN convolution over each location. STN model learns affine transformations that can cover different variations including translation, scale, and rotation, allowing to better attend to different object variations. This provides more flexibility compared to standard convolution. Figure 2.3 illustrates the difference between both types of convolution. In standard convolution, the sampling grid of the convolution is fixed (hence it has a fixed receptive field) while in our CSTN, we transform the sampling

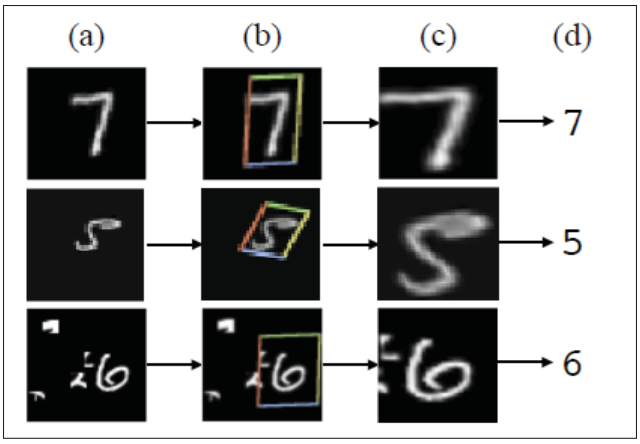


Figure 2.1 STN in action on the MNIST digit classification problem. Taken from Jaderberg *et al.* (2015). When fed with the (a) distorted MNIST images during training, (b) the Localization network of STN predicts a transform to align them properly, and (c) sampling from the aligned region by the sampler of STN

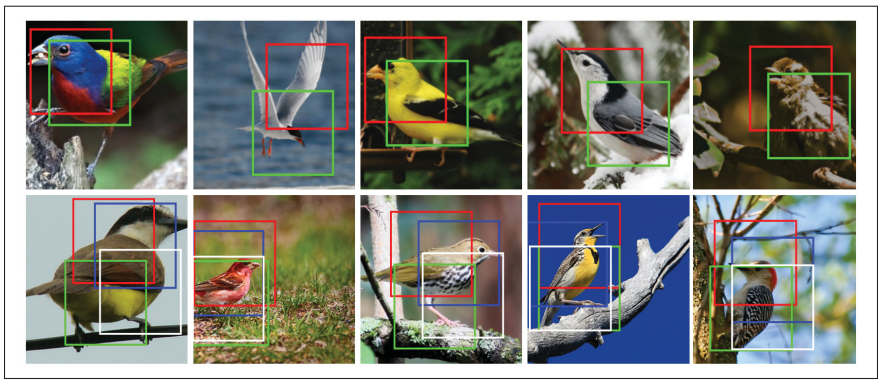


Figure 2.2 STN applied on the whole image localizing discriminative object parts. Taken from Jaderberg *et al.* (2015). In the top row, 2 STN are used parallel whereas 4 STN is used in the bottom row. With 2-STN one of the transformers (shown in red) learns to detect heads, while the other (shown in green) detects the body, and similarly for the 4 STN

grid using spatial transformers and sample the input feature map from the resulting locations allowing it to have a varying receptive field.

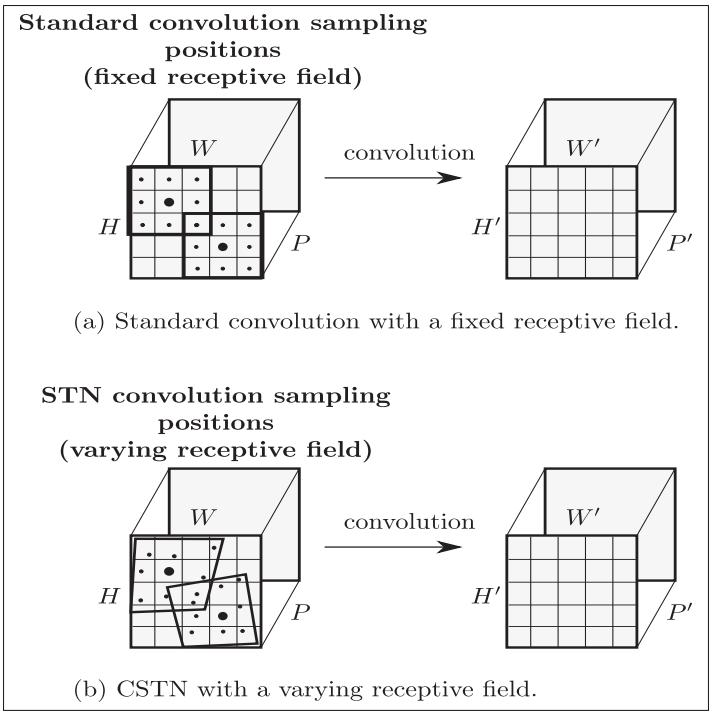


Figure 2.3 An illustration of the difference between standard convolution and CSTN. P and P' are the depth of the feature maps

The CSTN is in principle similar to deformable convolution proposed by Dai *et al.* (2017). To break the fixed geometry of a standard convolution, it learns a set of offsets for each position in the regular sampling grid. Deformable convolution has demonstrated improvements in object localization for the fully supervised object detectors (Dai *et al.*, 2017; Zhang & Kim, 2019). However, the deformation learned in this way is not a centralized one as each pixel in the sampling grid can move independently resulting in an irregular shape for the convolution. Active convolution unit proposed by Jeon & Kim (2017) attempts to learn the shape of the convolution. All these deformable convolution methods are studied in fully supervised algorithms, we are the first to study them in weakly supervised settings.

While the CSTN is able to adapt to relatively small local variations, it still faces the issue of adapting to large variations in terms of the receptive field. To alleviate this issue, we consider localizing objects of different scales at different levels (i.e., layers), using the FPN (Lin *et al.*,

2017b). The CSTN is applied at different levels of the feature pyramid. As the receptive field from the low layers can process only small regions of big objects, local convolution at that layer tends to localize small discriminative regions while missing the entire object. However, such layers are more adequate to localize small objects while high layers can miss them due to their large receptive field. To deal with this, an additional regularization term is introduced to drive specific layers to compete for the right scale. A joint probability over scale, location, and class is formulated based on the class scores through an aggregation process. This sums up the overall design of our CSTN for WSOL. The main contributions can be summarized as follows:

- a novel approach for WSOL with convolutional spatial transforms that explicitly learns to localize during classification.
- an adaptation of the FPN model (Lin *et al.*, 2017b) to weakly supervised settings for localizing objects of different scale, where the STN need to learn a small transform for the right scale.
- an empirical validation of the proposed approach over CUB-200-2011 bird dataset (Welinder *et al.*, 2010) and ILSVRC 2012 (Russakovsky *et al.*, 2015) localization dataset.

Next, we will present the detailed formulation of our CSTN for weakly supervised object localization.

2.3 Convolutional STN for Weakly Supervised Object Localization

To explain our architecture for WSOL, we start from the last convolutional layer of a CNN and show how it is used for object localization (see Figure 2.4(a)). Similar to one-stage object detection methods (e.g., SSD (Liu *et al.*, 2016), YOLO (Redmon & Farhadi, 2017)), we consider the location of a filter as the rough center of the object. In one-stage detectors, this location is then associated with a set of class probabilities that define which object is more likely to appear at that location and the coordinates of the object's bounding box, estimated as a regression. In our case, we do not have information about the bounding box of the object as our problem is weakly supervised (we only have image-level labels). Thus, to go from object labels to image labels we need an aggregation mechanism as detailed in the next subsection.

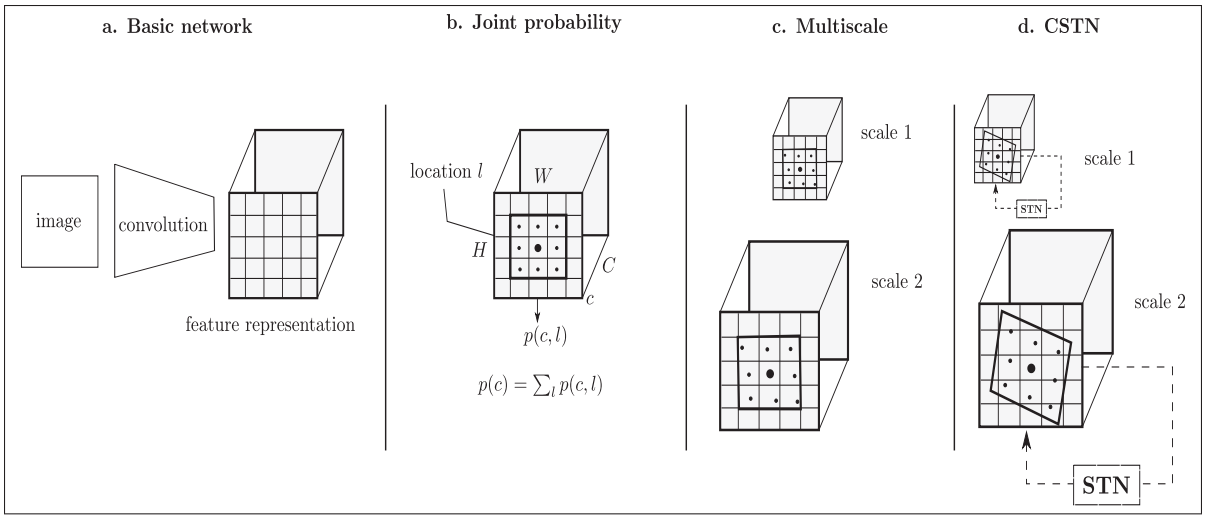


Figure 2.4 **Basic components of our system**: (a) One of the last convolutional layers of a CNN can already provide some information about the center of the object. (b) Our joint probability in location and classes is used to learn localization in a Weakly supervised manner (see text). (c) Using a multi-scale approach we can find not only the position of the object but also the scale (d) Adding our CSTN, we obtain a more refined localization of the object of interest

2.3.1 Joint class and location distribution

In our model, the last convolutional layer is a feature map f with $H \times W = L$ locations and C channels equivalent to the number of classes to classify. As shown in Figure 2.4(b), we can consider this feature map as voting for the most likely position and class in an image. We can thus convert this feature map into a multinomial probability distribution over classes and positions by applying a softmax on the two spatial dimensions and on the channels too. As we want each class and location to compete, we need to compute a single softmax on the three dimensions. Thus, instead of the common distribution over classes $p(c)$ as for classification, here we model the output of the CNN as a joint probability over classes and image locations:

$$p(c,l) = \frac{\exp(f_{c,l})}{\sum_{c'=1,l'=1}^{C,L} \exp(f_{c',l'})} \ . \tag{2.1}$$

With this joint probability distribution, we can obtain the class labels by marginalizing over locations: $p(c) = \sum_l p(c,l)$. This can be used to train our model for classification with standard

cross-entropy loss. However, with the joint probability, we can also obtain the maximum a posteriori (MAP) of the best location l^* and class c^* for a given image: $c^*, l^* = \arg\max_{c,l} p(c, l)$. This is the information required to estimate the location and class of the object of interest. This approach is simple and works well to find the center of the object. However, we are interested in getting the bounding box of the object in the image. We can consider the bounding box of the object as proportional to the receptive field of the used feature map. However, this would lead to square bounding boxes at the same scale. To overcome the scale problem, in the next section, we extend our approach to a multi-scale representation.

2.3.2 Multiscale search

For searching at multiple scales we use feature pyramids (Lin *et al.*, 2017b), because it does not add much computational cost to the method and it works quite well on several problems. With the feature pyramid, instead of considering a single feature map $f_{c,l}$, we use a representation composed by S feature maps, each representing the image at a different scale. Thus, we can extend our joint distribution to also scales: $p(c, l, s)$ (see Figure 2.4(c)). Again, by marginalizing over locations and scales we can obtain $p(c)$ used for training, and by selecting the MAP, we can find the location l^* and s^* of the object of interest. Now, we can find objects at different scales and different locations. However, still, all objects will have the same aspect ratio. A possible solution would be to use convolutional filters of different sizes that will generate different receptive fields and therefore different bounding box shapes. However, this approach will increase the computational cost and will be able to provide only discrete object sizes (defined by the convolutional filters aspect ratio). In the next subsection, we show how to learn a weakly supervised model that can adapt to any object size and aspect ratio.

2.3.3 Convolutional STN

While in fully supervised object detection most of the approaches regress a bounding box with the right object size, for weakly supervised models it is not possible because there is no ground truth to regress. In the original spatial transformer network (STN), a localization network is

trained to find global image transformations that can better represent the data and therefore minimize the training loss. The authors of the original paper (Jaderberg *et al.*, 2015) show that their approach improves the classification performance by focusing on the object of interest and at the same time being able to localize the object of interest without annotations in a weakly supervised manner.

However, we observed that STN works well when the data is quite clean (e.g., extended MNIST) and the sought transformations are relatively small. This is because the localization network of STN is trained with gradient descent, which is a local optimization. This means that when the transformation is too large or there is too much noise in the image, the local optimization will not be able to regress the correct transformation to localize the object and the training will fail. To overcome this problem, we propose to apply STN in a convolutional fashion. As shown in Figure 2.4(d), for each feature map location we apply a localization network that reads the local features and generates a transformation based on those. As the STN is applied locally to each part of the image, the required transformation is smaller and it is more likely that the simple gradient-based optimization used will work. Thus, in this work, the last layer is now composed of two stages: i) estimation of the local transformations $\theta = loc(f)$, in which loc is a convolutional localization network that for each feature map location f_l returns a corresponding transformation θ_l. ii). The final representation f' is the result of a convolution in which the convolutional filters are now applied with the feature map transformations θ: $f' = conv(f, \theta)$. The new layer is not much more expensive than a normal convolution because the additional computation is due only to the localization network. In contrast, being able to adapt the receptive field of the network to the local content of the image improves not only object localization but also image classification. Even though powerful, in the experimental evaluation, we note that the convolutional spatial transformer tends quite easily to overfit the training data. To avoid that in the next subsection we present two regularization techniques.

2.3.4 Regularization

Our multi-scale convolutional STN tends to focus on small regions. This is because, during training, the selected bounding boxes shrink to the most discriminative part of an object while the classification performance improves. To address this, we added a regularization/penalty term to the classification loss which prevents the affine transformation θ_i from having large deviations from its reference location θ_{ref}. This regularization term is,

$$L_\theta = \sum_{s \in S} \sum_{i=1}^{h_s \times w_s} ||\theta_{ref} - \theta_i||^2 \ . \tag{2.2}$$

Here we choose $\theta_{ref} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ corresponding to the identity transform.

The multi-scale search has also a bias toward localizing large objects from the lower levels of a feature pyramid. It is due to the fact that, in many cases, object parts are more discriminative than the entire object; lower-level layers will get strong activation for object parts of the large objects. In order to make the higher levels compete for localizing large objects, we enforce the difference between the maximum activation of the two levels to be zero or negative, such that the higher feature map will be more likely to be selected. This can be applied on any two scale-adjacent feature maps s_1 and s_2,

$$L_{scale}(x) = \max\left(0, \max_l p(s = s_1, l, c = c^*|x) - \max_l (p(s = s_2, l, c = c^*|x)\right) \tag{2.3}$$

Notice that for small objects that get localized from the lower level, this does not induce any penalty. Though competitiveness among the levels can be ensured in many ways, this simple regularization term has given satisfactory results in our experiments.

With these regularization terms, the final loss function optimized by our model is:

$$L(x, y) = L_{cls}(x, y) + \lambda L_\theta + \alpha L_{scale}(x) \tag{2.4}$$

where $L_{cls}(x, y)$ is the multi-class cross-entropy loss, α and λ are hyper-parameters to specify the strength of the STN and multi-scale regularization respectively.

2.3.5 Complete system

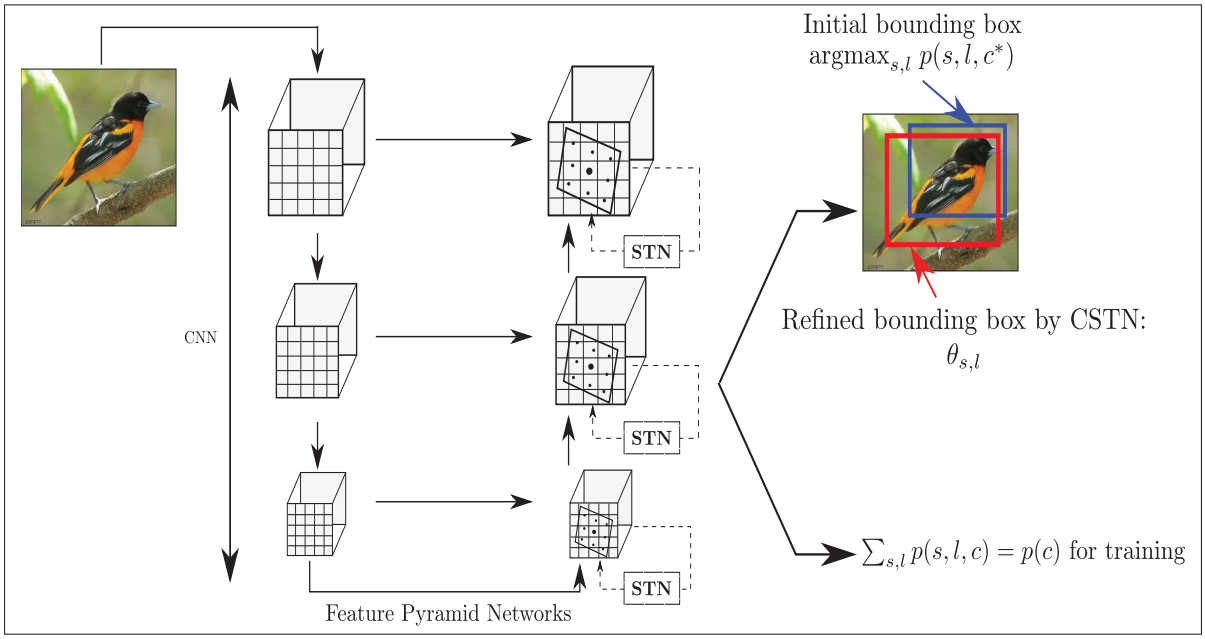


Figure 2.5 Overall CSTN system for WSOL when applied to multiple levels of the feature pyramid. The class probabilities for training are obtained by marginalizing the probabilities across the location and pyramid levels

Figure 2.5 summarizes our complete system. Given an image, a feature pyramid network builds semantic representations of the image at different scales. On all the scales, a CSTN is applied so that for each location and scale a localization bounding box is estimated. Finally, the scores of the STN are converted to a joint probability $p(c, l, s)$ over classes, locations, and scales. This can be converted to $p(c)$ by marginalizing over scales and locations to obtain the class probabilities needed to train the model in a weakly supervised manner. During training the proposed regularization is also applied. The joint probability is used at test time to localize the object by finding the maximum scoring transformation. This is estimated over the scales and location as shown below.

$$bbox_{max}^{c^*} = \max_{s,l} p(s, l, c^*) \tag{2.5}$$

2.4 Experiments

The detailed empirical study and ablations of the regularization terms are presented in this section. The datasets and experimental setup are detailed first.

2.4.1 Experimental setup

We evaluated our multiscale convolutional STN model on the CUB-200-2011 dataset (Welinder *et al.*, 2010) and ILSVRC 2012 (Russakovsky *et al.*, 2015) localization dataset. CUB-200-2011 contains 11,788 images of 200 bird species with 5,994 images for training and 5,794 for testing. The ILSVRC 2012 dataset contains 1.28M training images and 50,000 validation images. There are 1000 categories of objects. For both datasets, we evaluate the performance in terms of classification and localization accuracy. An image is said to be correctly localized if the predicted class matches the true class and the predicted bounding box has a 50% overlap with the ground truth. The localization accuracy is denoted as Top-1 Loc in the results. For explicitly measuring the localization performance(regardless of the classification accuracy), another metric called GT-Known Loc is used where the GT image label is provided. In that case, localization is deemed correct if the 50% overlap criterion is satisfied. Unlike CAM, our method can provide multiple bounding boxes per image. But in Top-1 Loc we are only using the box with the highest score. When this top-scoring box is centered on the object, we get the best bounding box. However, this is not always the case with CSTN, especially when the top-scoring box is focusing on the discriminative object regions. There could be other boxes, for which the score is very close to the top box but they overlap well with the ground-truth box. So we also considered a metric which we call the Top-5 box localization where we check if one among the top five boxes with high scores has 50% overlap with the object. Top-5 box localization gives interesting

results regarding the localization ability of our method in contrast to the CAM. We measured GT-Known Top-5 box Loc in this comparison.

We used ResNet101 (He, Zhang, Ren & Sun, 2016) as the backbone network which is pre-trained on ImageNet (Russakovsky *et al.*, 2015). We removed the last average pooling and fully connected layer and added an additional convolution(with 3×3 filter size and padding 1) and batch norm (Ioffe & Szegedy, 2015) layer. Feature pyramid is obtained from this network with its last two levels as described in Lin *et al.* (2017b). The input images are resized to 320×320 pixels. For data augmentation, we used horizontal flip with 50% probability. Images are normalized with mean = $[0.485, 0.456, 0.406]$ and std = $[0.229, 0.224, 0.225]$ as in ImageNet training (Russakovsky *et al.*, 2015). The model is trained on NVIDIA GTX 1080 GPU with 12GB memory.

2.4.2 Ablation Studies

The ablation studies are conducted to assess the impact of spatial transform, multi-scale localization, and the regularization on θ. We used CUB-200-2011 dataset in our ablation experiments.

2.4.2.1 Impact of CSTN over normal Convolution

To assess the importance of the spatial transform, we computed the localization accuracy when the default box is used for localization instead of the transformed output from STN. Note that this does not change the training procedure, since CSTN is still used the same way to learn the localization. At the implementation level, instead of using the transformed coordinates, we used the original coordinates to compute the localization performance. Table 2.1 shows the result of this study on both datasets. It can be observed that the transform is improving the localization around 5-8%. To see this impact visually, figure 2.6 shows some sample images where the transform is modifying the original receptive field box to improve the localization. It also highlights some failure cases where the transform is producing wrong localization.

Table 2.1 Impact of transform on the localization performance. For both datasets, the CSTN is important to obtain good localization performance

Dataset	**Top-1 Loc**	
	without transform	with transform
CUB-200-2011	40.64	49.03
ImageNet	36.69	42.38

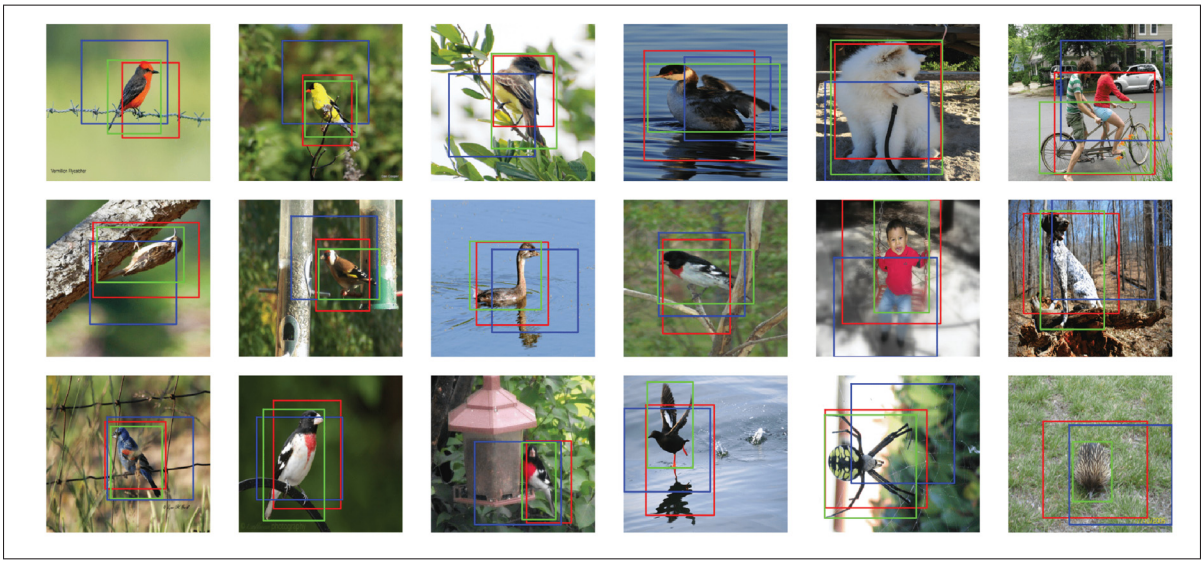


Figure 2.6 Demonstration of transforms learned by CSTN on CUB-200-2011 and ILSVRC dataset. The last column shows some failed localization on the ILSVRC dataset. The non-transformed box is shown in blue, the transformed box is in red, and the ground truth is in green

To further study whether the CSTN is learning a good representation for localization we compared the localization performance with and without CSTN. For the case without CSTN, we used the same architecture and classification head, the only difference is that no transform is learned in this setting, i.e., instead of CSTN, a normal convolution is used. The classification head is now classifying the fixed sampling space of the convolution. Table 2.2 shows the result of this study on the CUB-200-2011 dataset. It can be observed that without CSTN, the localization performance has reduced drastically. The classification performance also goes low but the

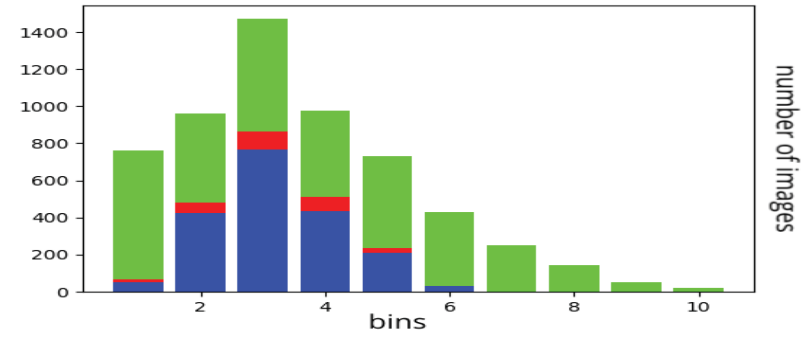
impact is less. This means that the CSTN not only learns to better localize objects in the image, but it also learns a better representation of the object that produces an improved classification.

Table 2.2 Localizing with and without convolutional STN on CUB-200-2011 dataset. It can be observed that the CSTN is very effective in learning a good representation for localization. It improves the localization by 26.79%

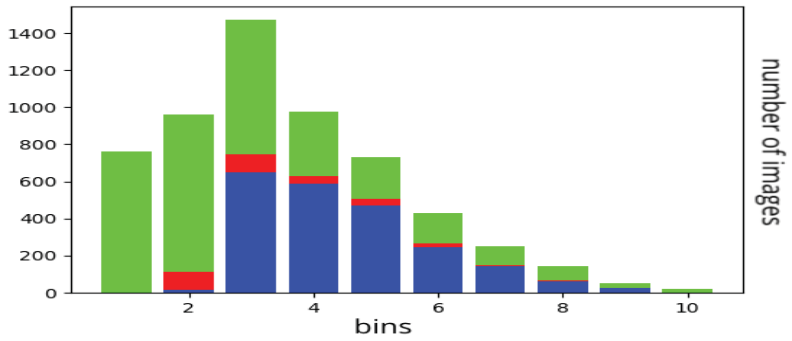
Type	**Top-1 Class**	**Top-1 Loc**
Without conv STN	77.40	21.64
With conv STN	78.46	49.03

2.4.2.2 Impact of multi-scale regularization

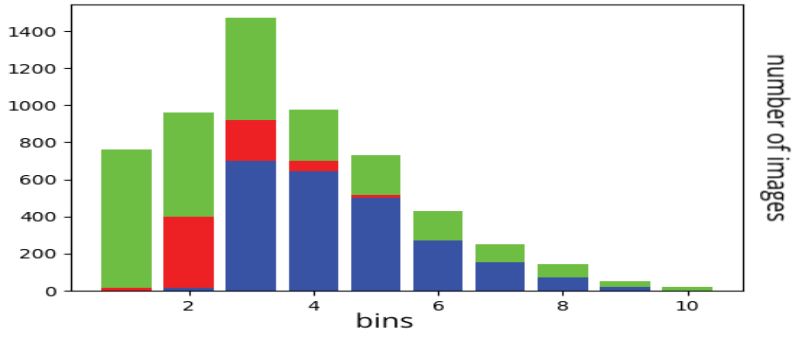
The multi-scale localization is another important component in our model. To assess the importance of this, we conduct ablation experiments with localization from two levels of the feature pyramid independently and compare it with the model where these levels are combined. Figure 2.7 shows the results of this study. Here a histogram is created by dividing the area of the bounding box into 10 bins of equal size. The histogram shows in green the total number of samples at each resolution and in blue and red the percentage of images that are correctly localized in each bin for the model without and with bounding box transformations respectively. From figure 2.7(a) and 2.7(b) we see that different levels are specialized on different object sizes. With the multi-scale model (Figure 2.7(c)) we balance the localization between the two levels and improve the localization accuracy. Notice also that the effect of the bounding box transformation becomes stronger when using a multi-scale model. This is in line with our hypothesis that the STN performs a local optimization and for improved performance, the transformations should be relatively small from a reference size. This can be compared to learning the transforms with respect to anchor boxes in the fully supervised object detectors (Ren *et al.*, 2015).



a) Localization from level 4 (Top-1 Loc is 37.78%)



b) Localization from level 5 (Top-1 Loc is 42.91%)



c) Localization with the multi-scale model combining levels 4 and 5 (Top-1 Loc is 48.43%)

Figure 2.7 Impact of multi-scale localization. Localization from each level is compared with the multi-scale model which combines all levels. The histogram is created by dividing the area of all bounding boxes into 10 equal bins. Green bars show the number of images in each bin, the red bar shows the number of images that are correctly localized by CSTN in that bin and the blue bars show the number of images correctly localized without the bounding box transformation

2.4.2.3 Impact of θ regularization

Another key component of our method is the regularization on θ. We observed that without this regularization, the learned transformations are not from the distribution of possible bounding boxes. The transformations tend to overfit and shrink to discriminative image parts resulting in poor localization. Figure 2.8 shows samples of bounding boxes learned without using regularization on θ. To obtain a good localization, tuning the hyperparameter λ is critical. Table 2.3 shows the performance in classification and localization for different values of λ. As expected, while the model classification is barely affected, localization is highly affected by this parameter. For the regularization on the scales, we found that α can vary in a range of values without affecting too much the localization results. Thus we did not include a study on that.

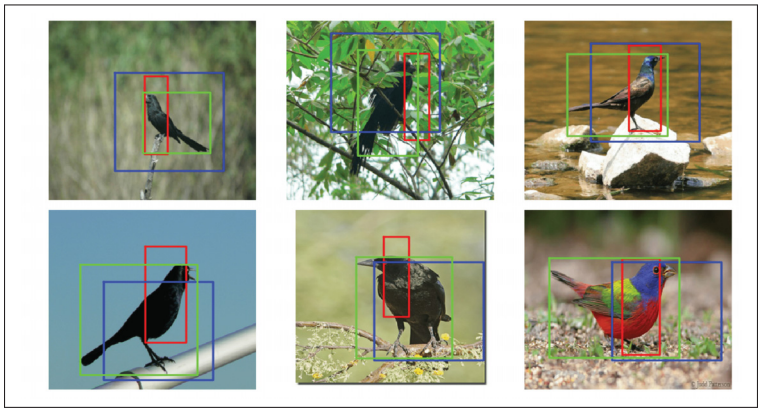


Figure 2.8 Transforms learned without using the regularization on θ. The receptive field box is shown in blue, the transformed box is red and the ground truth is green. It can be observed that the boxes learned without this regularization are not from the distribution of possible bounding boxes

2.4.3 Comparison with state-of-the-art methods

We compare the localization of the CSTN with state-of-the-art solutions for WSOL. Results are summarized in table 2.4 and table 2.5 for CUB-200-2011 and ILSVRC 2012 respectively.

Table 2.3 Impact of λ on classification and localization accuracy. For a high value of λ the localization accuracy tends to be the one obtained without STN. For no regularization, the transformations become too strong and focus on small parts of the object thus producing a very poor localization score

λ	**Top-1 Loc**	**Top-1 Class**
0.01	27.39	78.98
0.001	30.88	78.63
0.0001	49.03	78.46
0.00001	45.52	78.25
0	5.13	77.32

On the CUB-200-2011 dataset, CSTN performs better than all the CAM-based methods. except the ADL (Choe & Shim, 2019). In this dataset, the scale of objects is distributed unevenly, i.e., many objects are of nearly the same size and extreme variations in the size are very less (not too many small and large objects). As shown in the ablation study, different levels of the CSTN specialize on different scales, therefore, we can get the best of the localization from this model by focusing more on the crowded scales (where there are many objects). The hyper-parameter α is not very sensitive to the Top-1 Loc, so it can be tuned fairly easily. The difference in performance with ADL is mostly due to the wrong location selection as the recall is still close to 99%(so the CSTN is able to produce transformations that match the object sizes). The GT-Known Top-5 Loc is around **2.5%** higher than the GT-Known Loc of the ADL. This also reinforces our claim that the CSTN is learning better localization. Compared to all the CAM-based methods including the state-of-the-art ADL (Choe & Shim, 2019), CSTN has some clear advantages. These methods need rigorous tuning of their hyperparameters to obtain good localization. The hyperparameters of CSTN are not very sensitive to localization accuracy. The α regularization term can be avoided if we can find a suitable heuristic that tells whether the object is small or large. Thus small objects can be localized from the lower level and large ones from the higher level. Then it works similarly to the level selection in multi-scale fully supervised detectors based on the area of the ground truth box. The θ regularization is also fairly easy to tune as shown in the ablation

experiments. Recent studies observed that WSOL algorithms which improve the localization based on erase and learn strategy (Zhang *et al.*, 2018a; Choe & Shim, 2019; Singh & Lee, 2017) are very sensitive to their hyperparameters (Choe. *et al.*, 2020).

Table 2.4 Performance comparison on the CUB-200-2011 test set. Convolutional STN performs better than all other methods, except ADL. The Top-1 class is left blank for some methods because it is not reported in the original paper

Method	**Top-1 Loc**	**GT-Known Loc**	**Top-1 Class**
CAM (Zhou *et al.*, 2016)	41.00	71.13	-
HaS (Singh & Lee, 2017)	44.67	73.32	76.64
ACoL (Zhang *et al.*, 2018a)	45.92	75.30	71.90
SPG (Zhang *et al.*, 2018b)	46.64	74.11	-
ADL (Choe & Shim, 2019)	**62.29**	78.62	80.34
CSTN	49.03	76.06	78.46
CSTN Top-5 box	-	81.14	-

On the ILSVRC dataset, CSTN is outperformed by many of the CAM-based methods. This is probably due to the sensitivity to the scale. The number of objects in different scales is nearly uniformly distributed in this dataset. So the multi-scale localization should specialize on each scale equally well in this case. This can be better explained with the histogram of localization on ImageNet shown in figure 2.9. As we can see, it favors the localization towards large objects in this case. As a result, it fails to localize most of the smaller objects. The GT-Known Top-5 Loc in this case is comparable to the GT-Known Loc of the state-of-the-art methods including ADL and SPG.

We believe that improving the multi-scale localization component of our method can close this performance gap compared to the state-of-the-art CAM-based WSOL. If we try to localize an object with the wrong scale (i.e, from the wrong level of the feature pyramid), it will end in getting stuck at some discriminative object region. Figure 2.10 shows some failure cases of this when localizing large objects using CSTN. Since the end goal of STN is still to get a good classification, it will not try to localize the integral object. The softmax aggregation strategy is a simple and straightforward expansion to introduce the multi-scale capability. Having better

Table 2.5 Performance comparison on the ILSVRC validation set. The Top-1 Loc is competitive but due to the sensitivity to scale, convolutional STN fails to localize small objects. The sensitivity of the CAM to scale is less, so this can be the reason for the difference in Top-1 Loc

Method	**Top-1 Loc**	**GT-Known Loc**	**Top-1 Class**
CAM (Zhou *et al.*, 2016)	42.80	61.10	66.60
HaS (Singh & Lee, 2017)	45.21	63.12	70.70
ACoL (Zhang *et al.*, 2018a)	45.83	62.73	67.50
SPG (Zhang *et al.*, 2018b)	**48.60**	64.24	-
ADL (Choe & Shim, 2019)	48.43	63.72	75.85
CSTN	42.38	60.48	69.48
CSTN Top-5 box	-	63.45	-

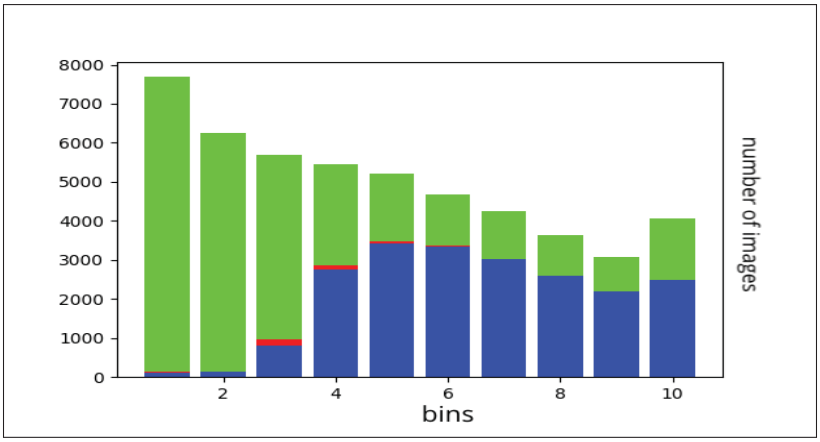


Figure 2.9 Histogram of localization on ImageNet validation set. The histogram is created by uniformly dividing the range of the area of objects in the validation set. It can be observed that the small objects are not localized well by CSTN

methods to select the matching scale can bring the benefit of CSTN to all such multi-scale improvements. Moreover, improving the box selection strategy can also give better Top-1 Loc, since our GT-Known Top-5 Loc is always good.

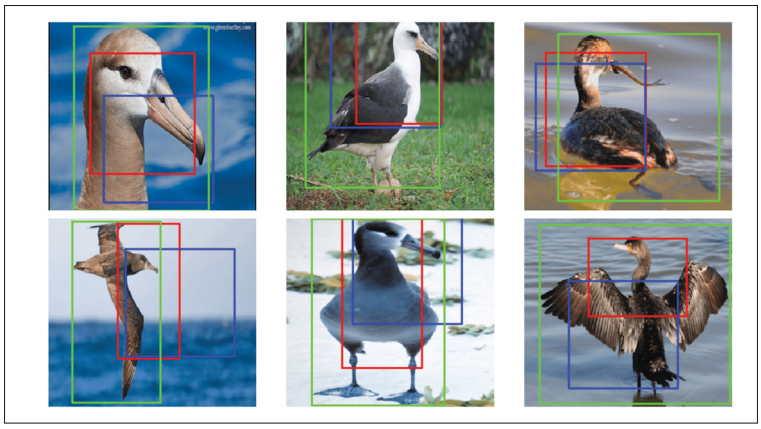


Figure 2.10 Localizing large objects using the wrong scale. The STN fails to learn large transforms for this case to give an accurate localization The receptive field box is shown in blue, the transformed box is red and the ground truth is green

2.5 Conclusion

In this work, we introduced a novel method for weakly supervised object localization. Different from the dominant paradigm of Class Activation Maps, we showed that the use of a convolutional spatial transformer can lead to a competitive performance in localization. Compared to the activation map-based methods, the convolutional spatial transformer is less sensitive to their hyperparameters for weakly supervised localization. This component can be plugged into any convolutional network giving an end-to-end weakly supervised localization module. Different from CAMs, CSTN can give multiple box predictions but selecting the correct localizing box has to be carefully designed. The learning of the convolutional STN is fairly easy and it adds a few additional convolutional layers to the standard CNN. Our Convolutional STN with multi-scale localization gives competitive results on the benchmark datasets. Empirical study reveals that the localization with convolutional STN is sensitive to the object scale and we proposed two regularization strategies to deal with it.

The main limitation of the proposed STN is its struggle to handle object scale variation. The impact of this struggle is visible in the localization results from the ImageNet dataset which has a large variation in scale compared to the CUB-200-2011 dataset. Furthermore, preliminary

studies conducted on the PASCAL VOC dataset agree with the same observation. The PASCAL VOC dataset has multiple objects per image unlike the ImageNet dataset because it is an object detection dataset. The multi-scale regularization component proposed here is not easy to use with more than two levels of the feature pyramid. Thus we concluded that the practical benefit of using CSTN for weakly supervised object detection will be less. Especially, CSTN is difficult to use for localizing multiple objects of varying sizes from an image. Although the transformation learned at one of the feature levels can localize the object well, it is hard to find the best box based on the joint probability alone. So the selection of the best bounding box for multiple objects cannot be easily performed. In FPN while considering multiple feature levels and every position in each feature level, there will be dense box predictions (of the order of 100k) to make this search process very hard. Thus we resort to not using the CSTN on the weakly supervised detection research. Instead, we will investigate WSOD with candidate proposals obtained from selective search or edge box in the next chapter.

CHAPTER 3

SEMI-WEAKLY SUPERVISED OBJECT DETECTION BY SAMPLING PSEUDO-GT BOXES

While WSOL solves the simple problem of localizing the single dominant object in the image, we often need to localize multiple object instances belonging to different categories from a single image. This is typically achieved with object detectors in computer vision (Girshick *et al.*, 2014; Sermanet *et al.*, 2014). Object Detection(OD) is a multi-task optimization problem that typically has a classification head and a localization head for instance/object classification and localization. The classification head will identify the object category localized by the localization head. The classification head typically optimizes cross-entropy loss and the localization head optimizes regression loss.

As our fundamental goal is to reduce annotation efforts, we are pursuing weakly supervised and semi-supervised methods for object detection. As we have seen in chapter 2, the Conv-STN model is struggling to localize objects when there are multiple instances and more categories of objects in an image. So we need advanced techniques to use weak image-level labels for detecting multiple object instances. In this chapter of the thesis, we focus on semi-weakly supervised object detection where we leverage weak-image level labels along with a few bounding box annotated images to efficiently train object detectors with less annotation burden. We propose an efficient method to obtain pseudo-GT labels on weakly labeled images. Then we train the detector with weakly labeled and fully labeled images together. The remainder of this chapter is organized as follows. First, we present our sampling-based learner for WSOD where we can plug in off-the-shelf detectors and train it with weak image-level labels. However, the localization of multiple objects in an image is still challenging. To mitigate this, we use a few images with bounding box annotations and design a semi-weakly supervised detector in section 3.3. Empirical studies to understand the strength and weakness of the proposed system is further presented.

Even today the advancements for WSOD are mainly focusing on techniques for better instance localization and ambiguity resolution (Vo *et al.*, 2022; Huang, Zou, Kumar & Huang, 2020) of the WSDDN model. However, the customized architecture for WSOD is making it difficult to translate the advances in fully supervised detection available in weakly supervised detection. What if we can utilize off-the-shelf detection architectures for weakly supervised object detection? We tried to address this research problem in this work.

3.1 Sampling-based Weakly Supervised Object Detection

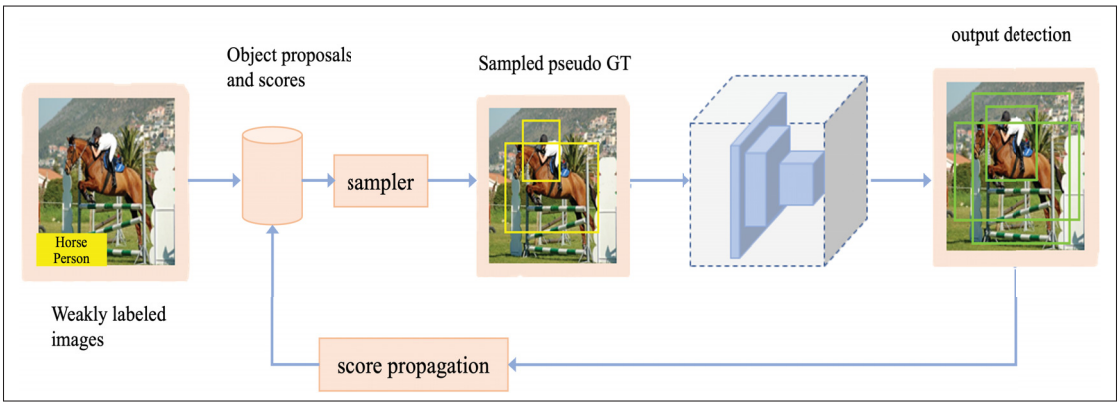


Figure 3.1 Our sampling based WSOD with of-the-shelf detector

Figure 3.1 illustrates the design of our sampling-based weakly supervised detector. Suppose we have object proposals extracted from each image in the training set. For each image, we also have an associated score matrix in memory with size $N \times C$ that stores the score of each proposal for each category. With the given weak labels provided for an image, we sample K boxes for each category label. These boxes are considered pseudo-GT boxes. Then we pass the pseudo GT boxes and the image to the detection network which computes the loss and produces output detections. The output detection includes a set of boxes and corresponding classification scores. The classification scores are propagated back to the object proposals based on an overlap criterion. As we can observe, the detection network can be any off-the-shelf fully supervised detection architecture. Because we use pseudo-GT in place of real GT in a fully supervised

case, a network architecture can be readily plugged into this framework. Now we will see the components of this framework in more detail.

3.1.1 Sampler

Let the scores of the class c in the i^{th} region proposal p_i is $s_i^c = f_c(p_i, I)$ where f is the RoI pooling operation. Let us denote by h^c the weighted sum of scores for a class c

$$h^c = \sum_{i=1}^{N} \theta_i f_c(p_i, I), \tag{3.1}$$

Here, we propose to approximate h^c with Monte Carlo sampling, in which instead of computing the sum over entire bounding box locations we uniformly sample K boxes:

$$h^c \approx \hat{h}^c = \sum_{k \sim \mathcal{U}} \theta_k f_c(p_k, I), \tag{3.2}$$

where θ_k is the weight associated with the bounding box k. This allows us to compute only K evaluations of the expensive f, while using an unbiased estimation of the weakly supervised scoring function. We call these samples pseudo-GT bounding boxes these samples can be passed to the detection algorithm as ground truth annotations. This allows our algorithm to use any off-the-shelf object detector without modification.

However, when sampling, the weights θ_k associated with each bounding box score cannot be computed directly because it is the normalized version of the object score $f_c(p_k, I)$ for the box p_k, but we do not have all scores for the normalization factor, as we sampled only a few boxes. Instead, for each image I we keep in memory the scores s_i (obtained using the score propagation) associated with the proposal p_i and update only the score of the K sampled bounding boxes. Then, θ_k will be computed as the normalized version of s_k:

$$\theta_k = \frac{\exp\{\frac{s_k}{T}\}}{\sum_j \exp\{\frac{s_j}{T}\}} \tag{3.3}$$

At convergence where the scores f do not change anymore, $\hat{h}$ becomes h. Therefore $\hat{h}$ is an unbiased estimation of h. While this approach would work, sampling uniformly any possible image bounding box proposal p_l would make the learning very slow because most of the time the sample would not come from the object of interest. Instead, in this work we use an importance sampling approach. We use θ_k as sampling probabilities associated with a multinomial distribution $\mathcal{M}(\theta_k)$ to sample bounding boxes so that the bounding box proposal p_k would have a probability θ_k to be sampled. In this case, in order to maintain the same estimation of h we need to divide by the sampling probability θ_k. Thus the final estimation of h will be:

$$\hat{h} = \sum_{k \sim \mathcal{M}(\theta_k)} f_c(p_k, I) \tag{3.4}$$

which is again an unbiased estimator of the classification score of an image, but with lower variance.

We also considered other approaches for sampling and the proposed importance sampling emerged as a clear winner in more practical settings. Ideally, we want the sampler to explore the search space of candidate regions well and at the same time sample more from the regions which has higher accumulated scores. To balance this explore-exploit tradeoff we considered a multi-armed bandit sampler. It has two components; The explore component which assigns higher scores for the regions that are not explored. The exploit component gives higher scores to regions with higher classification scores for the given class. The final score of a region proposal is a weighted sum of the two as shown below:

$$m_k^c = \overbrace{s_k^c}^{\text{exploit}} + \beta \overbrace{\sqrt{\frac{logn}{o_k}}}^{\text{explore}} \tag{3.5}$$

where β is a hyperparameter to control the tradeoff between the two. Here o_k is a measure quantifying how often a proposal is sampled and n is the current epoch. o_k is updated as the

IoU of the max-overlapping detection box to the proposal box p_k in a given epoch:

$$o_k = o_k + \max_j \texttt{IoU}(\text{detection\_box}_j, p_k) \tag{3.6}$$

The sampling in this case is performed by selecting top K proposals based on the score m_k^c. Let $\hat{\mathcal{B}}^c$ denote the sampled pseudo GTs for the class c. With a multi-armed bandit sampler, $\hat{\mathcal{B}}^c$ is obtained as:

$$\hat{\mathcal{B}}^c = \texttt{topk}(m_k^c) \tag{3.7}$$

While the explore-exploit paradigm works when the sampling is performed on a small set of curated proposals per image, it does not converge when there are 2,000+ proposals per image. Oftentimes, we extract thousands of proposals per image to have a good recall of all objects present in the image. More details will be provided in the results section.

3.1.2 Score propagation

Score propagation is the component that updates the score values $\mathcal{S}^P$ of the object proposals $\mathcal{P}$. If the output bounding boxes of the detector $\mathcal{D}$ would correspond to the object proposals $\mathcal{P}$ as in Girshick (2015), we could directly copy the detection values to our pool of proposals. Instead, as in modern detectors the output detections $\mathcal{D}$ are generated by a regression, we propose a method to propagate the scores from the output detection $\mathcal{D}$ to the scores $\mathcal{S}^P$ of our object proposals. During learning, the proposals will accumulate scores from their overlapping boxes produced by the detector. In our design, we define the score propagation according to the overlap between a proposal p_i and a detection d_i. This will help the proposals to aggregate the detection scores of its neighborhood region during learning.

The score values are initialized to 0. Then, during learning, their scores will be updated based on detection scores. We explored several criteria for propagating the score and observed that propagating scores from the maximum overlapping detection boxes helped the model collect better semantics for the region. Thus, we define γ as the maximum intersection over union

between proposal p and all detection boxes $d \in \mathcal{D}$: $\gamma = \max_{d \in \mathcal{D}} \frac{p \cap d}{p \cup d}$. So, for each proposal p_i present in the image, we propagate its score $s_{i,c}$ proportional to γ:

$$s_{i,c} = (1 - \gamma) s_{i,c} + \gamma \cdot s_{d,c}, \tag{3.8}$$

where $s_{d,c}$ is the score of the maximum overlapping detection box d for category c. In this way, scores associated to the proposal l with high overlap with the detection d will receive a strong update, while scores of detections with low overlap will not influence the stored score $s_{i,c}$.

As the key components of our sampling-based WSOD are presented, we will now try to understand the training process. The training process is shown in 3.2

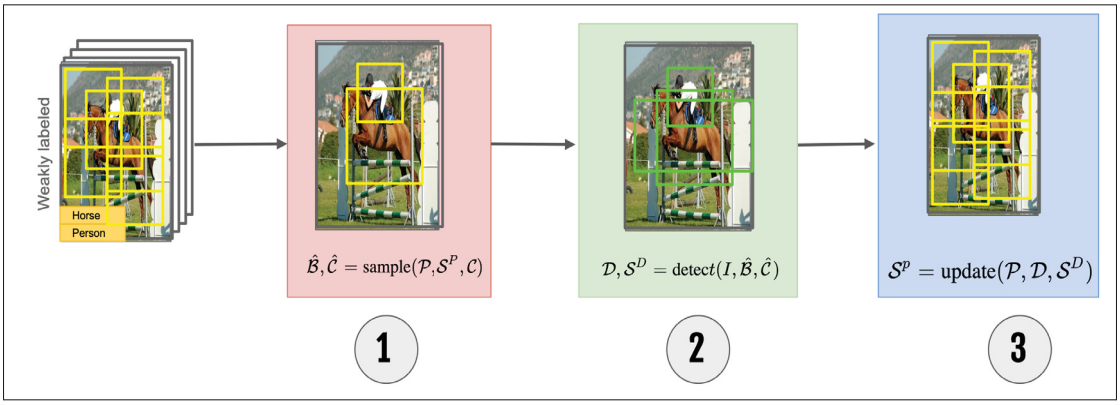


Figure 3.2 Sampling based WSOD training

The steps of one iteration in the training process are as follows:

(1) `sample`: For each proposal $p_i \in \mathcal{P}$ we have the corresponding classification score $s_{i,c}$ for a given class c. This score is accumulated based on the detector output via the score propagation process. For each given class present in an image, we consider the scores of all proposals $\mathcal{P}$, denoted by $\mathcal{S}^P$ and sample K boxes based on the multinomial distribution $\mathcal{M}$ with probabilities θ computed as in Eqn.3.3. The sampling step returns a set of boxes $\hat{\mathcal{B}}$ and corresponding labels $\hat{\mathcal{C}}$ which we consider as pseudo-GT in the subsequent `detect` step.

(2) `detect`: This can be performed with any detector. The detector takes as input the pseudo ground truth $(\hat{\mathcal{B}}, \hat{C})$ sampled by the sampler for an image I and returns detections $\mathcal{D}$ with associated scores $\mathcal{S}^D$ for all classes.

(3) `update`: The update step involves the score propagation process that updates the score values $\mathcal{S}^P$ of the object proposals $\mathcal{P}$ using equation 3.8.

3.2 Experiments with Sampling-based WSOD

We did extensive empirical studies to understand the strengths and limitations of our sampling-based WSOD. We used Faster RCNN (Ren *et al.*, 2015) as our backbone detector. The backbone is an ImageNet (Russakovsky *et al.*, 2015) pre-trained Vgg16 (Simonyan & Zisserman, 2015) network. The dataset used is Pascal VOC 2007 (Everingham *et al.*, 2010). Object proposals are extracted from the training images using selective search algorithm (van de Sande *et al.*, 2011). From an image, 2000 object proposals are extracted with a recall of approximately 92%. However, keeping so many proposals, means that we need to keep a large set of scores in memory. This will make the algorithm slow and more noisy at the beginning of the training as there are many possible regions to explore. In the experiments, we tested to use of a Class Activation Map (CAM) model (Selvaraju *et al.*, 2017) to reduce the number of proposals. Activation Maps using the gradCAM method are computed as shown in figure 3.3 as a preprocessing step. Then the selective search proposals overlapping with CAMs are subsampled to get a reduced set of more accurate proposals. In practice, for each image and for each class present in an image, we extract its CAM. Then, for each CAM region, only the proposals that overlap at least ρ with that CAM region are kept. The final set of proposals will be the union of the proposals selected for each class. This has resulted in approximately 500 proposals per image with a recall of approximately 88% (a reduction of 4% in the recall). CAM methods are used by some authors to refine the initial selective search proposals (Cheng, Yang, Gao, Guo & Han, 2020).

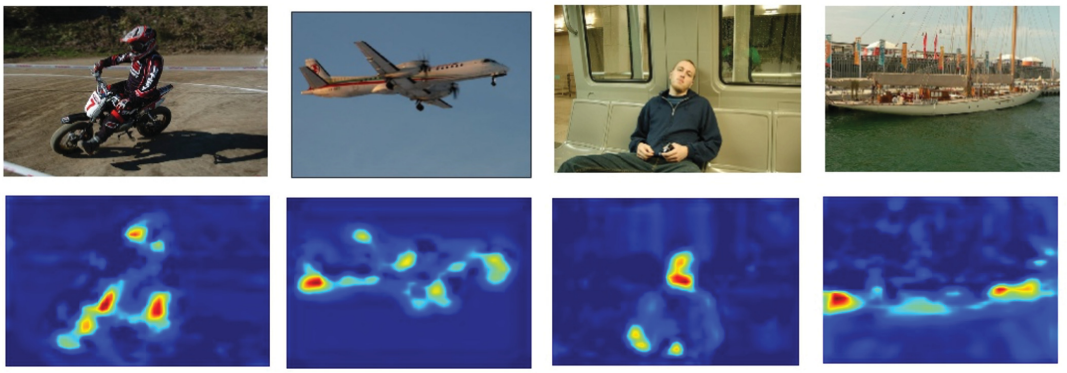


Figure 3.3 Activation maps obtained from gradCAM

3.2.1 Comparison with other methods

Table 3.1 shows the results of our sampling-based WSOD. It can be observed that importance sampling gives the best results regardless of the proposal type used. However, the gap is narrowed when CAM proposals are used. But our results are trailing when compared to customized popular weakly supervised architectures, except the base model WSDDN. As other weak detectors are built on top of the base model WSDDN, it is possible that additional acceleration is required in our case to make further improvements. So we tried to investigate the failure cases of our model.

We visualized the output detections from our model. Figure 3.4 shows the results. It can be observed that our model is producing many discriminative regions of the actual object as outputs. For example The faces of animals, and humans, the tires of vehicles, etc. Such discriminative regions will show a strong response to filters learned by a ConvNet in the classification settings (Zhou *et al.*, 2016). As the weakly supervised detectors are trained with classification labels, the problem of discriminative localization is ubiquitous in WSOD algorithms (Zhang *et al.*, 2021). Existing methods in the literature use context (Kantorov *et al.*, 2016), instance classifiers, (Tang *et al.*, 2017), entropy minimization (Wan *et al.*, 2018), etc., to cop up with discriminative localization.

Table 3.1 Results of the sampling-based WSOD with Vgg16 backbone

Settings	mAP
With selective search proposals (2000 per image)	
WSOD with importance sampling	26.72
WSOD with bandit sampler	14.99
With CAM refined proposals (500 per image)	
WSOD with importance sampling	34.91
WSOD with bandit sampler	27.21
WSDDN (Bilen & Vedaldi, 2016)	34.80
WCCN (Diba *et al.*, 2017)	42.80
OICR (Tang *et al.*, 2017)	41.20
PCL (Tang *et al.*, 2018a)	43.50
CASD (Huang *et al.*, 2020)	56.80
Fully supervised	69.90

Figure 3.4 Many wrong localizations of discriminative regions of an object

3.2.2 Issues in sampling when learning with weak labels

We conducted several studies to understand the issues faced by the sampler when training with weak image labels alone. These experiments revealed that the sampler is also facing the common issues of the WSODs. Particularly, the problem with discriminative object regions and the

inability to localize multiple instances of the same class properly. Below we present this with experimental evidence.

3.2.2.1 Proposals are sampled from discriminative regions

We also tried to understand the type of proposals sampled by our sampler. Figure 3.5 shows the plot from this study. Here "small proposals" indicate that the sampled object proposal is a discriminative area inside the ground-truth box. "No overlap" stands for the case when the sampled box has no overlap with the ground truth (this can happen from frequently co-occurring background with an object). Large proposals are those which overlap with the ground-truth boxes but are not enclosed inside (area-wise they are usually bigger than the object). It can be observed that as the training progresses, the sampler is sampling more discriminative small proposals around the object. We can also observe that the mAP is going down accordingly due to that. At the beginning of the training, large proposals are dominant probably because their scores get updated fast from the overlapping detection boxes. However, as the training progresses, discriminative object regions get more accumulated scores from the score propagation, so the sampler gets biased toward them.

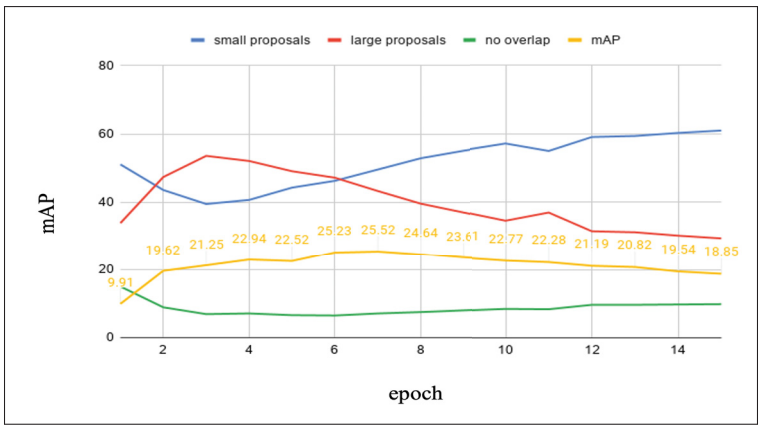


Figure 3.5 Type of proposals sampled over the training epochs

3.2.2.2 Problem with multiple instances of the same class

Apart from the discriminative region issue, the sampler also has issues when sampling in the presence of multiple instances of the same class present in an image. In such cases, the score propagation accumulates more score on one instance and that region gets sampled repeatedly. Figure 3.6 highlights this problem. As we can observe, though there are two cats in the image, the score of proposals for the one on the left is much higher than the one on the right. This results in the sampling process selecting more proposals around the white cat while completely ignoring the black cat.

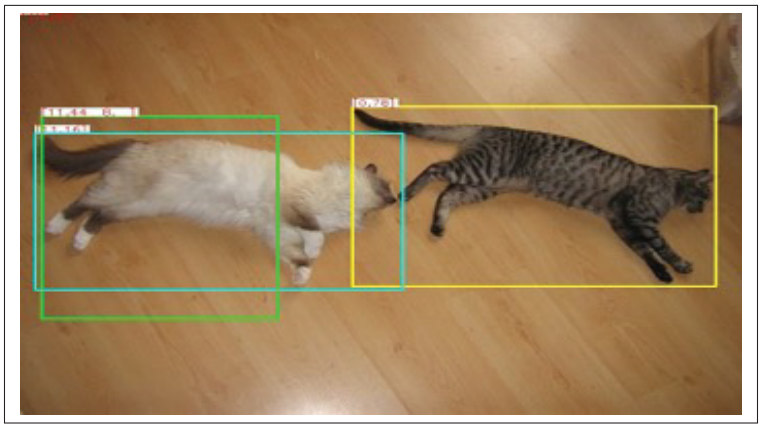


Figure 3.6 When multiple instances of the same class are present, one of them becomes dominant in the sampling process

The rationale for designing the Multi-Armed Bandit (MAB) sampler is this observation. As the MAB has a separate component for promoting exploration, the sampler may search more in the hypothesis space. In a controlled experiment with curated proposals, we observed that this is the case. Table 3.2 shows the results. There are two settings shown here. In the first setting, we assume there are 20 proposals per image. These proposals are obtained by adding ground-truth boxes of all instances and the remaining as random proposals extracted from the image using selective search. So this is a curated set of proposals offering a small search space (20 boxes). The second setting is our regular WSOD where 2000 object proposals are extracted from an image. With less number of proposals, the explore component pushes the MAB sampler to

Table 3.2 Comparison of multi-armed bandit and importance sampler with different number of proposals per image

Settings	mAP
With 20 proposals per image)	
WSOD with importance sampling	55.02
WSOD with bandit sampler	51.87
With 2000 proposals per image	
WSOD with importance sampling	26.72
WSOD with bandit sampler	14.99

explore the hypothesis space well and it quickly converges to the right locations by making the exploit score dominant. The importance sampler on the other hand is not exploring the search space well, so it misses many instances of the same class. With 2000 proposals per image, the MAB sampler keeps on exploring the huge search space and never converges. While the importance sampler finds some dominant instances for each given class and on average performs better than the MAB sampler.

Following the common practices in WSOD, we could use techniques like context information (Kantorov *et al.*, 2016), instance classifiers (Tang *et al.*, 2017), entropy minimization (Wan *et al.*, 2018) etc to minimize the localization difficulty. However, such detectors are difficult to train as they involve many hyperparameters and stages of training. We can see that the result of the best method using such techniques in table 3.1 is still far from the fully supervised upper bound. So we took an alternate route. We tried to investigate whether by using a few labeled images we improved the detector. It is practically feasible to get a few annotated images in most of the object detection settings. This gave rise to our semi-weakly supervised detector explained in the next section.

3.3 Semi-weakly Supervised Object Detection

Fig 3.7 illustrates the overall system design of our semi-weakly supervised detector. For every input image, the detector is employed in a different way, depending on the available annotation

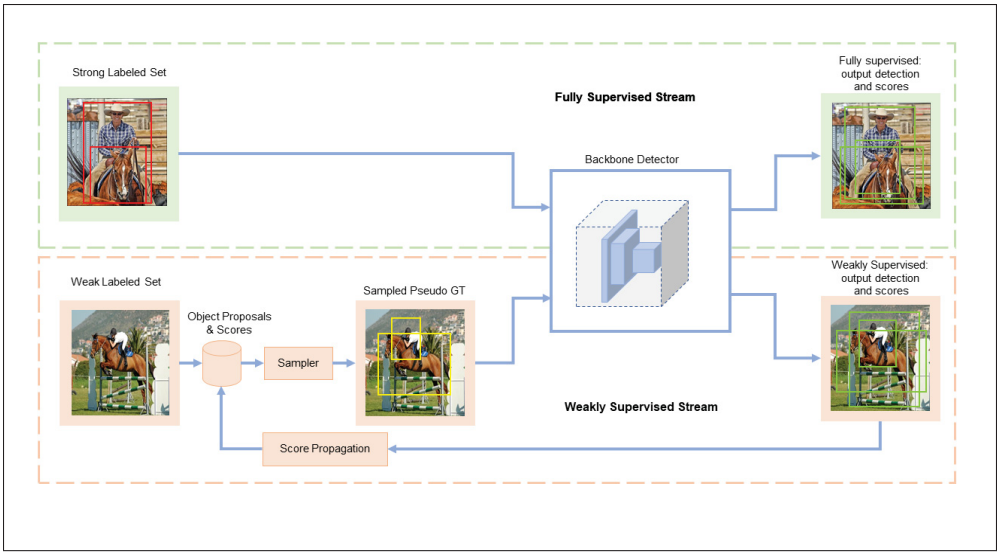


Figure 3.7 Proposed method for semi-weakly supervised object detection

level. For the fully-labeled images, we perform a normal forward-backward cycle by taking the real GT annotations provided. For the weakly-labeled images, we use our sampling approach to select the most likely bonding boxes for each class in the image, to use them for training as a pseudo ground-truth. This allows us to use the same detector employed for the fully-labeled images and also for the weakly-labeled ones. When combining weakly and fully supervised learning, we need to determine the right importance to associate with the two learning tasks. For doing that, we rely on a hyper-parameter that defines the sampling ratio between the fully- and weakly-labeled images. The rest of this section introduces the learning steps of both strongly and weakly annotated categories. Then, we present how their sampling ratio is applied within the training process.

3.3.1 Learning with strong annotations

For the images that are strongly annotated (.i.e bounding box annotation for each object present), the learning step is straightforward. Given an input image I, the ground truth (GT) annotations are defined with the bounding box positions $\mathcal{B} = \{b_0, b_1, \cdots b_N\}$ and corresponding classes $C = \{c_0, c_1, \cdots c_N\}$. $b = (x_0, y_0, x_1, y_1)$ is a vector with 4 values that represent for instance the

top left and bottom right corner of a box, while $c \in C$ is a discrete value that represents the object category. In our experiments we use faster RCNN as detector Ren *et al.* (2015), and thus use a loss as:

$$L_F = \sum_{j \in \mathcal{D}} \sum_{i \in \mathcal{B}} \frac{1}{N_{cls}} L_{cls}(f_{c_i}(d_j, I)) + \lambda \frac{1}{N_{reg}} L_{reg}(c_i, d_j, b_i), \tag{3.9}$$

For each GT bounding box b_i it generates a loss based on the scores f_{c_i} and overlap of the obtained detections d_j. L_{cls} and L_{reg} denote the classification and localization loss, respectively. N_{cls} and N_{reg} are the normalization factors that depend on the number of foreground and background RoIs considered. λ is a hyperparameter that controls the relative importance of the classification and localization loss. Note that the exact form of loss can vary according to the fully supervised detector architecture used in the model, but our approach is independent of the specific fully supervised loss.

3.3.2 Learning with weak annotations

The weakly supervised stream works exactly as explained before. In case of weak supervision, we know the object classes that are present in the image c, but not the bounding box locations b_i. To this end, we used our sampling approach to get bounding box locations of categories present in the image. For sampling, we used the importance sampler; the score propagation is used as before. The sampled boxes and their object classes are then used as the pseudo-label for the weakly labeled images. The weakly labeled image along with its pseudo label ($\hat{\mathcal{B}}$ and $\hat{C}$) is fed to the detector and the loss is computed the same way as in equation 3.9 but with pseudo labels.

$$L_W = \sum_{j \in \mathcal{D}} \sum_{i \in \hat{\mathcal{B}}} \frac{1}{N_{cls}} L_{cls}(f_{c_i}(d_j, I)) + \lambda \frac{1}{N_{reg}} L_{reg}(\hat{c}_i, d_j, \hat{b}_i), \tag{3.10}$$

While the fully supervised object detector uses ground truth boxes that are correct, the weakly supervised counterpart estimates the object box location during training, and therefore the estimation can be noisy. Thus, when learning with strong and weak labels we might want to set

a hyper-parameter value that balances the relative importance of the two losses. In this case the final loss is $L = L_F + \lambda L_W$. To control the relative importance of the losses, we expressed the weight λ with a sampling ratio for the input data instead. Specifically, we use a ratio parameter r that controls the amount of training data from the fully and weakly labeled pool of data. For instance, $r = 0.6$ means that 60% of the data is from the pool of the fully-labeled samples and 40% from the weakly-labeled samples in a minibatch. With this design, we can feed both the fully annotated and weakly annotated images in parallel to the model and train it in a single stage.

3.3.3 Learning algorithm

The proposed learning algorithm with two streams of annotated data is summarized in Algorithm 10. For the sake of simplicity, the algorithm is shown for the case of a single image I, but it could be trivially extended to a batch of images.

For supervised samples, our algorithm uses directly the bounding box annotations $\mathcal{B}$ and the corresponding classes C for inference (`detect`). For weak supervision, the inference is performed on pseudo-GT annotations $(\hat{\mathcal{B}}, \hat{C})$ that are obtained by sampling object proposals (`sample`). Then, the obtained detections $\mathcal{D}$ and scores $\mathcal{S}^D$ are used to `update` the proposal scores ($\mathcal{S}^P$). In both cases, the obtained detections $\mathcal{D}$ and scores $\mathcal{S}^D$ are used to compute the loss L and update the recognition model (`backprop`).

The `sample`, `detect`, and `update` steps are the same as the weakly supervised detection.

3.4 Experiments with Semi-WSOD

The empirical studies are conducted on Pascal VOC 2007 and 2012 (Everingham *et al.*, 2010). Particularly, VOC 2007 is used as the fully labeled set (5011 images) and VOC 2012 as the weakly labeled set (17125 images). Images are sampled randomly to create fully annotated and weakly annotated splits. For evaluation, the VOC 2007 test set is used (4952 images). The standard VOC AP metric (AP 50) is used to measure the performance of the model. The network is trained

Algorithm 3.1 Semi-Weakly supervised learning with Pseudo GT

Input: Image: I, GT:$(\mathcal{B}, C)$ proposals and scores: $(\mathcal{P}, \mathcal{S}^P)$
if $\mathcal{B} \neq \emptyset$ **then**
    fully supervised ;
    $\mathcal{D}, \mathcal{S}^D$ = `detect`(I, $\mathcal{B}, C$);
else
    weakly supervised;
    $\hat{\mathcal{B}}, \hat{C}$ = `sample` $(\mathcal{P}, \mathcal{S}^P, C)$;
    $\mathcal{D}, \mathcal{S}^D$ = `detect` (I, $\hat{\mathcal{B}}, \hat{C}$) ;
    $\mathcal{S}^P$ = `update` $(\mathcal{P}, \mathcal{D}, \mathcal{S}^D)$;
end if
`backprop`(I , $\mathcal{B}, C, \mathcal{D}, \mathcal{S}^D$)

end-to-end using stochastic gradient descent(SGD) with a momentum of 0.9 and a weight decay of 0.0005. The initial learning rate is set to 1e-2 and decayed at epochs [5,10] by a factor of 10. We trained the model for 20 epochs with a batch size of 8. The temperature parameter T for the multinomial distribution used for sampling is set to 2.5. From an image for each class present, we sample $K = 5$ object proposals as pseudo-GT during training. During training, the shorter edges of input images are randomly re-scaled within {480, 576, 688, 864, 1200} to introduce an augmentation in the image scale. Random horizontal flipping is also used. Object proposals are extracted using the selective search algorithm (van de Sande *et al.*, 2011) and then refined using CAM (Selvaraju *et al.*, 2017). Images are normalized with mean = $[0.485, 0.456, 0.406]$ and std = $[0.229, 0.224, 0.225]$, as in ImageNet training (Russakovsky *et al.*, 2015). The network is trained on an NVIDIA V100 GPU with 32GB memory.

3.4.1 Comparison with state-of-the-art methods

Table 3.3 shows the comparison of our method with state-of-the-art methods for semi- and weakly-supervised learning of object detectors (with ResNet50 backbone (He *et al.*, 2016)). We first evaluate our method for semi-weakly supervised training. The only other method performing Semi-weakly supervised learning is WSSOD (Fang *et al.*, 2021). In this setting, our method outperforms it, while being also more flexible (our method is not detector-specific). Compared

Table 3.3 mAP performance of state-of-art methods on VOC 2007 test set. Models are trained using VOC 2007 as the fully annotated set, and VOC 2012 as the weakly annotated set

Method	**AP 50**	**AP**
Fully Supervised VOC07 *(lower bound)*	74.4	-
Semi-Weakly-Supervised VOC07 (fully) + VOC12 (weakly)		
WSSOD (Fang *et al.*, 2021), ArXiv 2021	78.9	-
Ours	79.4	47.3
Semi-Supervised VOC07 (fully) + VOC12 (unsup.)		
CSD (Jeong *et al.*, 2019), NeurIPS 2019	74.7	42.7
STAC (Sohn *et al.*, 2020), ArXiv 2020	77.4	44.6
WSSOD (Fang *et al.*, 2021), ArXiv 2021	78.0	-
ISD (Jeong, Verma, Hyun, Kannala & Kwak, 2021), CVPR 2021	74.4	-
Ours	77.8	44.2
Fully Supervised VOC07+VOC12 *(upper bound)*	80.9	-

to the model trained only on VOC 2007 with full supervision (*lower bound*), we observed a significant improvement (5.0%) when using additional weak labeled data, approaching a model with full annotations in both datasets (*upper bound*). Figure 3.8 shows a qualitative evaluation of our model's detection results where we visualized the confident predictions.

We then compare our model to the state-of-the-art in the normal semi-supervised settings where unlabeled data is provided without any labels. To report the results of our method in semi-supervised settings, we trained a classifier on the available fully labeled images and used that classifier to obtain image-level labels for the unlabeled images. This requires the training of an additional classifier, but it is less expensive than the detector pre-training used in most of the semi-supervised methods. Results indicate a significant improvement in terms of performance, with the additional moderate cost of collecting weak image-level labels. In the semi-supervised case also, our method shows an improvement of 3.4%, outperforming most of the methods in terms of mean average precision with an IOU threshold at 0.5 (AP 50) and mean average precision averaged over several IOU: 0.5- 0.9 (AP).

Figure 3.8 Visualization of detection results of our Semi-WSOD model

3.4.2 Ablation studies

Several ablation studies are conducted in order to assess the individual components of our proposed model. First, we study the working of the sampler and score propagation modules. Different methods for doing score propagation are studied. Then, we analyze the performance of our method with varying degrees of supervision (changing the number of supervised data points). Finally, we study different types of errors made by the model. All of these studies are conducted on PASCAL VOC 2007 by training the model using its trainval (training and validation images combined) set and testing on its test set. We use 10% annotated images in this analysis, while the rest of the images are weakly annotated.

3.4.2.1 Sampler and score propagation

To understand the sampling progress, we analyzed the sampled proposals over the training epochs. Figure 3.9 illustrates an example sampling process during the training phase for the person category. It can be observed that, though in the beginning, we sample pseudo-GT boxes randomly from the image, it converges to meaningful locations for the person category in the later stages. This also shows the exploration and exploitation phases during the course of training.

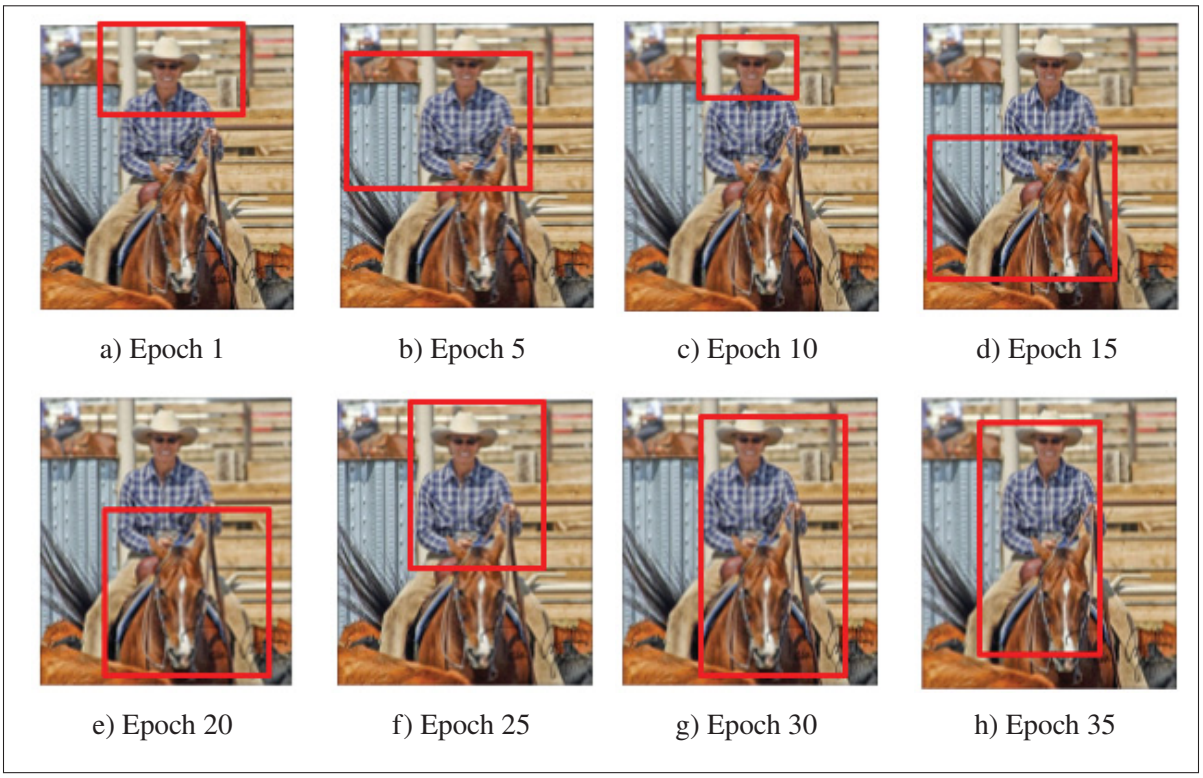


Figure 3.9 Evolution of the Pseudo GT sampling. While in the first iterations of the training, bounding boxes are samples almost randomly (exploration), after some training, the algorithm learns to sample only from meaningful locations (exploitation)

To understand whether the sampler is learning a meaningful object location, we analyze the heatmap produced by the score distributions of the object proposals for a given class. To obtain the heatmap, for each pixel location, the scores from all object proposals covering that pixel are added, and then normalized by the number of object proposals covering that pixel. Fig. 3.10 shows some examples of heatmaps. It can be observed that active regions of heatmaps correlate

well with object locations, and hence the sampler is finding meaningful semantic information through sampling and score propagation. We also notice that for small objects (ducks on the top right image) or objects with a recurrent background (train), the sampler selects not only the object of interest but also some background. However, this is a common problem of all weakly supervised approaches.

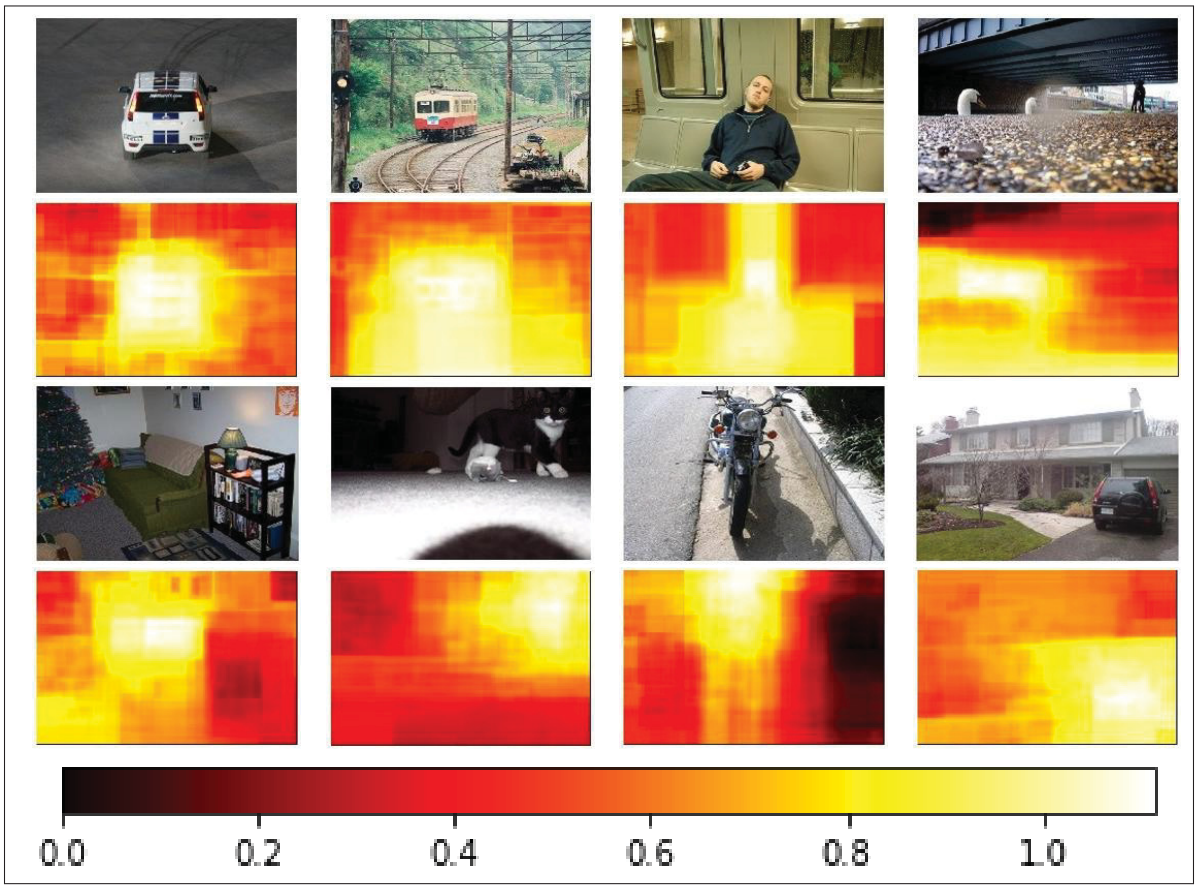


Figure 3.10 Heatmaps of sampler scores for images belonging to different categories from the Pascal VOC dataset

Next, we will ablate the score propagation component. Score propagation can be from all detection boxes or a selected set of boxes matching some quality criteria. We considered 3 settings: (1) score propagation from all detection boxes, (2) from the maximum overlapping boxes, and (3) from the maximum overlapping boxes when the IOU overlap is above a threshold t. We found that $t = 0.3$ provides the best performance. Table 3.4 summarizes the results from this

Table 3.4 Analysis of score propagation strategies. mAP performance is measured on a weakly semi-supervised model using 10% full annotations and remaining weakly-labeled images on the VOC 2007 dataset

Score Propagation Strategy	mAP
Propagate from all boxes	57.2
Propagate from max-overlapping boxes	58.3
Propagate from max-overlapping boxes when IOU > t	60.3

study on the VOC 2007 dataset. The model is trained using different 10% splits on its trainval set, and evaluated on the test set. It can be observed that propagating scores from the maximum overlapping detection box of each proposal provide the highest mAP accuracy. When the overlap is above a threshold t imposes more quality constraints for score propagation, and improves results. Score propagation from all detection boxes does not perform well, although it can provide a smoother update to the object proposal scores. This may be due to the concentration of the high scores over large proposals when all detection boxes are propagating their scores. This results in an incorrect sampling of over-sized proposals, especially for smaller objects.

3.4.2.2 Impact of fully annotated images

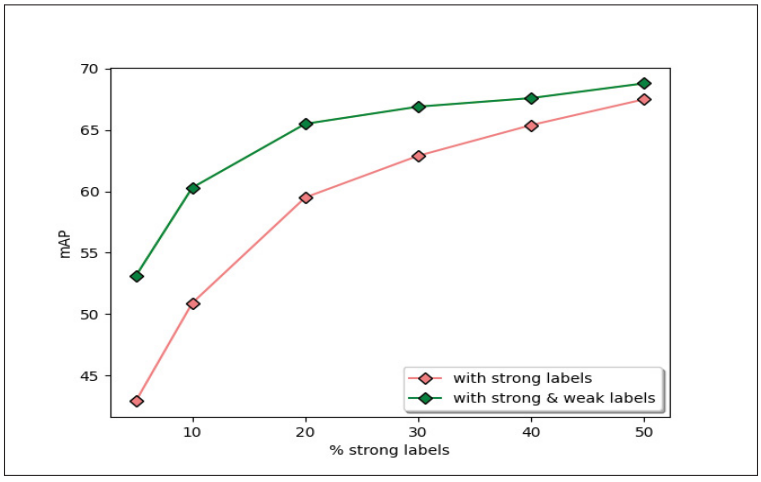


Figure 3.11 Change in mAP with varying amounts of fully annotated images during training on the VOC 2007 dataset

Table 3.5 Impact on mAP performance of the ratio for fully to weakly-annotated images

Settings	mAP
10% fully annotated images	50.9
10% fully annotated and remaining weakly annotated images (without ratio balancing)	47.5
10% fully annotated and remaining weakly annotated images (with ratio balancing)	60.3

In Figure 3.11 we compare the performance of a detector baseline trained only with strong labels (red line) with a model trained with strong and weak labels using our sampling approach (green line). As expected, the gain of our model is more significant when the amount of strong labels is reduced. For instance, with 5% of strong labels, our model improves over the baseline by 10 points. It is also reaching the best weakly supervised model's performance (Huang *et al.*, 2020) with a very easy training process using 250 labeled images. When increasing the percentage of strong labels, the gain reduces. This experiment shows how our approach is useful when the amount of fully labeled data is limited. In this setting, our model can approach the performance of a fully supervised model, but with much fewer annotations.

3.4.2.3 Impact of the ratio parameter

We analyze the importance of the ratio parameter r for balancing the number of fully and weakly annotated training images. In Table 3.5, it can be observed that without this balancing term, the detection performance is even worse than the settings where only annotated images are used. With the proper ratio balancing ($r = 0.7$), the mAP performance of the detector significantly outperforms the baseline using only fully supervised images. Thus by tuning this ratio parameter, we can effectively leverage the large pool of weakly annotated images. One of the appealing properties of this strategy is that it does not require any change to the model architecture or loss function.

Table 3.6 Impact on mAP performance of using proposals filtered by a CAM method

% of images with bounding box annotations	mAP without CAM proposals	mAP with CAM proposals
0%	26.7	34.9
5%	48.4	53.1
10%	57.6	60.3
20%	64.6	65.5

3.4.2.4 Impact of CAM proposals

Table 3.6 shows the impact on performance when subsampling object proposals based on their overlap with class activation maps during sampling. The CAM is obtained by training a vgg16 (Simonyan & Zisserman, 2015) network on the multi-label VOC 2007 image-level labels. Then the overlap of selective search proposals (van de Sande *et al.*, 2011) to the CAM of all classes present in the image is computed. Based on the overlap, the object proposals without sufficient overlap to the CAM, which are perhaps from the image background region, are ignored. This results in a slight loss of recall, but an improvement in terms of the mAP, especially with few fully annotated images, due to the reduction of noisy proposal regions that could misguide the sampler. In practice, we used an overlap threshold of 0.1 which resulted in a 5% reduction of recall, but the average number of proposals is reduced 4 times to approximately 500 object proposals per image. From table 3.6, it is clear that filtering noisy proposals using CAM brings improvement in mAP. However, the impact of the CAM proposals reduces with the availability of more fully annotated images. This is according to the general facts that with more annotations, the appearance model will be more accurate and hence, the model itself will be powerful enough to better distinguish objects.

3.4.3 Type of errors the model is making

The distribution of the error of our model is also analyzed using the TIDE (Bolya, Foley, Hays & Hoffman, 2020) evaluation tool (see Figure 3.12). It can be observed that the localization

Table 3.7 Results comparison on the COCO dataset with different percentages of labeled images.

Settings	**1%**	**5%**	**10%**
Supervised	11.6	18.7	23.8
STAC (Sohn *et al.*, 2020)	14.0	24.4	28.6
Soft-teacher (Xu *et al.*, 2021)	20.5	30.7	34.0
Ours	15.4	22.9	24.3

error contributes the most toward the overall errors made by our detection model. This is expected, since there is a large fraction of images without bounding box labels, so the objectness distilled from a small fraction of fully annotated images is insufficient to capture large variations in appearance. Missed ground truth is the next major error with our model. This is mainly the consequence of the exploration capacity of the sampler. Once some dominant object regions start providing higher scores from the score propagation, the sampler can miss other difficult instances, especially smaller objects. Thus, our sampler will not sample candidate proposals from those regions, and they remain undetected. Frequently co-occurring background regions are also challenging for the sampler since such regions can also accumulate higher scores over time from the score propagation block. Those regions might also be sampled many times, resulting in detection boxes in background regions.

3.4.4 Limitations

While using a few labeled images along with weakly labeled images helped us to improve the localization difficulty of a purely weakly supervised detector, there are still challenges to overcome. It was evident when our method was studied on the MS-COCO (Lin *et al.*, 2014) dataset. Table 3.7 shows the results from this study. We compared our model with the semi-supervised methods where annotations are provided for 1%, 5%, and 10% of the available training set images.

Compared to STAC (Sohn *et al.*, 2020), our method is better only in the 1% case. But the Soft-teacher method (Xu *et al.*, 2021) outperforms us in all benchmarks. Both STAC and

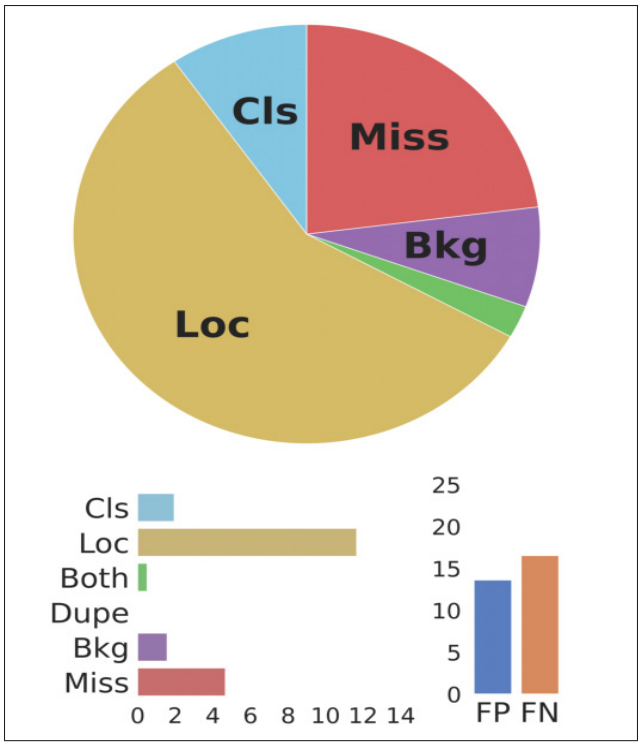


Figure 3.12 **Evaluation of performance loss.** TIDE Evaluation of detection results. Error types are: **Cls**: localized correctly but classified incorrectly, **Loc**: classified correctly but localized incorrectly, **Both**: both cls and loc error, **Dupe**: duplicate detection error, **Bkg**: detected background as foreground, **Miss**: missed ground truth error

Soft-teacher are using mean-teacher (Tarvainen & Valpola, 2018) approach. The key difference between their approach to ours is in the pseudo-label box location. In our case, the pseudo labels are sampled from a pre-computed set of object proposals, so the location of pseudo labels remains fixed throughout the training. With the mean-teacher method, the pseudo-ground-truth boxes are provided by a teacher network that is continuously improving as the training progresses. The mean-teacher network learns to produce pseudo boxes of varying sizes improving the recall. Our pseudo boxes are fixed. While the Pascal VOC dataset has bigger objects with less scale variance, for MS-COCO the scale variation is high and contains many small objects. The pre-computed proposals are not meeting these quality requirements. Their recall is close to 75%

Table 3.8 Comparison of other weakly supervised methods with ours with 1% labels on the COCO dataset.

Settings	**COCO AP**
MELM (Wan *et al.*, 2018)	7.8
Instance-Aware WSOD (Ren *et al.*, 2020)	12.6
CASD (Huang *et al.*, 2020)	13.9
Ours(with 1% labels)	15.4

and they mostly miss the small objects. This explains the reason for the performance drop of our method in the MS-COCO dataset.

We also tried to understand how well our method favors the weakly supervised detectors on the MS-COCO dataset. Table 3.8 shows the comparison. In general, the weakly supervised detectors are not performing well on the COCO dataset. The reason is as explained before, they rely on the object proposals from selective search (van de Sande *et al.*, 2011) or edge box (Zitnick & Dollar, 2014) method which has a low recall on datasets with high scale variation. With 1% labeled images, our approach however performs better than all the WSODs while still using an off-the-shelf Faster RCNN detector with a large collection of weakly labeled images. This alleviates the need for customized architectures to train on weakly labeled images.

3.5 Conclusion

We proposed a sampling-based learning method to train off-the-shelf detectors with weak image-level labels. It can be trained with weak image-level labels alone or with a mix of weak and strong labels (with exact bounding box locations). While the method works fine with weak labels alone with a performance comparable to the baseline models of existing WSODs, the best utility is when combined with a small amount of fully annotated images. When combined with a small fraction of labeled images, we get the best results among other methods using weak and strong labels. Compared to the recent WSODs with customized architecture, multiple stages of training, and complicated training objectives, our method offers a flexible alternative where an off-the-shelf detector can be directly used and trained in a single stage just like the standard

fully supervised training. With a few annotated images, our results significantly outperform existing WSODs on all datasets. Using single-stage learning, our method effectively makes use of the images with only weak image-level labels by sampling pseudo-GT boxes from the object proposals extracted in that image. The scores for sampling proposals are obtained via the proposed score propagation mechanism.

Experiments on the MS-COCO reveal the need to refine pseudo-bounding boxes during training. Our method with a fixed set of pre-computed object proposals fails to localize small objects properly. With the modern end-to-end mean-teacher methods, object proposals are improving during the training and they localize the small objects better. In the next chapter, we will be dealing with a practical application setting of object detection where small objects dominate the input image. We will explore the capability of the mean-teacher detector in that setting and propose techniques for improving small object detection.

CHAPTER 4

DENSITY CROP-GUIDED SEMI-SUPERVISED DETECTION FOR AERIAL IMAGES

In this chapter, we consider the problem setting of object detection from high-resolution aerial images. Typically in aerial images, the objects are very small as they tend to cover a large area from the top (Cheng *et al.*, 2023). On top of that, the small objects appear in clusters increasing the localization difficulty further (Yang *et al.*, 2019). The images usually have a very high pixel resolution (2k to 16k in the popular benchmark datasets). Thus learning with fewer bounding box annotations is practically very demanding in aerial image object detection. Creating bounding box annotations for tiny objects is even more difficult than the natural images of Pascal VOC or MS-COCO. Unlike natural images, sometimes domain expertise is also required to annotate object classes with fine-grained differences (eg: building types, vegetation types). Thus there are chances for ambiguity even in providing weak image-level labels. Considering these practical difficulties, we attempt to use semi-supervised detection for the problem settings in this chapter so that unlabeled data can be supplied without any processing or labeling.

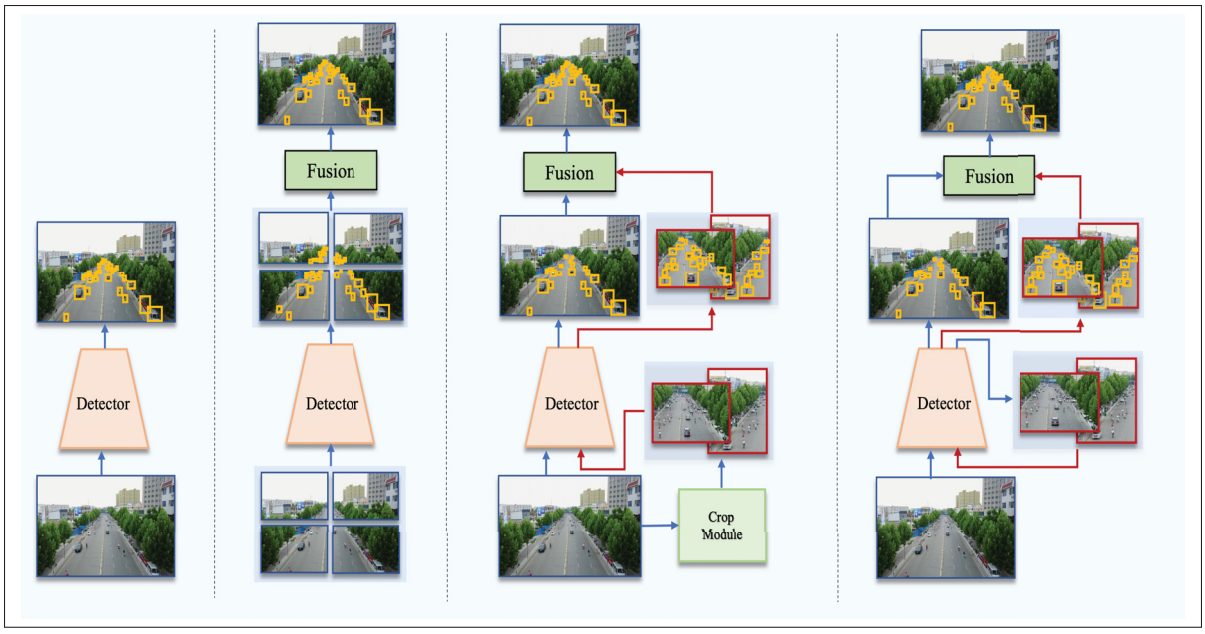
The success of deep learning based object detection methods on natural images (Lin *et al.*, 2017b; Carion *et al.*, 2020; Cai & Vasconcelos, 2018; Tian *et al.*, 2019; Ren *et al.*, 2015; Redmon & Farhadi, 2017), has resulted in a fast growth of their adoption to many downstream applications, including aerial image detection from drones or satellites, for earth monitoring, surveillance, inspection, etc (Lacoste *et al.*, 2021; Xia *et al.*, 2018; Cheng, Zhou & Han, 2016; Han, Ding, Xue & Xia, 2021; Long, Gong, Xiao & Liu, 2017). However, unlike natural images in the Pascal VOC (Everingham *et al.*, 2010) and MS-COCO (Lin *et al.*, 2014) datasets, aerial images are captured in high pixel resolution and are typically comprised of many small objects, that are sparsely distributed in crowded object regions. As a comparison, the average number of objects in Pascal VOC and MS-COCO images are 3 and 7, respectively, whereas images in the VisDrone (Zhu *et al.*, 2018) and DOTA (Xia *et al.*, 2018) datasets – two popular benchmarks in the aerial detection community – have an average number of 53 and 67 objects, respectively. The average width of Pascal VOC and MS-COCO images are 500 and 640 pixels, respectively, while

the same in VisDrone and DOTA images are 1500 and 4000 pixels, respectively. Therefore, improvements observed in object detection methods applied to natural images do not easily translate to object detection in high-resolution images from drones and satellites. Though Semi-supervised Object Detection(SSOD) has achieved tremendous progress in recent years on natural images (Guo *et al.*, 2022; Li *et al.*, 2022a; Liu *et al.*, 2021a; Xu *et al.*, 2021; Tang *et al.*, 2021; Jeong *et al.*, 2019; Sohn *et al.*, 2020; Meethal *et al.*, 2022), we are yet to see large-scale adoption of them on aerial images. We hypothesize that the above mentioned difficulties contributed to the lower adoption rate of semi-supervised detectors in aerial image detection. In this chapter, we will investigate methods to do semi-supervised object detection in this imagery considering the challenges particular to the aerial imagery.

4.1 Density Crops for Small Object Detection

As we have seen, the mean-teacher semi-supervised detection works with a single backbone detector whose target for the unlabeled data can be obtained by the pseudo-labeling approach. But existing methods perform small object detection from high-resolution aerial images by cropping clustered object regions with the help of an external "crop module" as shown in the figure 4.1 (c). For example, Yang *et al.* (2019) proposed to use a separate network called ClusNet to extract clustered object regions. Li *et al.* (2020) employed a density generation module to identify the dense regions. Though such density-based methods work in a purely supervised setting, it is not immediately clear how they can be trained in a semi-supervised setting like the mean-teacher framework. The pseudo ground-truths for the "crop module" on unlabeled images are difficult to obtain. Moreover the "crop module" is often trained in a separate stage before the detector training. This multi-stage training is also difficult to translate to the mean-teacher network where the training is end-to-end.

Note that the uniform cropping shown in figure 4.1 (b) is also a widely used technique for dealing with the scale issue of small objects in aerial images. In uniform cropping, the input image is cropped into uniform patches, and then object detection is performed on these patches in high resolution by upsampling. Although these uniform patches help to improve the accuracy,



(a) Baseline (b) Uniform crops (c) Density crops (d) Our approach

Figure 4.1 objects in high-resolution aerial images. (a) The image is down-scaled and processed at the detector's input size. (b) The image is split into uniform, possibly overlapping patches, and each patch is processed by the detector. (c) An external learnable module crops the image into dense object regions. Each crop is re-scaled and processed at the detector's input size. (d) Our proposed CZ detector is re-purposed to detect the density crops along with the base class objects, eliminating the need for an external module. Each crop is re-scaled and processed at the detector's input size in a second stage of inference. Blue arrows show the path of the original image and red shows the path of density crops

this approach does not respect the distribution of the objects in the image, and hence the scale normalization achieved is not optimal (Yang *et al.*, 2019; Li *et al.*, 2020). As the objects in aerial images usually appear crowded in sparsely distributed regions of the image, density-based cropping usually yields better results than uniform cropping. Since it is difficult to input high-resolution aerial images directly to a detector due to the computational cost and large memory footprint, they are often resized to the standard input size range of 300-512 pixels(see Fig. 4.1 (a)). This rescaling, coupled with the feature down-sampling in ConvNets, often results in feature representations linked to small objects diminished or corrupted by the noisy background activations (Yang *et al.*, 2022).

While uniform cropping is still inferior to density-based crops, practitioners still use it widely due to the simplicity it offers. Existing density-based approaches rely on an external crop module and more parameters to train whereas the uniform crops can be wrapped on top of a standard object detector. Also, many approaches resort to multi-stage training where density cropping modules are trained first. Even with single-stage end-to-end methods, the crops obtained are noisy in the beginning and are only useful for aiding small object detection in the later stages of the training (Yang *et al.*, 2019).

To get the best out of the density-based cropping and the practical simplicity of the uniform cropping approach, we designed a density-based detection approach with the detector itself as shown in figure 4.1 (d). The detector identifies the base class objects and density crops from the image. It then zoom-in on the density crops by detecting small objects on the up-scaled crops from there. The detection from the crops and the original image are later merged. We call our detector with the zoom-in capability a Cascade Zoom-in (CZ) Detector. As the entire process is wrapped on top of a standard detector, the mean-teacher framework can be easily applied here. We simply make use of the detector itself to discover the density crops, by adding the "crop" as a new class to the detector. The crops are labeled as a pre-processing step using a crop labeling algorithm, and hence the detector receives a consistent signal of what constitutes a crop during training. During inference, while other methods require complex post-processing to filter the noisy crops, we can simply perform it based on the confidence of the "crop" class from the detector.

4.2 Cascaded Zoom-in Detector

Figure 4.2 illustrates the training and testing of our CZ Detector. First, the density crops are extracted from each training image as a pre-processing step, using our crop labeling algorithm. These density crops are added as a new class to be detected in the corresponding image. Then we augment the training set with the higher resolution version of the density crops, and the corresponding ground truth (GT) boxes of objects inside the crop. Then, the detector is trained as usual. This training process has an almost negligible overhead over standard detector training,

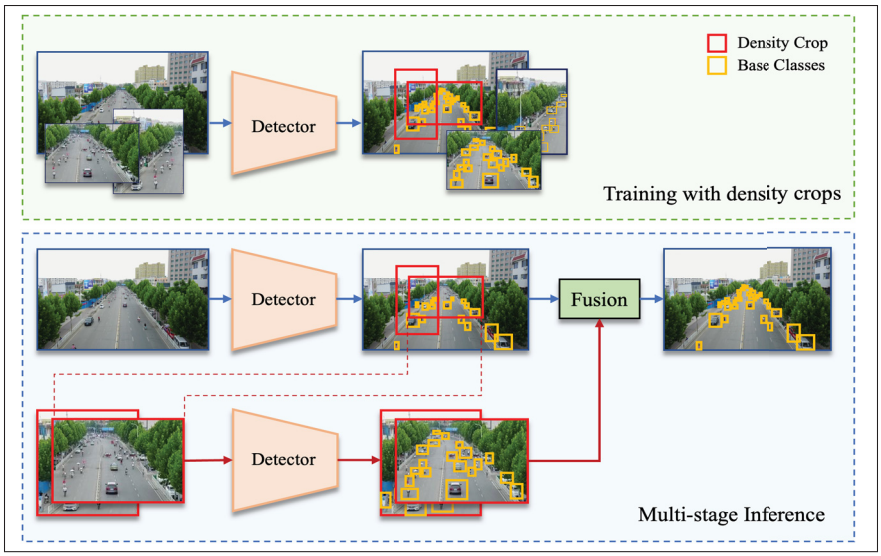


Figure 4.2 Overview of our proposed Cascaded Zoom-in detector. During training (top), density crops are extracted, and labeled as a new class (red boxes) on the original image. The training set is augmented with the rescaled density crops and the corresponding ground truth boxes within these crops. During the first stage of inference (bottom), the base class objects and density crops (red boxes) are detected on the whole image. In the second stage, the density crops are rescaled to a common larger size, and a second inference is performed. Finally, the detections on density crops are combined with the detections on the whole image

and it is similar to that of uniform crop-based training. The inference is performed in two stages. In the first stage, the base class objects and density crops are detected from each input image. In the second stage, high-quality density crops are selected based on their confidence score, and another inference is performed on an up-sampled version of these crops. Finally, the detections from stages one and two are fused to get the output detection. Compared to standard object detector learning, the extra work required at training time is the crop labeling which can be performed as a pre-processing step. While making predictions, the extra work required is one more inference. As both of these processes don't require any significant modification on a normal object detection pipeline, similar to uniform crops, our method can be easily incorporated for accelerating small object detection.

We will now dive into how to transform any detector into a cascaded zoom-in (CZ) object detector. Let us consider the *original image*, which is kept at its high image resolution, the *down-sampled image*, which is an image containing the same view of the original, but down-scaled to the detector input size, and the *cropped images*, which are the selected regions of the image that are up-scaled to the detector input size. First, we will present the crop labeling algorithm that labels the crowded object regions as "density crops" and augments the training data by adding up-scaled versions of those regions. Then, we will look into the two-step inference procedure shown above.

4.2.1 Training with density crops

In order to use a standard detector for our approach, we need to add a new class that we call "density crop" to the training annotations. In this way, our approach is detector agnostic (as we don't change the internals of the detector, we just add one additional class to the list of target classes) and does not require any additional component than the detector itself. The density crop class should label those parts of the image that contain many small objects and include them in a bounding box. This will allow training and inference to focus on those parts by analyzing them in higher image resolution. Several different ways could be considered for defining the density crop. The quality constraints we used to define density crops are: (i) they should enclose groups of small target objects, (ii) they are easy to localize at inference time, and (iii) they are optimal in number to reduce the computational cost.

We note that existing methods leveraging density crops are computing the density crops on the fly during the detector training (Yang *et al.*, 2019; Duan *et al.*, 2021; Li *et al.*, 2020). They predict density maps and synthesize the crops from them by post-processing. But recently it has been shown that such prediction-based label assignments are error-prone (Zhou *et al.*, 2022). Instead selecting a max-size object proposal that will enclose the object very likely and give a more consistent signal during the course of training works better. Zhou *et al.* (2022) used the biggest object proposal from the RPN for label assignment. Their observations are shown in figure 4.3. The prediction-based labels are changing a lot during the course of training (top row). Also,

they may produce boxes that cover objects partially as well. Whereas the max-size box gives a consistent signal and always encloses the objects in its boundary (bottom row). In our case, the crops are evolving as the training of the density module is progressing. Also, the density maps may not cover the entire clusters as we do thresholding and find the regions with maximum activation which might lose the boundaries of the clusters. Based on these observations, we decided to label the density crops apriori so they stay consistent during the training. Also, the crop labeling algorithm produces crops that enclose the cluster of objects in a much bigger box. This is similar to the max-size box, so we have no issue with partial covering as well.

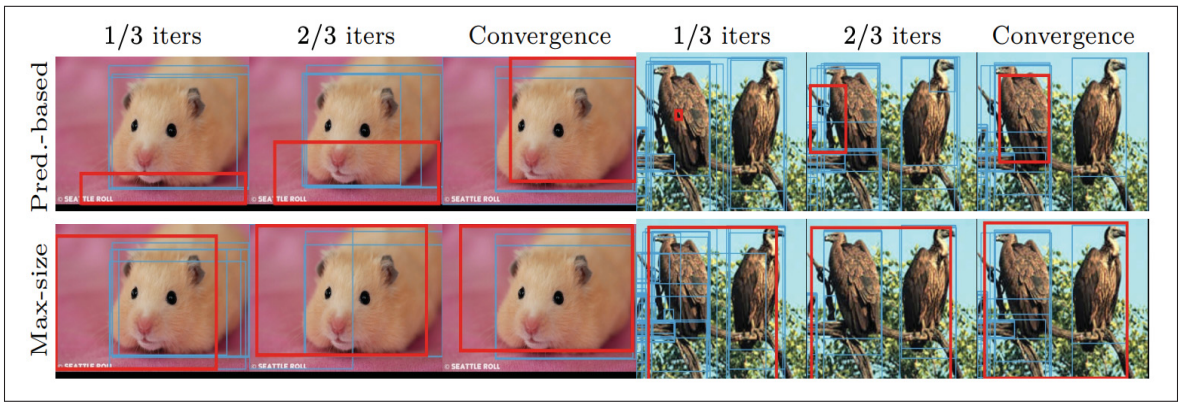


Figure 4.3 Observation from Detic when training with prediction based labels. Taken from Zhou *et al.* (2022). **Top**: The prediction-based method selects different boxes across training, and the selected box may not cover the objects in the image. **Bottom**: By simply selecting the max-size proposal, we get a box that covers the objects and is more consistent across training. All boxes with scores > 0.5 are shown in blue and the assigned (selected) box in red

Algorithm 13 describes the procedure we used for discovering and labeling density crops from the GT annotations. In summary, we perform an iterative merging of the GT boxes to discover the density crops. In the first step, all GT boxes $\mathcal{B}$ are scaled by expanding the min and max coordinates of the boxes by σ pixels (`scale`$(\mathcal{B}, \sigma)$). Then we calculate the pairwise Intersection over Union (IoU) between the scaled boxes (`pairwise_IoU`$(\mathcal{D})$) in O as a $|\mathcal{D}| \times |\mathcal{D}|$ matrix. Connections are labeled in C by assigning one to all overlap values above a threshold θ in the pairwise IoU matrix O. Then we select in C the row m^* with the maximum number of connections. An enclosing box is computed (`enclosing_box`(C_{m^*})) by finding the min and

Algorithm 4.1 Density Crop Labeling Algorithm

Input: $\mathcal{B}$: GT boxes in an image
Output: $\mathcal{D}$: Density crops
Parameters : N: no. of merging steps,
σ: expansion pixels,
θ: overlap threshold,
π: maximum crop size

1. $\mathcal{D} \leftarrow$ `scale`$(\mathcal{B}, \sigma)$;
2. **for** $i \leftarrow 1$ **to** N **do**
    a) $O =$ `pairwise_IoU`$(\mathcal{D})$
    b) $C = O > \theta$ (connection matrix)
    c) $\mathcal{D} \leftarrow \emptyset$
    while $|C| > 0$ **do**
        i) $m^* = \text{argmax}_m \left(\sum_i C_{m,i}\right)$
        ii) $d =$ `enclosing_box`(C_{m^*})
        iii) $\mathcal{D} \leftarrow \mathcal{D} + d$
        iv) $C_{m^*} = 0$
    end while
    d) $\mathcal{D} \leftarrow$ `filter_size`$(\mathcal{D}, \pi)$
end for

max coordinates of all crop members connected to m^*. The newly obtained crop box is added to the list of crops and the row C_{m^*} is set to zeros. Subsequently, the crops that are bigger than a maximum threshold π are removed from the list $\mathcal{D}$ (`filter_size`$(\mathcal{D}, \pi)$). This procedure of iterative merging is performed N times. The crop size threshold π used here is the ratio of the area of the crop to that of the image.

The quality of the crops is important for our method. It is in fact the iterative merging that brings out the best quality crops. Naive scaling and merging to find the maximum enclosing boxes based on pairwise IoU results in either bad crops or too many small crops (with fewer objects in them) depending on the value of the scaling factor. Iterative merging produces good-quality crops enclosing groups of small objects respecting the quality constraints. In the experiments section, we present the ablation studies validating the effectiveness of our density crop labeling algorithm. We also show that hyperparameters of the algorithm can be easily set.

With the newly obtained crop labels, we can also augment the training set with additional image crops. The original image and its annotations $\mathcal{B}$ are down-scaled using the maximum training size $W \times H$. Note that it is expected the detector will not detect many small objects in the down-scaled image. But the augmented up-scaled version of the density crop $d \in \mathcal{D}$ of a given image will have those small objects that fall inside the crop in higher pixel size. This will reduce the extreme scale variation at training time. We used bilinear interpolation for up-scaling the density crops. The crop labeling can be performed as a pre-processing step. The up-scaled version augmentation of density crops is simply a data augmentation process. Thus our method does not introduce any change in the standard training pipeline of a detector, except when the new class "density crop" is added. In this regard, it is practically easy to use like uniform cropping.

4.2.2 Inference with density crops

As the detector is trained to recognize density crops, at inference time, we can get the density crop from its prediction itself. Figure 4.2 bottom explains our inference process in detail. It consists of two stages. In stage one, it predicts the base class objects and density crops on the input image. Then we select the high-quality density crops based on their confidence score. In stage two, the upscaled density crops are passed through the same detector again, producing small object detection on the density crops. Finally, we re-project the detections on the crops to the original image and concatenate them with the detections on the original image. Let $c \in C$ be an up-scaled crop image of size (I_c^W, I_c^H) defined by its bounding box coordinates $(c_{x1}, c_{y1}, c_{x2}, c_{y2})$ in the original image. Given the scaling factors $(S_c^W, S_c^H) = (\frac{c_{x2}-c_{x1}}{I_c^W}, \frac{c_{y2}-c_{y1}}{I_c^H})$, the re-projection box p_i scales down and shifts the detection boxes $(x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}) \in \mathcal{B}^c$ in the crop c as:

$$\begin{aligned} p_i =& (S^W x_{1,i}, S^H y_{1,i}, S^W x_{2,i}, S^H y_{2,i}) \\ &+ (c_{x1}, c_{y1}, c_{x1}, c_{y1}) \end{aligned} \tag{4.1}$$

The Non-Maximal Suppression(NMS) is then applied to remove duplicate detections. While other methods need complex post-processing to filter the noisy crops (Yang *et al.*, 2019), we can

simply use the confidence score of the density crops to do the same. Stage one of the inference is the standard inference procedure in any detector. The filtering of the noisy crops can be easily performed with the confidence scores given by the detector. The second stage of the inference is performed with the same detector, but a different input (the up-scaled density crops). So, we are simply repeating the standard inference procedure of a detector one more time. All of these operations can be easily wrapped on top of the inference procedure of any detector, thus keeping the simplicity of the uniform cropping approach at inference time too.

4.3 Experiments with CZ Detector

Datasets and evaluation measures. For the evaluation of methods, we employed two popular challenging benchmark datasets for Aerial Image Detection, namely the VisDrone (Zhu *et al.*, 2018) and DOTA (Xia *et al.*, 2018) datasets. The measure used for assessing and comparing the performance of methods is COCO style average precision (AP) (Lin *et al.*, 2014). The AP of small, medium, and large objects are also reported, particularly to understand the performance of our method for small object detection. Finally, the number of frames per second (FPS) is reported as a measure of time complexity.

VisDrone. This dataset contains 8,599 drone-captured images (6,471 for training, 548 for validation, and 1,580 for testing) with a pixel size of about 2000 ×1500 pixels. The objects are from ten categories with 540k instances annotated in the training set, mostly containing different categories of vehicles and pedestrians observed from drones. It has an extreme class imbalance and scale imbalance making it an ideal benchmark for studying small object detection problems. As the evaluation server is closed now, following the existing works, we used the validation set for evaluating the performance.

DOTA. This dataset is comprised of satellite images. The images in this dataset have a pixel size ranging from 800×800 to 4000×4000. Around 280k annotated instances are present in the dataset. The objects are from fifteen different categories, with movable objects such as planes, ships, large vehicles, small vehicles, and helicopters. The remaining ten categories are

roundabouts, harbors, swimming pools, etc. Many density crop-based detection papers report results only on movable objects(Yang *et al.*, 2019) with the assumption that immovable objects usually won't appear crowded. But they are also small objects, so we kept all classes to assess the improvement in small object detection. The training and validation data contain 1411 images and 458 images, respectively.

Implementation details. The Detectron2 toolkit (Wu, Kirillov, Massa, Lo & Girshick, 2019) was used to implement our CZ detector. The backbone detector used in our study is primarily Faster RCNN (Ren *et al.*, 2015), but we also show results on the modern anchor-free one-stage detector FCOS (Tian *et al.*, 2019). We used Feature Pyramid Network (FPN) (Lin *et al.*, 2017b) backbone with ResNet50 (He *et al.*, 2016) pre-trained on ImageNet (Russakovsky *et al.*, 2015) dataset for our experimental validation. For data augmentation, we resized the shorter edge to one randomly picked from (800, 900, 1000, 1100, 1200), and applied horizontal flip with a 50% probability. The model was trained on both datasets for 70k iterations. The initial learning rate is set to 0.01 and decayed by 10 at 30k and 50k iterations. For training, we used one NVIDIA A100 GPU with 40 GB of memory.

4.3.1 Comparison with state-of-the-art methods

Table 4.1 compares our approach with the existing methods on the VisDrone dataset. Similarly to us, some methods perform density cropping (Yang *et al.*, 2019; Li *et al.*, 2020; Duan *et al.*, 2021; Deng *et al.*, 2020), while QueryNet (Yang *et al.*, 2022) and CascadeNet (Zhang, Izquierdo & Chandramouli, 2019) use other approaches to improve the detection performance on aerial images. We obtained the best detection AP among the state-of-the-art methods. Only for large objects, DensityMap performs better than our approach. This is probably because our method gets biased to detect small objects, thanks to the additional crops on training. In fact, for small object detection, we obtained the best AP_s, significantly outperforming all existing approaches. AP_m also shows a good improvement of more than 2 percentage points.

Table 4.1 Performance of our proposed method compared against state-of-art approaches with Faster RCNN detector on the VisDrone validation set (results in %). "MF" stands for model fusion

Method	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$
ClusterNet (Yang *et al.*, 2019)	26.72	50.63	24.70	17.61	38.92	51.40
DensityMap (Li *et al.*, 2020)	28.21	47.62	28.90	19.90	39.61	**55.81**
CDMNet (Duan *et al.*, 2021)	29.20	49.50	29.80	20.80	40.70	41.60
GLSAN (Deng *et al.*, 2020)	30.70	55.40	30.00	-	-	-
QueryDet (Yang *et al.*, 2022)	28.32	48.14	28.75	-	-	-
CascadeNet (Zhang *et al.*, 2019)	28.80	47.10	29.30	-	-	-
CascNet+MF (Zhang *et al.*, 2019)	30.12	58.02	27.53	-	-	-
CZ Det. (Ours)	**33.22**	**58.30**	**33.16**	**26.06**	**42.58**	43.36

4.3.2 Comparison with baselines

Table 4.2 presents a comparison between uniform cropping and density cropping on the VisDrone dataset, with and without the last feature map of the feature pyramid (P2), which has a strong impact in memory and computation (Yang *et al.*, 2022). For the uniform cropping, we crop the original image into 4 equal-sized crops by splitting at half height and width. In order to have a fair comparison, we use our method with a confidence threshold of 0.7 to obtain an average of 1-3 crops per image. The observations in the table suggest that uniform cropping improves performance compared to vanilla training on the whole image, but it is still inferior to our density-based cropping. When high-resolution feature maps P2 are not used, density cropping gains more than 3.5 percentage points in AP, and the AP of small objects is improved by 3.4 percentage points. It is worth noting that compared to uniform cropping, our approach introduces additional parameters to recognize one extra class and no changes in learning and inference dynamics. So this can be easily used as a plug-and-play replacement for the uniform crop-based training, popular among the community. In terms of frame rate, our approach is slightly slower than uniform crops. However, we observe that our method without the expensive P2 features performs better than uniform crops with P2, while also being faster. In Figure 4.4, a visual comparison of the highly confident detections between the baseline model and our density crop-based model is shown. When the density crops are used, we can observe an increase in the

number of detections. It can be observed that more objects are getting discovered in the crop regions when the detection results from the second inference are augmented. This explains the impact of our zoom-in detector for small object detection in high-resolution images.

Table 4.2 Comparison of detection performance between a baseline detector, uniform crops, and density crops on the VisDrone dataset (1.5K pixels). The results are in %. The small, medium, and large objects are grouped according to the coco evaluation protocol

Settings	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**FPS**
Without P2							
Baseline	29.48	51.68	29.55	22.33	38.66	39.30	26.31
Uniform crops	30.68	54.44	30.54	22.91	40.62	41.03	12.30
CZ Det. (ours)	33.02	57.87	33.09	25.74	42.93	41.44	11.64
With P2							
Baseline	30.81	55.06	30.68	23.97	39.19	41.17	18.25
Uniform crops	31.73	56.31	31.57	25.13	40.41	41.06	9.85
CZ Det. (ours)	33.22	58.30	33.16	26.06	42.58	43.36	8.44

To further verify the observations, we repeated the same type of study in the satellite images of the DOTA dataset. In this dataset images are at higher pixel size (4k pixels), thus due to memory constraints, the baselines are already performing uniform cropping. Table 4.3 shows the results of a uniform cropping baseline and our CZ detector for two different configurations. Similar to VisDrone, significant improvement is seen in the case of not using high-resolution features P2, with a gain of 2.9 percentage points. APs of small and medium objects are improved by 3.0 and 4.0 percentage points respectively from the baseline without using high-resolution features. In terms of computation, we can see that, as expected, our approach has a slightly slower frame rate than the baseline. However, this is compensated by the higher detection accuracy. We see for instance that the best baseline with P2 features has an AP of 33.44% with an FPS of 0.49, while our CZ detector without P2 features has a higher AP (34.14%) while being also faster (0.62 FPS).

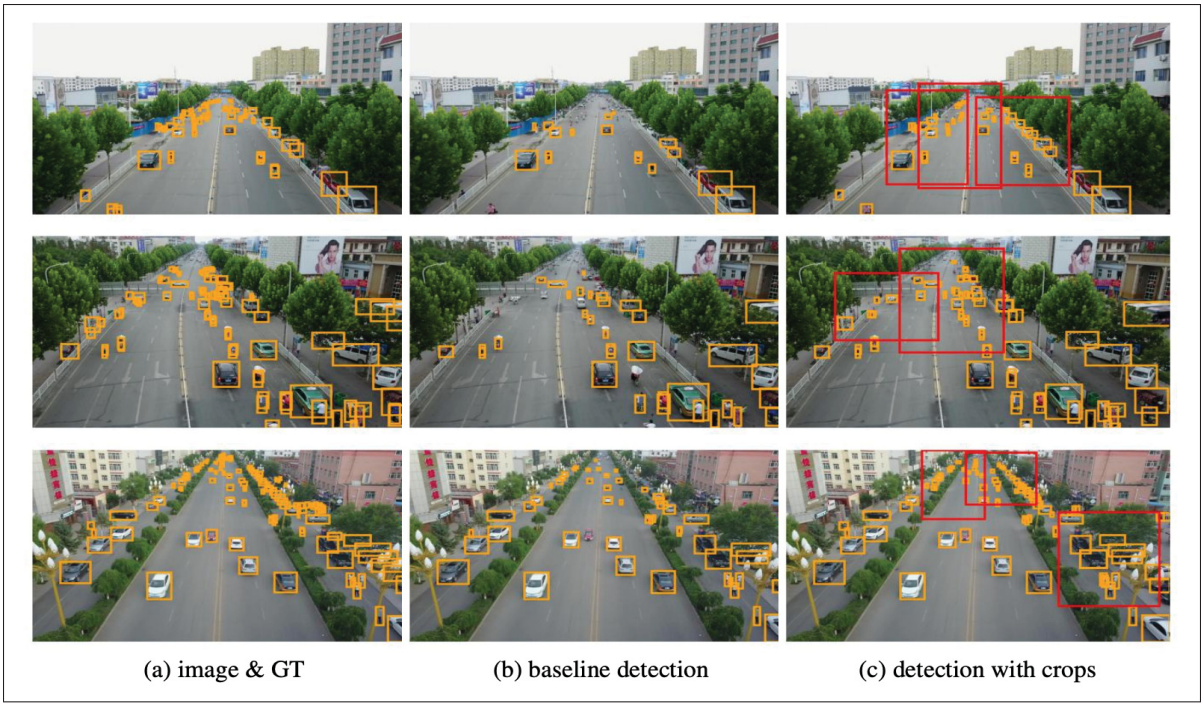


Figure 4.4 Visualization of density crop-based detection. (a) the original image and its GT. (b) detection with the baseline detector. (c) detection with density crops; the density crops are shown in red color. Our method detects more objects, especially inside the crop regions

Table 4.3 Performance comparison of our method against baselines on DOTA dataset (4k pixels). The results are in %

Settings	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**FPS**
Without P2							
Baseline	31.29	51.57	33.10	12.69	34.04	42.83	0.93
CZ Det. (Ours)	34.14	56.69	35.69	15.66	38.16	44.20	0.62
With P2							
Baseline	33.44	54.03	35.56	16.86	36.76	43.65	0.49
CZ Det. (Ours)	34.62	56.86	36.17	18.17	37.84	43.83	0.30

4.3.3 Ablation studies

The effectiveness of the proposed CZ detector is characterized by ablation experiments on the VisDrone dataset. We perform ablation studies to understand the impact of density crops at

training and test time, the impact of the quality of the crops, and the impact of the iterative merging in the crop labeling algorithm.

4.3.3.1 Density crops effect at training and inference

We used density crops at the training and test time to achieve the best performance. In particular, while training, the rescaled density crops are augmented with the training images; while testing we do the two-stage inference where stage one performs inference on the whole image and stage two performs inference on the density crops. In this section, we study the importance of this configuration. Table 4.4 shows the results. When the density crops are not augmented with the training set but only used in the two-stage inference, the improvement is marginal over the baseline (most importantly, AP_s has no change). This is because the scale imbalance in the input image is not mitigated as the detector does not see the small objects at a bigger scale. When density crops are added to the training set, the detection accuracy improves significantly. However, the inference is still happening on the whole image so the detection accuracy of small objects is affected. When inference is performed on the density crops and fused with the detection on the whole image, we get the best results.

Table 4.4 Detection results with and without density crops at train time and test time (results in %)

Train	**Test**	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_s$**	**AP$_m$**	**AP$_l$**
		29.48	51.68	29.55	22.33	38.66	39.30
	✓	29.93	53.29	29.52	22.33	39.35	39.46
✓		32.64	57.36	32.78	24.81	43.04	41.07
✓	✓	33.02	57.87	33.09	25.74	42.93	41.44

4.3.3.2 Impact of the quality of crops

Figure 4.5 illustrates how the confidence of crops impacts the detection accuracy and the number of density crops extracted. The impact is studied for two settings, with and without the high-resolution features P2. This is to verify how the density crops aid small object detection

with and without utilizing expensive high-resolution feature maps. The crop confidence, which is used as the proxy for crop quality, is varied from 0.1 to 0.9. In general, with lower confidence values, we are observing more crops but many of them are noisy and redundant even after Non-Maximal Suppression. So when the quality of the crops is low, the detection accuracy decreases (Figure 4.5 left). When the quality is increased, the accuracy increases until 0.7, and then it gradually comes down as we use very few crops in that case. The trend is the same with and without P2.

From Tables 4.2 and 4.3, we observed that density crops obtained better gain in detection accuracy over the baseline without high-resolution features. Though this is expected, we decided to understand how exactly this is happening. We analyzed the number of density crops retained after filtering out the low-quality crops at multiple confidence levels ranging from 0.1 to 0.9. Figure 4.5 right shows the results with and without high-resolution features P2. It can be observed that for "without P2", we are getting more density crops at all confidence levels. With higher crop confidence levels, we get more high-quality crops for the "without P2" case, hence we observe a better gain in detection accuracy over the baseline. We used a confidence of 0.7 in all our experiments to have the best trade-off between detection precision and speed. While other methods use post-processing on the crop detections (Yang *et al.*, 2019) or density maps (Li *et al.*, 2020) to filter the noisy crops during inference, we can filter them out based on their confidence score simplifying the inference procedure.

4.3.3.3 Why iterative merging for crop discovery?

Simply scaling and doing a one-step merging operation to create density crops results in sub-optimal crops. We empirically verify this with multiple scaling strategies and argue that the iterative merging strategy is superior to them. Yang *et al.* (2019) also used iterative crop merging on the output of their crop detection module to reduce the redundant crops. This has to be performed at training and test time to refine the initial crop detections. To label the crops for training, they used a single-step aggregation. Our iterative merging for labeling crops can be

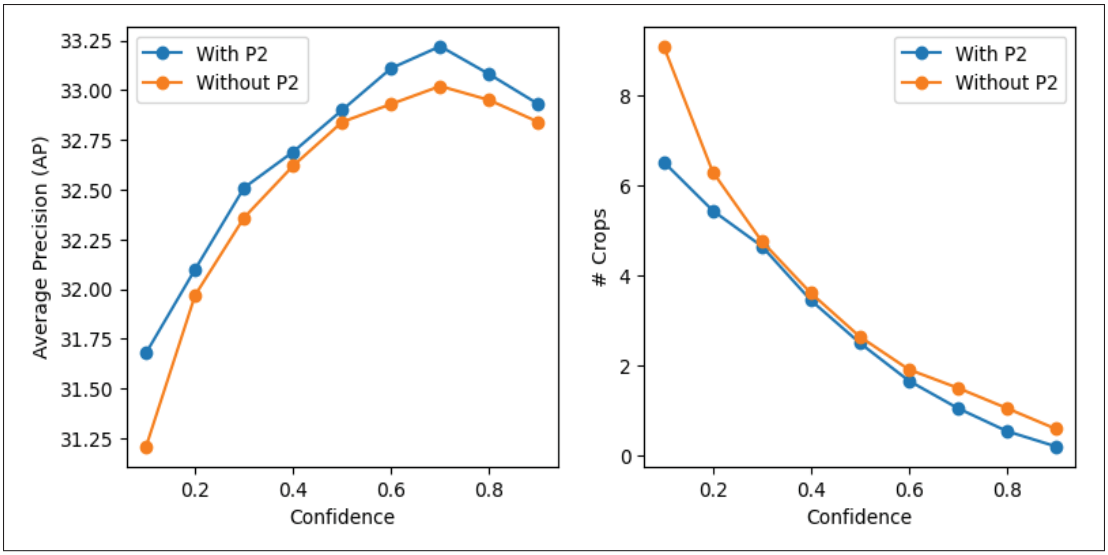


Figure 4.5 Change in detection precision and the number of crops according to crop confidence. The crop confidence is varied from 0.1 to 0.9. The crop confidence for best detection accuracy is 0.7

performed as a pre-processing step before training. We avoided redundant crops at inference, by filtering them out based on the confidence score.

Table 4.5 top provides the comparative results of single-step merging with our iterative merging strategy where GT boxes are scaled by a scaling factor. Using a low scaling factor creates too many crops, containing fewer objects. More specifically, it produces multiple small crops containing fewer objects in crowded regions in the image. When the scaling factor is increased, the number of crops decreases and performance increases up to a point but declines later as the crops become too big and the object density of the image is less respected. This is because large scaling factors significantly blow up the big GT boxes and it alters the density of the crops. The detection performance obtained is also far below our iterative merging. Table 4.5 bottom shows the same comparison when GT boxes are scaled by constant pixel values. As this avoids the blowing of large bounding boxes due to the constant scaling, the detection performance is better than the former one. Iterative merging produces the optimal number of crops with the best performance. The scaling used in the iterative merging is small and only performed at the first stage of merging. We used 20 pixels as the scaling magnitude. Large values are not possible

here since the `filter_size` operation while restricting the crop size will reduce the number of crops. Thus it is easy to set.

Table 4.5 Comparison of iterative merging strategy with single-step merging where GT boxes are scaled according to scaling factors, and scaled uniformly by pixel values (results in %)

Scaling	**# crops**	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_s$**	**AP$_m$**	**AP$_l$**
Baseline	0	29.48	51.68	29.55	22.33	38.66	39.30
factor = 2.0	74417	24.39	44.38	23.73	15.96	34.96	47.24
3.0	67906	30.64	53.71	30.68	23.23	40.64	39.35
6.0	43300	31.30	55.33	31.42	23.79	41.06	38.40
8.0	34663	30.95	55.18	30.31	23.38	40.93	39.24
pixels = 30	62677	31.26	54.55	31.50	23.83	40.78	50.07
60	46753	31.98	55.84	32.07	25.12	41.11	45.52
90	35442	31.47	55.62	30.96	24.03	41.27	44.08
120	25146	31.07	54.74	30.95	23.18	41.43	42.39
Ours	14018	33.02	57.87	33.09	25.74	42.93	41.44

4.3.4 Results with other detectors

To validate the effectiveness of our approach with other detection architectures, we conducted experiments on the modern anchor-free one-stage detector FCOS (Tian *et al.*, 2019). Table 4.6 shows the performance comparison of the vanilla FCOS detector with our density crop-based FCOS detector. Similar to the results in Table 4.2, AP is improved by a significant margin, and AP$_s$ has gained almost 5 percentage points. We can also see that density crop-based FCOS has superior performance than their Faster RCNN counterpart in terms of most of the metrics. This is interesting because other density-based approaches weren't producing better results with one-stage detectors than the two-stage ones.

4.4 Density Crop-guided Semi-supervised Detector

In this section, we will elaborate on the semi-supervised detection method proposed which is guided by density crops from clusters of objects. The annotation challenge is further exacerbated

Table 4.6 Results with anchor free detector FCOS on the Visdrone dataset (results in %). All results are without using P2

Settings	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
Base FCOS	29.51	50.40	29.92	21.25	40.51	37.29	26.01
CZ FCOS Det.	33.67	56.20	34.15	26.16	43.98	46.87	12.69

in aerial images where the annotators have to label small objects often distributed as crowded in those clusters on high-resolution images. Getting sufficient labeled data is difficult in aerial images, especially at instance-level recognition tasks like object detection (Xu *et al.*, 2021; Liu *et al.*, 2021a; Meethal *et al.*, 2022), limiting the scalability of the popular supervised detectors to aerial images. Practical applications with aerial imagery produce large amounts of unlabeled data (Caillouet, Giroire & Razafindralambo, 2019; Sun, Shao, Cheng, Huang & Wang, 2022; Song *et al.*, 2021) but they are simply not utilized in the learning process. This builds a perfect scope for semi-supervised detectors in aerial images where we can train a detector with limited annotated images and a large collection of unlabeled data.

Though Semi-supervised Object Detection(SSOD) has achieved tremendous progress in recent years on natural images (Guo *et al.*, 2022; Li *et al.*, 2022a; Liu *et al.*, 2021a; Xu *et al.*, 2021; Tang *et al.*, 2021; Jeong *et al.*, 2019; Sohn *et al.*, 2020; Meethal *et al.*, 2022), we are yet to see large-scale adoption of them on aerial images. Needless to say, the mean-teacher based semi-supervised learning framework is the core component behind the success of these semi-supervised detectors. Even though the mean-teacher based semi-supervised detectors are excellent in natural images, their direct translation on aerial images is not optimal as we will see from the empirical studies. With our density crop-guided semi-supervised detection, we improved the vanilla mean-teacher significantly. We believe that the reason why the vanilla mean-teacher method struggles is because it is not produce enough pseudo-labels for small objects. The number of target objects is fairly high in aerial images compared to natural images. For example, the average number of objects in Pascal VOC and MS-COCO images are 3 and 7, respectively, whereas images in the VisDrone (Zhu *et al.*, 2018) and DOTA (Xia *et al.*, 2018)

datasets – two popular benchmarks in the aerial detection research – have an average number of 53 and 67 objects, respectively. The pseudo label-based mean-teacher detectors, in this case, are not labeling enough small objects in the unlabeled images. This is probably due to the fact that a baseline detector will not detect enough small objects on high-resolution aerial images. Figure 4.6 summarizes our observation. When the detector is trained with density crops, it creates more pseudo-GT boxes compared to the vanilla mean-teacher training.

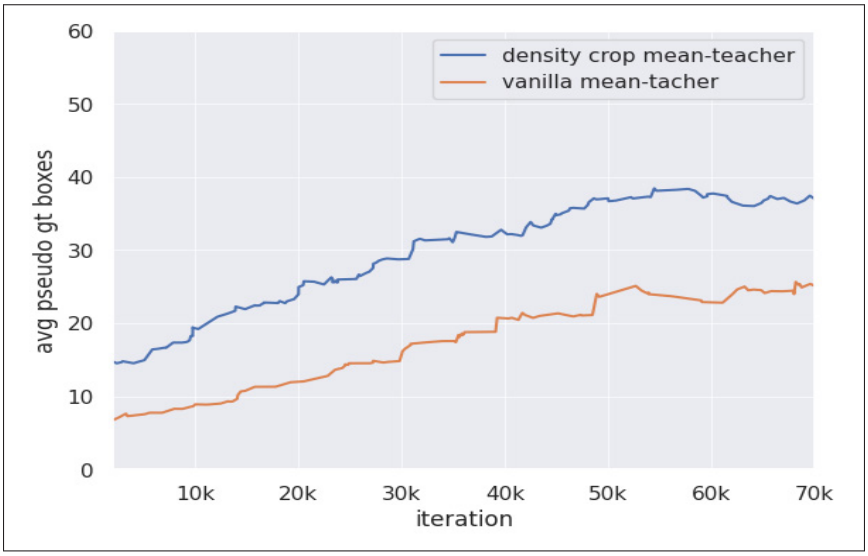


Figure 4.6 Average number of pseudo-GT boxes over iteration in a minibatch. The density crop-guided mean-teacher is producing more pseudo labels compared to the vanilla mean-teacher method. This will result in more pseudo-labels for small objects

While density crops can be used in supervised settings with external learnable modules and additional loss functions (Duan *et al.*, 2021; Yang *et al.*, 2019; Li *et al.*, 2020), using them in the semi-supervised settings with mean-teacher method (Tarvainen & Valpola, 2018) requires the crops to be detected with the detector itself. This is where the CZ Detector design shines compared to other density-based approaches. The external module in other approaches may need additional loss functions and often times they are trained before the detector with sufficient labeled data. Also, it is not immediately clear how to construct pseudo labels for the density module if one wants to train them in the mean-teacher settings using unlabeled images.

With the CZ detector, density crops can be identified on the labeled and unlabeled images. For the labeled images, they are identified as apriori with the available ground-truth (GT) labels. For the unlabeled images, pseudo-GT predictions are utilized to locate the cluster of small objects and then labeled as density crops. Crops identified on both labeled and unlabeled images are used to augment the training set. The augmented crops result in more samples of small objects seen at higher pixel resolution improving their detection chance. The detector is then trained in the mean-teacher fashion with weak-strong augmentation consistency and pseudo labels for the unlabeled images. At inference, detection is performed separately on both the input image and upscaled density crops if any are present in that image. They are then fused and post-processed to get the final results. Figure 4.7 shows how the density crops are improving the detection AP on the VisDrone dataset. It can be observed that by utilizing the density crops effectively in the semi-supervised settings, our detection accuracy increases significantly over the vanilla semi-supervised detector, as seen in the fully supervised settings.

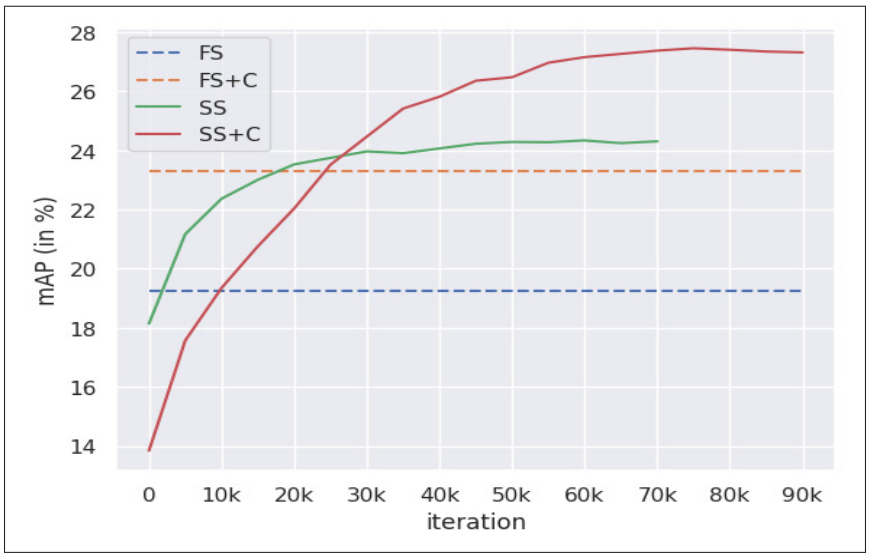


Figure 4.7 Change in mAP over the epochs with and without density crops on supervised and semi-supervised settings. FS: Fully Supervised, FS+C: Fully supervised + density crops, SS: Semi-supervised (mean-teacher baseline), SS+C: Semi-supervised + density crops (on labeled and unlabeled images)

Our main contributions can be summarized as follows:

(1)A density-crop guided semi-supervised detector is proposed for aerial images. It adapts the vanilla mean-teacher semi-supervised detector with mechanisms to identify and process the cluster of small objects, improving their suitability for training semi-supervised detectors on high-resolution aerial images.

(2) We empirically validate the benefits of our semi-supervised detection method on aerial images from drones (VisDrone) and satellites (DOTA), and observed a consistent improvement in the detection accuracy on both datasets over the supervised training on the labeled data.

4.4.1 Semi-supervised training

Semi-supervised learning takes place by distilling the weights of a detector (called a student network) during training to another identical copy of the network (called the teacher network) by exponential moving average (EMA). The teacher network is generally more stable due to the slower pace at which it temporally ensembles the noisy student weights, so it is used to give pseudo-GT for the unlabeled images (Liu *et al.*, 2021a). The student network learns its weights by optimizing a combination of supervised and unsupervised loss. For the labeled data, we have the GT annotations to compute the supervised loss $\mathcal{L}_{sup}$. Let the available labeled data is $D_s = \{x_i, y_i\}_{i=1}^{N_s}$, where each y_i is a bounding box coordinate and its class label ($y_i = (b_i, c_i)$). Here N_s is the number of labeled samples. For the unlabeled data $D_u = \{x_i\}_{i=1}^{N_u}$, we get pseudo-GT $\hat{y}_i$ from the teacher network which is used to calculate the unsupervised loss $\mathcal{L}_{unsup}$. Here N_u is the number of unlabeled samples. Finally, the network is trained by optimizing the following loss

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda \mathcal{L}_{unsup} \tag{4.2}$$

where λ is a hyperparameter to control the relative importance of the supervised and unsupervised loss. Figure 4.8 shows the overall architecture of our semi-supervised learning system. At each iteration, we sample a minibatch of labeled and unlabeled samples following a preset ratio d_r. Each data point in the minibatch undergoes two types of transformation, referred to as weak and strong augmentation. The weak augmentation is simply the rescaling and horizontal flip

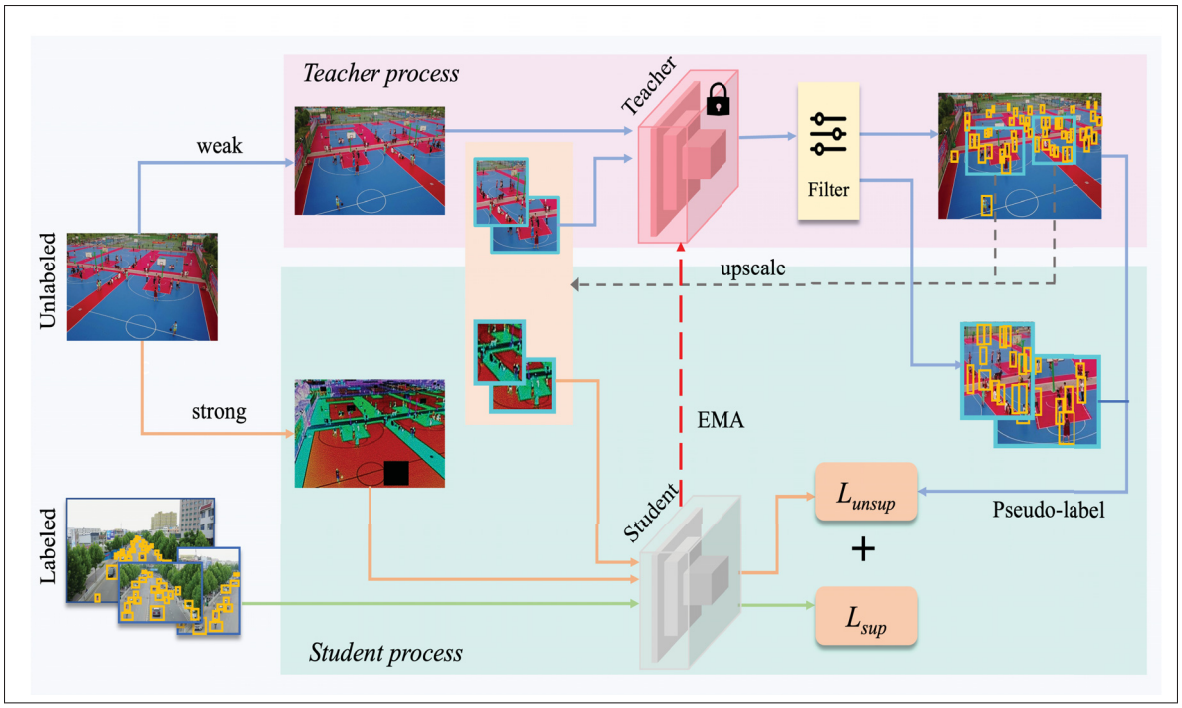


Figure 4.8 The pipeline of our proposed density crop guided semi-supervised detection. The training data contains both labeled and unlabeled images. There are two networks that are identical copies of the backbone detector. The student network is learned via backpropagating the loss gradients, whereas the teacher network is an exponential moving average (EMA) of the student weights. The labeled images are passed through the student network and supervised loss $\mathcal{L}_{sup}$ is calculated. Unlabeled images are passed to the teacher network, whose predictions are then filtered (we used confidence thresholding here) to get good-quality pseudo-labels. If there are dense clusters of small objects in the unlabeled image, such clusters are cropped and passed after up-scaling to the teacher network. Then pseudo-labels are computed on newly added density crops as well in a similar fashion. A strongly augmented version of the unlabeled images and their density crops are then passed to the student network. The loss $\mathcal{L}_{unsup}$ is calculated based on the pseudo-labels obtained before. The combined loss is then backpropagated to update the student weights. Teacher weights are then updated by EMA of the student weights

transformation. The strong augmentation includes color jittering, grayscale, Gaussian blur, and cutout patches which perform only pixel-level transforms, thus the bounding box labels need not be transformed. We followed the scale ranges provided in Liu *et al.* (2021a) for the strong augmentation. The augmented images then go through the mean-teacher semi-supervised learning process. We followed Liu *et al.* (2021a) for the mean-teacher training implementation.

We compute density crops on the unlabeled images using pseudo-labels from the teacher. This is then used to augment more crops, this time from the unlabeled images. In the following, we will describe the semi-supervised learning process in detail.

4.4.1.1 Burn-in stage

To get reliable pseudo-GT for the unlabeled images, the teacher network should have a good initialization. Typically, existing methods perform a supervised pre-training with the available supervised data to get this good initialization (Sohn *et al.*, 2020; Tang *et al.*, 2021; Liu *et al.*, 2021a). This supervised pre-training is called the Burn-in stage. During burn-in, we optimize $\mathcal{L}_{sup}$ only which is a sum of classification and localization losses of the detector:

$$\mathcal{L}_{sup} = \sum_{i=1}^{N_s} \mathcal{L}_{cls}(f_W(x_i), y_i) + \mathcal{L}_{reg}(f_W(x_i), y_i) \tag{4.3}$$

that are defined as in Ren *et al.* (2015). After burn-in, the weights of the network W are copied to the teacher ($W \rightarrow W_t$) and student network ($W \rightarrow W_s$). From this point, unsupervised data is also used in the learning process with teacher-student mutual learning.

4.4.1.2 Teacher-student learning stage

The teacher-student learning process optimizes the loss in equation 4.2 to learn the student network (with backpropagation), whereas the teacher network is learned by temporally accumulating the student weights (with EMA). It combines consistency regularization and pseudo label-based learning - the most popular approaches for semi-supervised learning - in one framework. The consistency regularization is ensured with the weak-strong augmentation prediction consistency. Pseudo-label-based learning is performed by producing pseudo-labels on the unlabeled images.

The weakly augmented version of unlabeled data first goes through the teacher network producing the instance predictions. Let $y_j^{pred} = (b_j^{pred}, c_j^{pred}, p_j^{pred})$ be instance predictions containing

predicted box b_j^{pred}, class c_j^{pred} and probability p_j^{pred} where y_{pred} is obtained as

$$y^{pred} = f_{W_t}(x) \tag{4.4}$$

This prediction then undergoes confidence thresholding to produce pseudo labels $\hat{y}$. The confidence thresholding considers all predictions with a class probability above a threshold τ as foreground instances:

$$\hat{y} = \{y_j^{pred} | p_j^{pred} > \tau, \forall j \in y^{pred}\} \tag{4.5}$$

This is the filtering process shown in figure 4.8. We then obtain the pseudo labels for the unlabeled images and compute the $\mathcal{L}_{unsup}$. For that, the strongly augmented version of the unlabeled data is passed through the student network to get the predictions. The unsupervised loss is then applied to the classification head as follows:

$$\mathcal{L}_{unsup} = \sum_{i=1}^{N_u} \mathcal{L}_{cls}(f_{W_s}(x_i), \hat{y}_i) \tag{4.6}$$

$\mathcal{L}_{unsup}$ is not applied to the localization head of the detector because the pseudo labeling with confidence thresholding is suitable only for estimating confident class predictions; it has no information about the bounding box correctness. After computing $\mathcal{L}_{unsup}$, we update the student network weights W_s by optimizing equation 4.2. The teacher weights W_t are then updated by EMA as follows:

$$W_t = \alpha W_t + (1 - \alpha) W_s \tag{4.7}$$

where α is a hyperparameter that controls the pace at which the student weights are updated to the teacher weights.

4.4.1.3 Semi-supervised training algorithm

Algorithm 4.2 summarizes our density crop-guided semi-supervised training process. Given the labeled and unlabeled data D_s and D_u respectively, we first compute and label crops in D_s using the available ground-truth labels. The training process then begins. We load a batch of

Algorithm 4.2 Density-crop Semi-supervised Training

Input: labeled and unlabeled images: D_s, D_u
Output: teacher and student weights: W_t, W_s
Parameters : n: start of labelling D_u iteration,
N: maximum iterations, μ: learning rate

1. M = crops(D_s) # *Compute crops on* D_s
2. $D_s \leftarrow D_s + M$ # *Add M to* D_s
3. **for** $i \leftarrow 1$ **to** N **do**
    a) $x_s, y \leftarrow$ batch(D_s)
    b) $x_u \leftarrow$ batch(D_u)
    b) Compute $\mathcal{L}_{sup}(x_s, y)$ using eqn. 4.3
    c) Obtain $\hat{y}_u$ using eqn. 4.5
    d) Compute $\mathcal{L}_{unsup}(x_u, \hat{y}_u)$ using eqn. 4.6
    e) Compute $\mathcal{L}$ using eqn. 4.2
    f) $W_s \leftarrow W_s - \mu \frac{\partial \mathcal{L}}{\partial W_s}$
    g) update W_t using eqn. 4.7
    i) **if** $i == n$ **then**
        for $j \leftarrow 1$ **to** $|D_u|$ **do**
            I) compute $\hat{y}^j$ using eqn 4.5
            II) m = crops(x_u^j, $\hat{y}^j$) # *Gets crops on* x_u^j
            III) $D_u \leftarrow D_u + m$ # *Add m to* D_u
        end for
    end if
end for

images from both the labeled and unlabeled pool. The batch loaded from labeled pool x_s is directly used to calculate $\mathcal{L}_{sup}$. For the batch from unlabeled pool x_u, strongly augmented and weakly augmented versions are produced. The teacher processes weakly augmented images computing pseudo labels $\hat{y}_u$ for the images in x_u. This is then used to compute $\mathcal{L}_{unsup}$ where the loss is computed against the student predictions obtained using strongly augmented images. The combined loss $\mathcal{L}$ is backpropagated, and then teacher weights are updated using the EMA update rule in equation 4.7. When this training process is converged (after a sufficient number of iterations n), crops are computed on the unlabeled images and used to further augment D_u.

4.4.2 Density Crops on unlabeled images

As density crops help to process crowded image regions in higher pixel resolution and improve small object detection performance, it is useful to find them on unlabeled images as well. Moreover, there are more unlabeled images than labeled images in the standard semi-supervised settings. Thus we will be able to recover more density crops if we identify them from the unlabeled images. While for the labeled data D_s we have the GT labels y to run crop-labeling algorithm 13, we don't have annotations for the unlabeled data D_u to produce density crops. As we have plenty of unlabeled images, we could get more augmented crops from dense regions of unlabeled images, also increasing samples for the crop category. Thus we expect further improvement in performance if density crop-based learning can be utilized on unlabeled images as well. To do so, we rely on the predictions of the teacher network. Particularly, we utilize the pseudo labels provided by the teacher network to label crops on the unlabeled images, again using algorithm 13.

After the semi-supervised training with labeled and unlabeled data (where crops are only augmented on labeled images) is converged, we use the final teacher model to get the predictions on the unlabeled images. These predictions are then processed to get accurate pseudo GTs following confidence thresholding as in equation 4.5. Crop labeling on the unlabeled images is then performed following algorithm 13 this time with pseudo-GT boxes. The semi-supervised training is then continued as before but with more unlabeled data points obtained from the cluster of small objects in the unlabeled images. As the clusters mostly remain intact on the unlabeled images at this point, it is not necessary to recompute them at every iteration. We recomputed them at every 10,000 iterations to make the training faster.

4.5 Experiments with Semi-supervised CZ Detector

The empirical study is again performed on the VisDrone (Zhu *et al.*, 2018) and DOTA (Xia *et al.*, 2018) datasets. A Faster RCNN (Ren *et al.*, 2015) model with FPN (Lin *et al.*, 2017b) backbone is used as the detector.

4.5.1 Comparison with different percentages of labeled data

We analyzed the effectiveness of our semi-supervised learning method by using partially labeled data from the train set of VisDrone and DOTA datasets. In particular, we used 1%, 5%, and 10% randomly chosen data points from the train set as labeled data and the remaining as unlabeled for the semi-supervised training. There are five settings in the comparison; supervised baseline, supervised baseline with density crops (Supervised + Dcrop), semi-supervised with the mean teacher (SSOD), SSOD with density crops on labeled images (SSOD + Dcrop (L)), and SSOD with density crops on labeled and unlabeled images (SSOD + Dcrop (L + U)). These settings progressively assess the impact of the components of our density crop-guided semi-supervised object detection.

Table 4.7 presents the results for the VisDrone (Zhu *et al.*, 2018) dataset. It compares the detection average precision values obtained using the COCO evaluation protocol (Lin *et al.*, 2014) for Intersection over Union (IoU) thresholds [0.5:0.05:0.95] (**AP**), and 0.5 ($\mathbf{AP}_{50}$). It can be observed that **AP** is improved by more than 6% in all cases with our density-guided SSOD over their supervised baseline. Compared to the vanilla mean-teacher method (SSOD), our density crop-guided SSOD shows an average improvement of more than 2% on all metrics. Compared to 1% and 5% cases, with very limited labeled samples per class, 10% shows a better boost in performance while leveraging density crops with SSOD. Another interesting result is that the improved performance with semi-supervised learning for 1% settings is more than that of supervised training with 5% labels and 2% below with the 10% labels. This is achieved with less than 100 labeled samples. AP_{50} has a gain of more than 5% compared to the vanilla mean-teacher when semi-supervised learning is performed with density crops in the 10% setting.

We also studied how the AP of small, medium, and large objects behave in the same five settings described above. Figure 4.9 shows the results. The trend here is similar to that of table 4.7. Using density crops increases the detection accuracy both in supervised and semi-supervised settings. Compared to the supervised settings, the AP of all-sized objects increases by more than 5% when semi-supervised learning is performed with density crops. The improvement

Table 4.7 Performance comparison of our density crop guided semi-supervised object detection with 1%, 5%, and 10% labeled images on the VisDrone dataset. The detection speed is also reported in FPS. SSOD - semi-supervised detection with mean-teacher, Dcrop(L) - density crops on the labeled images, Dcrop (L + U) - density crops on the labeled and unlabeled images

Settings	1% (#Labeled =64)		5% (#Labeled =323)		10% (#Labeled =647)	
	AP	**AP$_{50}$**	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**
Supervised	10.7±0.2	23.5±0.2	16.1±0.1	32.6±0.2	19.3±0.1	37.7±0.2
Supervised + Crop	13.0±0.2	27.0±0.2	20.9±0.3	40.2±0.3	23.3±0.2	43.9±0.3
SSOD	15.3±0.3	29.2±0.5	21.9±0.2	40.6±0.3	24.4±0.2	43.1±0.2
SSOD + Crop (L)	16.6±0.2	31.1±0.2	22.5±0.1	41.3±0.2	26.5±0.2	47.5±0.1
SSOD + Crop (L + U)	**17.2**±0.2	**31.2**±0.2	**23.6**±0.2	**42.3**±0.2	**27.5**±0.2	**49.0**±0.1

over the vanilla mean-teacher is more than 3% in most settings. The APs of small, medium, and large objects with fully supervised training using 100% labeled data are 25.74, 42.93, and 41.44 respectively. It can be observed that our model with 10% labeled data performs competitively with this fully supervised upper bound.

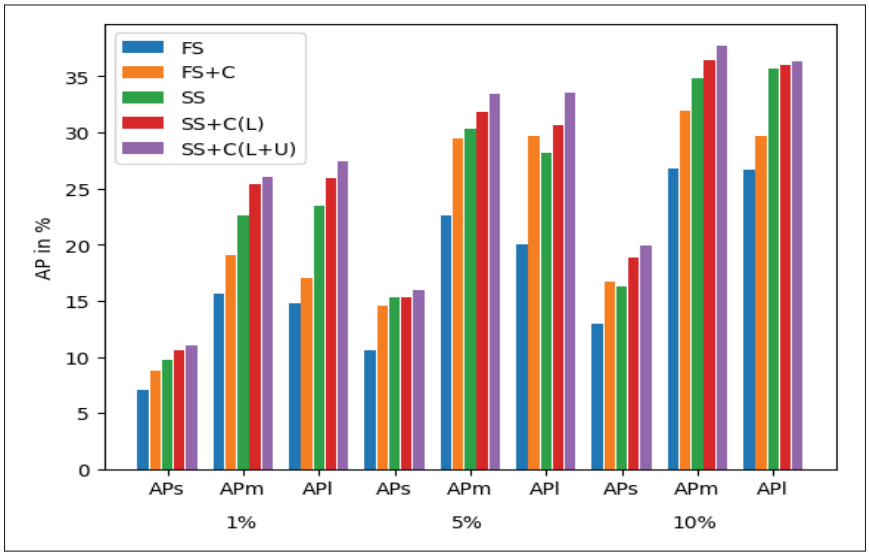


Figure 4.9 Detection AP of small, medium, and large objects with different percentages of supervised data on the VisDrone dataset. FS: fully supervised, FS+C: fully supervised with crops, SS: vanilla mean-teacher, SS+C: mean-teacher with density crops on labeled images, SS+C+U: mean-teacher with density crops on all images

We further verified this observation by conducting the same type of study in the satellite images of the DOTA dataset. Table 4.8 shows the results. The magnitude of improvements is comparable to that of the VisDrone dataset. AP shows an average improvement above 2% compared to the mean-teacher method. AP_{50} has a gain of more than 3% in this dataset compared to the mean-teacher. Also, the APs of small, medium, and large objects are studied in the same way as above. Figure 4.10 shows the results. APs of small, medium, and large objects with 100% supervised data on the DOTA dataset are 15.66, 38.16, and 44.2 respectively. While for small objects, our method with 10% labeled data is 3% below the supervised upper bound, the gap is around 10% for medium and large objects. This implies the boost from the density-guided training is more concentrated on the small objects. All of these experiments confirm the impact of each component in our model as well. The performance gain with our density-guided semi-supervised detector over the supervised baseline is significant and consistent.

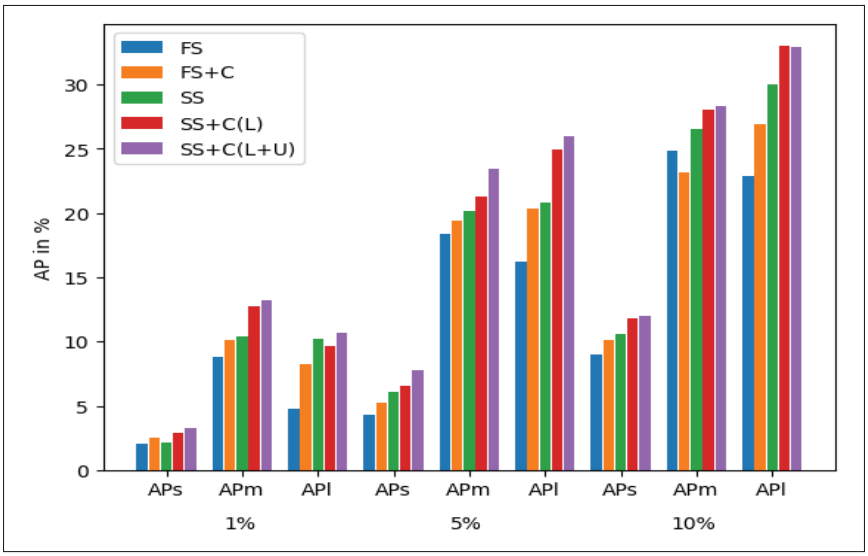


Figure 4.10 Detection AP of small, medium, and large objects with different percentages of supervised data on the DOTA dataset. FS: fully supervised, FS+C: fully supervised with crops, SS: vanilla mean-teacher, SS+C: mean-teacher with density crops on labeled images, SS+C+U: mean-teacher with density crops on all images

We also produced a qualitative comparison of the detection results from our semi-supervised model with that of its supervised baseline. Figure 4.11 shows the comparison on the DOTA (top

Table 4.8 Performance comparison of our density crop guided semi-supervised object detection with 1%, 5%, and 10% labeled images on the DOTA dataset. SSOD - semi-supervised detection with mean-teacher, Dcrop(L) - density crops on the labeled images, Dcrop (L + U) - density crops on the labeled and unlabeled images

Settings	1% (#Labeled=14)		5% (#Labeled=71)		10% (#Labeled =141)	
	AP	**AP**$_{50}$	**AP**	**AP**$_{50}$	**AP**	**AP**$_{50}$
Supervised	5.6±0.3	12.4±0.5	14.6±0.1	25.5±0.2	19.2±0.2	34.4±0.4
Supervised + Crop	6.8±0.2	14.2±0.4	16.0±0.2	29.0±0.4	20.4±0.2	36.5±0.5
SSOD	8.9±0.2	16.2±0.4	16.8±0.1	29.6±0.2	23.2±0.1	39.5±0.3
SSOD + Crop (L)	9.7±0.2	18.4±0.3	18.4±0.2	31.6±0.3	24.3±0.1	42.3±0.1
SSOD + Crop (L + U)	**10.3**±0.2	**19.7**±0.2	**20.0**±0.1	**34.8**±0.2	**25.2**±0.1	**43.1**±0.2

two rows) and VisDrone (bottom two rows) datasets. The supervised baseline is shown at the top and the semi-supervised results at the bottom among each pair of rows. We can see that many tiny objects are getting detected with our density-guided semi-supervised detector. In the case of VisDrone datasets, the baseline detector is missing most of the small objects at the farther end of the camera, whereas our method with zoom-in capability is discovering them. In the DOTA dataset, the missing happens at a much higher rate as the images are very high in pixel resolution. Objects like small cars are mostly missed by the baseline detector on the DOTA dataset. But our method shows impressive results in detecting them.

4.5.2 Comparison with other semi-supervised detectors

As other density-based approaches for small object detection use an external module (and multi-stage training) for crop extraction, we cannot adapt them to the semi-supervised settings with mean-teacher. So, we choose the recently proposed scale-aware detection QueryDet (Yang *et al.*, 2022) as it also accelerates small object detection with a detector itself. In particular, they proposed sparse querying on the high-resolution feature maps to improve small object detection. This is implemented on the feature pyramids within a detector, so we can wrap the mean-teacher training on top of this method. We used the VisDrone dataset with 10% labels in this study. The result is shown in table 4.9. Our method has an AP of more than 7% compared to the QueryDet semi-supervised detector. The AP$_s$ is improved by 7% whereas AP$_m$, AP$_l$ has

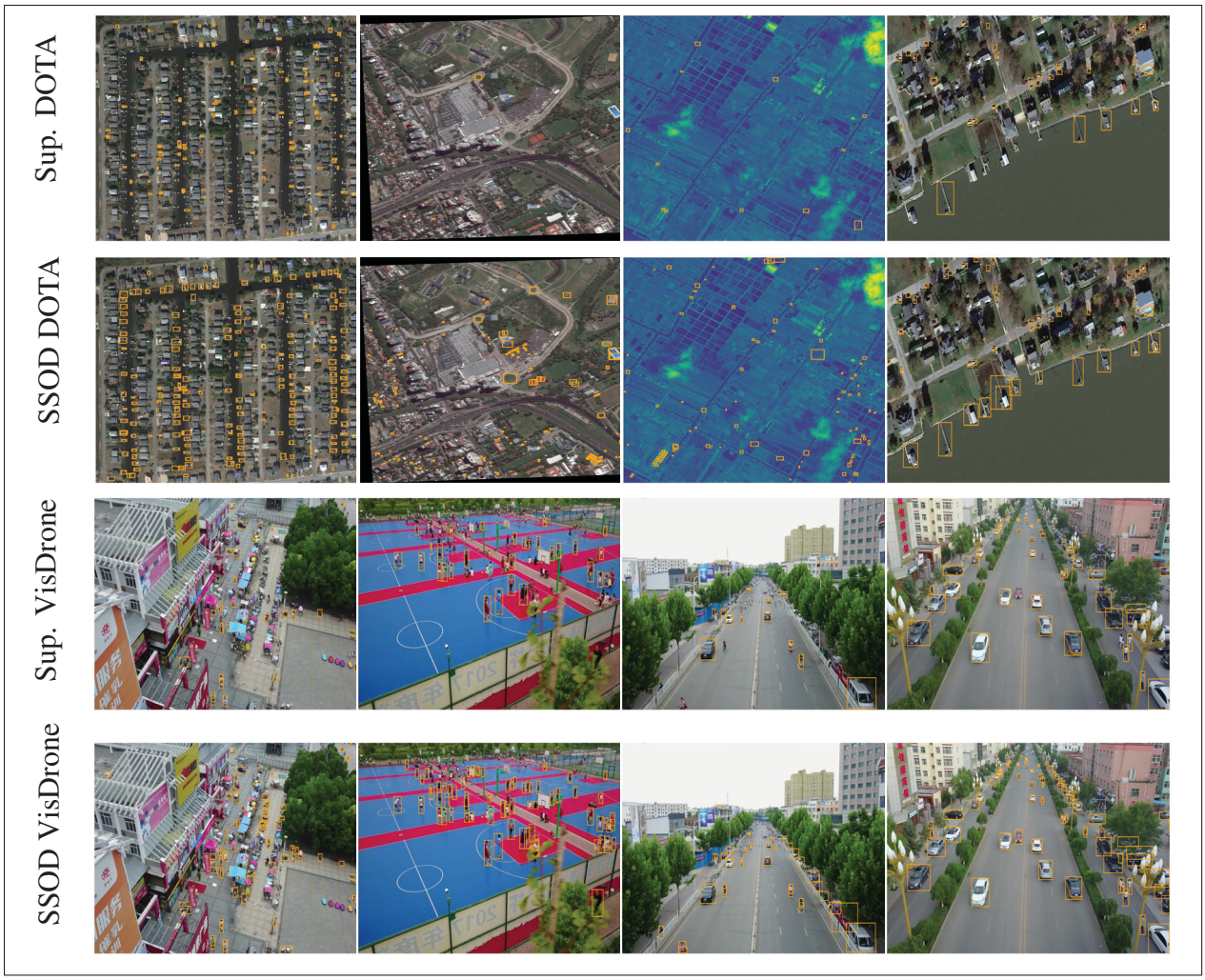


Figure 4.11 Qualitative comparison of detection results between supervised baseline and semi-supervised detector trained with density crops. More objects are detected with our semi-supervised zoom-in detector, especially the small ones

an improvement of more than 10%. While the semi-supervised QDet has an improvement of 3% over its supervised baseline, our method has an improvement of 8% over the supervised baseline. Note that the supervised baselines are different here because QueryDet proposed a method specific to the RetinaNet (Lin *et al.*, 2017a) detector. This study establishes the superiority of density-based detection over scale-aware training as well.

Table 4.9 Performance comparison with QueryDet method for small object detection in the semi-supervised settings using 10% labeled images on the VisDrone dataset

Settings	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Sup. QDet	16.58	31.13	15.45	10.89	23.93	23.46
Sup. Ours	19.26	37.73	17.48	12.94	26.85	26.65
QDet SSOD	19.56	35.78	18.67	13.90	26.17	30.53
Ours	**27.46**	**48.95**	**26.92**	**19.88**	**37.73**	**36.31**

4.5.3 Inference

The inference with density crops can be performed in two ways; taking the crop prediction directly from the model or running the cluster labeling algorithm with output detections. While the crop predictions are fast for inference, we observed that running the cluster labeling algorithm on the detection output is slightly more accurate. So one can choose the inference procedure among the two based on the speed vs accuracy trade-off of the downstream application. In the results reported so far, we used crop predictions directly from the model. To compare the performance of both we performed inference in two ways and reported the performance in table 4.10. In this study, the VisDrone data set with 10% labels is used. We can observe that while the improvement is small in AP, AP_{50} has a gain of more than 1%. We can also see that crop-labeled inference is improving the AP of small objects significantly, but at the same time, the AP of medium and large objects is declining. As the data set is dominated by small objects, we still observe an overall improvement in performance. We also reported the detection speed in Frames Per Second (FPS). The FPS is only reduced by 5 frames when the expensive crop-labeled inference is used.

Table 4.10 Results comparison of inference with predicted crops vs labeled crops based on prediction. The Visdrone dataset with 10% labels is used in the study

Settings	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
Inference with Predicted crops	27.46	48.95	26.92	19.88	37.73	36.31	12.45
Inference with labeled crops	27.78	50.02	27.12	20.99	36.32	35.84	7.17

4.5.4 Comparison with the supervised upper-bound

In table 4.11, we compare the results of our semi-supervised model with the fully supervised upper bound where 100% images are labeled. The setting used here is 10% labeled images. The lower bound of the performance when only the available 10% labeled data is also provided. It can be observed that our method with 10% labeled data is approximately 6% points close to the upper bound, both in the AP and AP_s. AP_m and AP_l are also showing a similar trend. Therefore, it can be concluded that, by effectively leveraging unlabeled data, our method is able to achieve a performance close to the fully supervised upper bound, while using minimal labeled data points.

Table 4.11 Performance comparison with the fully supervised upper-bound on the VisDrone dataset with 10% labeled images

Settings	**AP**	**AP_{50}**	**AP_{75}**	**AP_s**	**AP_m**	**AP_l**
Lower bound 10% Labeled	19.26	37.73	17.48	12.94	26.85	26.65
Semi-supervised Ours	27.46	48.95	26.92	19.88	37.73	36.31
Upper bound 100% Labeled	33.22	58.30	33.16	26.06	42.58	43.36

4.5.5 Computational cost

Using unlabeled data for mean-teacher training comes with additional training costs. Exponentially averaged teacher weights must be learned with a small α value to have stable distillation. We used a 0.9996 following the standard practices (Liu *et al.*, 2021a; Tang *et al.*, 2021). This results in many iterations for the mean-teacher training. In table 4.12, we compared the training iterations and time for different settings. Inference time per image is also provided. Finding crops on unlabeled images is performed only after the pseudo labels on unlabeled images are converged. The augmentation then adds an additional set of crops to the training process. That is why the SS+C (L+U) setting is taking longer iterations. For inference, the difference when using crops is due to the second detection performed on the crops. Even though there is an effective increase in training and inference time, the improvement in detection performance is significant.

Table 4.12 Comparison of the training and test time for fully supervised and semi-supervised methods with and without density crops. All settings are evaluated using one A100 GPU with the Visdrone dataset having 10% labels

Settings	FS	FS+C	SS	SS+C (L)	SS+C (L+U)
Train iters	5k	15k	65k	75k	180k
Train time in HH:MM	1:03	2:28	15:36	15:19	33:35
Test time in s/image	0.0348	0.0661	0.0348	0.0661	0.0661

4.5.6 Analysis of the type of errors

To understand how the addition of semi-supervised learning and density crops affects the detector's abilities, we profiled different error types based on the TIDE (Bolya *et al.*, 2020) evaluation protocol. Figure 4.12 shows the comparison results. With the addition of density crops on a supervised detector, we observe the localization error reduced. Other types of errors remain mostly the same. With semi-supervised training using the vanilla mean-teacher method, the classification error is reduced. Using density crops with semi-supervised learning is reducing the localization error similar to the fully-supervised case and other errors remain the same mostly. Compared to fully supervised detectors, semi-supervised detectors reduce classification error, but they tend to miss objects too. This is probably due to the imbalance in object classes of this dataset such that dominant classes get more pseudo-labels on unlabeled images. This can result in rare class objects being missed on the unlabeled images.

4.6 Conclusion

In summary, our proposed CZ Detector is observed effective in improving small object detection both in fully supervised and semi-supervised settings. It is easy to use like a plug-and-play module due to the fact that it can be embedded within any detection architecture. This is bringing the simplicity of uniform cropping or sliding window approach which is widely used by practitioners for training fully supervised detectors on aerial images. The density cropping approach that researchers proposed in the past is difficult to use in practice due to additional learnable components for density extraction, change in loss functions, and multi-stage training.

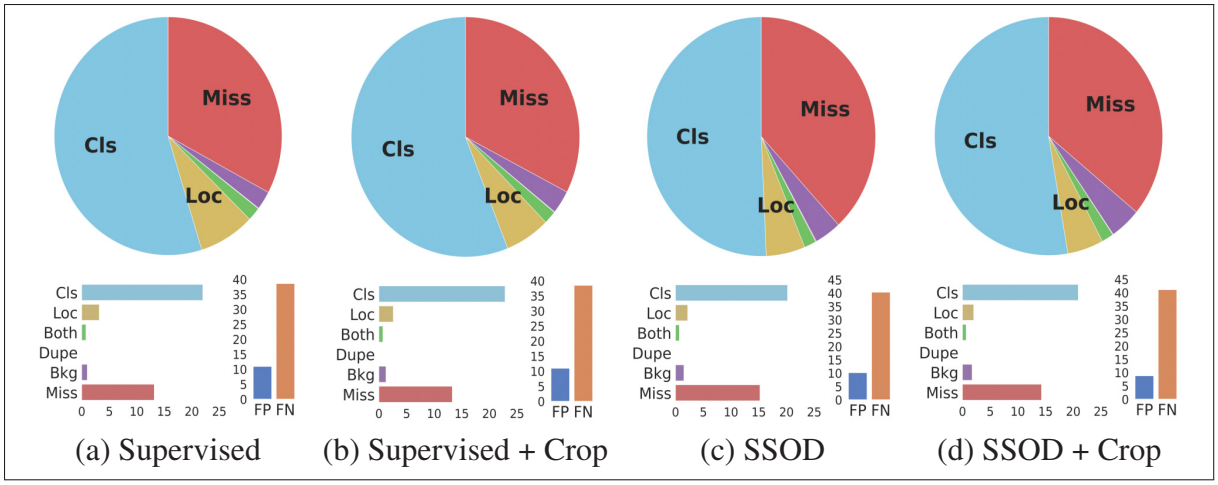


Figure 4.12 TIDE evaluation of detection results of the detectors trained with (a) supervised, (b) supervised with density crops, (c) vanilla semi-supervised, and (d) semi-supervised with density crops modes. Error types are: **Cls**: localized correctly but classified incorrectly, **Loc**: classified correctly but localized incorrectly, **Both**: both cls and loc error, **Dupe**: duplicate detection error, **Bkg**: detected background as foreground, **Miss**: missed ground truth error

The training step of the CZ Detector simply adds an additional class called "density crop" to the detector whose labels are obtained from a crop labeling algorithm. The inference is performed in two steps, one on the original image and then on the up-scaled version of the high-quality crops detected from it. For both modifications, we re-purpose the original detector similar to the uniform cropping.

Our proposed adaptation of the mean-teacher semi-supervised method to high-resolution aerial images for the detection of small objects with density crops is also shown to be effective. This is achieved by identifying the clusters of small objects from labeled and unlabeled images and processing them in higher resolution. For the labeled images, the original ground truth is used for cluster identification, whereas the pseudo ground-truth labels from the mean-teacher detector are used on unlabeled images. The embedding of the cluster identification within the detector made it possible to wrap the mean-teacher training on top of it. The clusters identified are cropped and used to augment the training set. The training with augmented crops is producing more pseudo-labels than the vanilla mean-teacher. This translates to improved detection performance.

The inference is performed on the original image and crops of clusters obtained on it to boost the small object detection. Empirical studies on the popular benchmark datasets reveal the superiority of our method over supervised training and vanilla mean-teacher training. We also find more boost in performance for density-based approaches than the scale-aware training with the mean-teacher method for small object detection.

CONCLUSION AND RECOMMENDATIONS

The contributions of this thesis explored techniques for localizing objects under reduced supervision settings, particularly weakly supervised and semi-supervised. In weakly supervised settings, for localization (a single object), the explainability-centric architecture was the dominant one. We observed that localization should be learned for accurate bounding boxes and proposed a localization-centric architecture. For detection (multiple objects), we proposed a learning scheme that facilitates the reusing of existing fully supervised detectors. This avoids the need for customized detection architectures for weakly supervised object detection. Thus, in summary, our contributions to the weakly supervised settings were innovations in architecture. For the semi-supervised methods, notable success has been achieved with natural images, but the same is not easy to translate into other images. We observed the difficulty in using the popular mean-teacher semi-supervised detectors on high-resolution aerial images with tiny objects. Applications using aerial images usually collect a significant amount of unlabeled data as they are usually deployed in surveillance mode. We proposed semi-supervised learning with density crops on these images to improve the localization. Particularly, the localization of tiny objects showed significant improvements.

The first takeaway of this thesis is that localization cannot be merely obtained as a byproduct of the classification activation. While such localization might be sufficient for explainability, for precise localization, a classification network alone is not sufficient. Instead, a separate component has to be used to learn for localization. We highlight that this is the reason why methods proposed on top of the explainability-centric architecture CAM use erase and learn techniques to produce integral object localization. From this observation, we argue that for better localization we should use a localization-centric architecture which has explicit components for learning localization. To this end, we propose to use spatial transformers in a convolutional fashion for learning localization. The localization network in the spatial transformer learns the required localization. The entire architecture is designed in a fully convolutional fashion. A

common issue when localizing objects with weak labels is the resulting localization focusing on the discriminative image regions. In our method, this is easily handled by regularizing the spatial transformation parameters by minimizing their distance with reference transforms.

A second takeaway from this research is that for weakly supervised object detection, customized architecture with MIL pooling is not required. One can use the existing fully supervised detection architectures and train them on weakly labeled datasets. Our proposed sampling and score propagation recipe will do the translation process to train fully supervised detectors on weak labels. Within this recipe many customizations are possible. The object proposals can be obtained using unsupervised localization results from modern foundation models. Other sampling distributions can be used, not necessarily the multinomial distribution. The score propagation logic can be altered corresponding to the RoI to ground-truth matching process used in the backbone detector. In summary, many customizations are possible in this basic sampling and score propagation recipe proposed here. It is also interesting to see that with this new architecture, both the bounding box supervision and image label supervision can be used in the same way. Thus at one end, we can easily use the existing classification datasets for detection because we do the same training and testing process as the conventional object detectors. On the other end, bonding box supervision can be easily injected to make the model better. So we hope in the future, training object detectors with both classification and detection datasets will be possible.

Finally, we hope our studies on aerial images shed light on the possibilities of semi-supervised learning on these images. Practical applications often collect large amounts of images but only annotate a fraction of them to avoid annotation costs. Then they train the object detection model using only a small set of annotated images. But the unlabeled images are vast in number and they are simply not utilized in the learning stages. This is where semi-supervised training of object detectors is important. While for natural images the literature is huge with many techniques

for training semi-supervised detectors, they don't easily translate to particular applications. In applications using aerial imagery, we often deal with tiny objects on high-resolution images. As the standard detectors themselves struggle with localization here, additional modules to improve localization are often used. They work in a focus and detect fashion where clusters of small objects are identified and focused for accurate detection. However, this setup is not easy to train in semi-supervised settings. To mitigate this, we proposed to identify the focus regions with the help of a detector alone, not using additional modules. Then focus regions are identified on labeled and unlabeled images are used to augment the training set. The detector is then trained in a semi-supervised fashion using the mean-teacher method. Inference is performed in two stages where the detection of the focused region constitutes the second stage. This system improved the semi-supervised object detection on aerial images significantly.

We presented an array of techniques for improving localization when training with reduced supervision. We hope the findings of this thesis will be valuable to the community. In particular, we recommend more research opportunities in the following findings. First, it brings an exciting line of possibilities when classification data and detection data can be used together for training object detectors. Our sampling-based training recipe can act as a strong proof of concept opening doors to this possibility. We conjecture that for open-world type settings where the object classes are unlimited, this recipe can serve new designs. The caption data from the web can provide unlimited noisy weak labels for open-world training. With existing detection datasets where annotations are available for their object classes, we can learn class-agnostic localization which will help other classes without box annotations. It will be interesting to see how our vanilla sampler needs modifications when using noisy weak labels. We also hope our studies convey the importance of utilizing unlabeled data collected from the deployment of aerial images. They carry important information from which we can learn more about data distribution and hence use them for learning better decision boundaries. Our density crop-guided semi-supervised detector tailored for high-resolution aerial images with tiny objects can produce better localization of

them. It will be interesting to see how this density crop-guided semi-supervised learning system translates to other detector families such as one-stage, anchor-free, and set prediction-based detectors.

BIBLIOGRAPHY

Alexe, B., Deselaers, T. & Ferrari, V. (2010). What is an object? *CVPR*.

Antonelli, S., Avola, D., Cinque, L., Crisostomi, D., Foresti, G. L., Galasso, F., Marini, M. R., Mecca, A. & Pannone, D. (2022). Few-Shot Object Detection: A Survey. *ACM Computing Surveys (CSUR)*, 1-37.

Bearman, A. L., Russakovsky, O., Ferrari, V. & Fei-Fei, L. (2015). What's the Point: Semantic Segmentation with Point Supervision. *ECCV*.

Belharbi, S., Sarraf, A., Pedersoli, M., Ben Ayed, I., McCaffrey, L. & Granger, E. (2022). F-CAM: Full Resolution Class Activation Maps via Guided Parametric Upscaling. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Belharbi, S., Ben Ayed, I., McCaffrey, L. & Granger, E. (2023). TCAM: Temporal Class Activation Maps for Object Localization in Weakly-Labeled Unconstrained Videos. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Bell, S., Zitnick, C. L., Bala, K. & Girshick, R. (2016). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *CVPR*.

Bilen, H. & Vedaldi, A. (2016). Weakly Supervised Deep Detection Networks. *CVPR*.

Bolya, D., Foley, S., Hays, J. & Hoffman, J. (2020). TIDE: A General Toolbox for Identifying Object Detection Errors. *ECCV*.

Cai, Z. & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. *CVPR*.

Caillouet, C., Giroire, F. & Razafindralambo, T. (2019). Efficient data collection and tracking with flying drones. *Ad Hoc Networks*, 35-46. doi: https://doi.org/10.1016/j.adhoc.2019.01.011.

CaiZhaowei, Z., Fan, C., Feris, R. S. & Vasconcelos, N. (2016). A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. *ECCV*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *ECCV*.

Chattopadhay, A., Sarkar, A., Howlader, P. & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)*. doi: 10.1109/WACV.2018.00097.

Chen, B., Li, P., Chen, X., Wang, B., Zhang, L. & Hua, X. (2022). Dense Learning based Semi-Supervised Object Detection. *CVPR*.

Chen, C., Seff, A., Kornhauser, A. & Xiao, J. (2015). DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. *ICCV*.

Chen, L., Yang, T., Zhang, X., Zhang, W. & Sun, J. (2021). Points as Queries: Weakly Semi-supervised Object Detection by Points. *CVPR*.

Cheng, G., Zhou, P. & Han, J. (2016). Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. *CVPR*.

Cheng, G., Yang, J., Gao, D., Guo, L. & Han, J. (2020). High-Quality Proposals for Weakly Supervised Object Detection. *IEEE Transactions on Image Processing*, 5794-5804.

Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q. & Han, J. (2023). Towards Large-Scale Small Object Detection: Survey and Benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-20.

Choe, J. & Shim, H. (2019). Attention-based Dropout Layer for Weakly Supervised Object Localization. *CVPR*.

Choe., J., Oh, S. J., Lee, S., Chun, S., Akata, Z. & Shim, H. (2020). Evaluating Weakly Supervised Object Localization Methods Right. *CVPR*.

Cinbis, R. G., Verbeek, J. J. & Schmid, C. (2016). Weakly supervised object localization with multi-fold multiple instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 189-203.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H. & Wei, Y. (2017). Deformable Convolutional Networks. *ICCV*.

Dai, J., Li, Y., He, K. & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *NeurIPS*.

Dalal, N. & Triggs, B. (2005a). Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2005.177.

Dalal, N. & Triggs, B. (2005b). Histograms of oriented gradients for human detection. *CVPR*. doi: 10.1109/CVPR.2005.177.

Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), 141–142.

Deng, S., Li, S., Xie, K., Song, W., Liao, X., Hao, A. & Qin, H. (2020). A global-local self-adaptive network for drone-view object detection. *IEEE Transactions on Image Processing(TIP)*, 1556-1569.

Detlefsen, N. S. & Hauberg, S. (2019). Explicit Disentanglement of Appearance and Perspective in Generative Models. *NeurIPS*.

Diba, A., Sharma, V., Pazandeh, A. M., Pirsiavash, H. & Gool, L. V. (2017). Weakly Supervised Cascaded Convolutional Networks. *CVPR*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Du, X., Wang, X., Gozum, G. & Li, Y. (2022). Unknown-Aware Object Detection: Learning What You Don't Know from Videos in the Wild. *Computer Vision and Pattern Recognition (CVPR)*, pp. 13668-13678. doi: 10.1109/CVPR52688.2022.01331.

Duan, C., Wei, Z., Zhang, C., Qu, S. & Wang, H. (2021). Coarse-grained density map guided object detection in aerial images. *ICCVW*.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. & Tian, Q. (2019). CenterNet: Keypoint Triplets for Object Detection. *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Etten, A. V. (2018). You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery. *CoRR*, abs/1805.09512, 1-8. Retrieved from: http://arxiv.org/abs/1805.09512.

Everingham, M., Gool, L. V., Williams, C. K., Winn, J. & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*, 303–338.

Fang, S., Cao, Y., Wang, X., Chen, K., Lin, D. & Zhang, W. (2021). WSSOD: A New Pipeline for Weakly- and Semi-Supervised Object Detection. *arXiv: 2105.11293*, 1-10.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-20.

Feng, C., Zhong, Y. & Huang, W. (2021). Exploring Classification Equilibrium in Long-Tailed Object Detection. *ICCV*.

Girshick, R. (2015). Fast R-CNN. *ICCV*.

Girshick, R. B., Donahue, J., Darrell, T. & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016). *Deep learning*. MIT Press.

Guo, Q., Mu, Y., Chen, J., Wang, T., Yu, Y. & Luo, P. (2022). Scale-Equivalent Distillation for Semi-Supervised Object Detection. *CVPR*.

Gupta, A., Dollar, P. & Girshick, R. (2019). LVIS: A Dataset for Large Vocabulary Instance Segmentation. *CVPR*.

Han, J., Ding, J., Xue, N. & Xia, G. (2021). ReDet: A Rotation-Equivariant Detector for Aerial Object Detection. *CVPR*.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. *CVPR*.

Huang, Z., Zou, Y., Kumar, B. & Huang, D. (2020). Comprehensive Attention Self-Distillation for Weakly-Supervised Object Detection. *NeurIPS*.

Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ICML*.

Jaderberg, M., Simonyan, K., Zisserman, A. & Kavukcuoglu, K. (2015). Spatial transformer networks. *NeurIPS*.

Jeon, Y. & Kim, J. (2017). Active Convolution: Learning the Shape of Convolution for Image Classification. *CVPR*.

Jeong, J., Lee, S., Kim, J. & Kwak, N. (2019). Consistency-based semi-supervised learning for object detection. *NeurIPS*.

Jeong, J., Verma, V., Hyun, M., Kannala, J. & Kwak, N. (2021). Interpolation-based Semi-supervised Learning for Object Detection. *CVPR*.

Jie, Z., Wei, Y., Jin, X., Feng, J. & Liu, W. (2017). Deep Self-Taught Learning for Weakly Supervised Object Localization. *CVPR*, pp. 4294-4302. doi: 10.1109/CVPR.2017.457.

Johnson, J., Karpathy, A. & Fei-Fei, L. (2016). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. *CVPR*.

Kantorov, V., Oquab, M., Cho, M. & Laptev, I. (2016). ContextLocNet: Context-aware Deep Network Models for Weakly Supervised Localization. *ECCV*.

Köhler, M., Eisenbach, M. & Gross, H. (2021). Few-Shot Object Detection: A Comprehensive Survey. *TNNLS*, 1-21.

Kosiorek, A., Sabour, S., Teh, Y. & Hinton, G. (2019). Stacked Capsule Autoencoders. *NeurIPS*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, pp. 1097–1105.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T. & Ferrari, V. (2020). The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 1956–1981.

Lacoste, A., Sherwin, E., Kerner, H., Alemohammad, H., Lutjens, B., Irvin, J., Dao, D., Chang, A., Gunturkun, M., Drouin, A., Rodriguez, P. & Vazquez, D. (2021). Toward Foundation Models for Earth Monitoring: Proposal for a Climate Change Benchmark. *NeurIPS*.

Li, A., Yuan, P. & Li, Z. (2022a). Semi-Supervised Object Detection via Multi-instance Alignment with Global Class Prototypes. *CVPR*.

Li, C., Yang, T., Zhu, S., Chen, C. & Guan, S. (2020, June). Density Map Guided Object Detection in Aerial Images. *CVPRW*.

Li, D., Huang, J.-B., Li, Y., Wang, S. & Yang, M.-H. (2016). Weakly Supervised Object Localization with Progressive Domain Adaptation. *CVPR*, pp. 3512-3520. doi: 10.1109/CVPR.2016.382.

Li, J., Liang, X., Wei, Y., Xu, T., J.Feng & Yan, S. (2017). Perceptual generative adversarial networks for small object detection. *CVPR*.

Li, Y., Chen, Y., Wang, N. & Zhang, Z. (2019). Scale-aware trident networks for object detection. *ICCV*.

Li, Y., Fan, B., Zhang, W., Ding, W. & Yin, J. (2021). Deep active learning for object detection. *Journal of Information Sciences*, 418-433.

Li, Y.-J., Dai, X., Ma, C.-Y., Liu, Y.-C., Chen, K., Wu, B., He, Z., Kitani, K. & Vajda, P. (2022b). Cross-Domain Adaptive Teacher for Object Detection. *CVPR*.

Lin, C.-H., Yumber, E., Wang, O., Shechtman, E. & Lucey, S. (2018). ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing. *CVPR*.

Lin, T., Goyal, P., Girshick, R., He, K. & Dollar, P. (2017a). Focal loss for dense object detection. *ICCV*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P. & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *ECCV*.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B. & Belongie, S. J. (2017b). Feature pyramid networks for object detection. *CVPR*.

Linardatos P,Papastefanopoulos V, K. S. (2020). Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy (Basel)*, 1-22. doi: 10.3390/e23010018.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. & Pietikäinen, M. (2019). Deep Learning for Generic Object Detection: A Survey. *IJCV*, 261–318.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C. & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *ECCV*.

Liu, Y.-C., Ma, C.-Y., He, Z., Kuo, C.-W., Chen, K., Zhang, P., Wu, B., Kira, Z. & Vajda, P. (2021a). Unbiased Teacher for Semi-Supervised Object Detection. *ICLR*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021b). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *ICCV*.

Long, J., Shelhamer, E. & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *CVPR*, pp. 3431-3440. doi: 10.1109/CVPR.2015.7298965.

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E. & Wen, S. (2020). PP-YOLO: An Effective and Efficient Implementation of Object Detector. Retrieved from: https://arxiv.org/abs/2007.12099.

Long, Y., Gong, Y., Xiao, Z. & Liu, Q. (2017). Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 2486-2498.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the seventh IEEE international conference on computer vision*.

Mai, J., Yang, M. & Luo, W. (2020). Erasing Integrated Learning: A Simple Yet Effective Approach for Weakly Supervised Object Localization. *CVPR*.

Meethal, A., Pedersoli, M., Zhu, Z., Romero, F. P. & Granger, E. (2022). Semi-Weakly Supervised Object Detection by Sampling Pseudo Ground-Truth Boxes. *IJCNN*.

Mi, P., Lin, J., Zhou, Y., Shen, Y., Luo, G., Sun, X., Cao, L., Fu, R., Xu, Q. & Ji, R. (2022). Active Teacher for Semi-Supervised Object Detection. *CVPR*.

Oksuz, K., Cam, B. C., Kalkan, S. & Akbas, E. (2021). Imbalance Problems in Object Detection: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3388-3415. doi: 10.1109/TPAMI.2020.2981890.

Oquab, M., Bottou, L., Laptev, I. & Sivic, J. (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks. *CVPR*.

Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J. & Zhang, C. (2021). DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection. *ICCV*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. Retrieved from: https://arxiv.org/abs/2103.00020.

Redmon, J. & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *CVPR*.

Redmon, J., Divvala, S. K., Girshick, R. B. & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *CVPR*.

Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *NeurIPS*.

Ren, Z., Yu, Z., Yang, X., Liu, M., Lee, Y. J., Schwing, A. G. & Kautz, J. (2020). Instance-Aware, Context-Focused, and Memory-Efficient Weakly Supervised Object Detection. *CVPR*.

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *International Conference on Knowledge Discovery and Data Mining SIGKDD*.

Roy, S., Unmesh, A. & Namboodiri, V. P. (2018). Deep active learning for object detection. *BMVC*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 211-252.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *ICCV*.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*.

Shen, Y., Ji, R., Zhang, S., Zuo, W. & Wang, Y. (2018). Generative Adversarial Learning Towards Fast Weakly Supervised Detection. *CVPR*.

Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*.

Singh, B. & Davis, L. S. (2018). An analysis of scale invariance in object detection snip. *CVPR*.

Singh, B., Najibi, M. & Davis, L. S. (2018). Sniper: Efficient multi-scale training. *NeurIPS*.

Singh, K. K. & Lee, Y. J. (2017). Hide-and-Seek: Forcing a network to be Meticulous for Weakly-supervised Object and Action Localization. *ICCV*.

Sohn, K., Zhang, Z., Li, C., Zhang, H., Lee, C. & Pfister, T. (2020). A simple semi-supervised learning framework for object detection.

Song, J., Jeong, J.-H., Park, D.-S., Kim, H.-H., Seo, D.-C. & Ye, J. C. (2021). Unsupervised Denoising for Satellite Imagery Using Wavelet Directional CycleGAN. *IEEE Transactions on Geoscience and Remote Sensing*, 6823-6839. doi: 10.1109/TGRS.2020.3025601.

Sun, Y., Shao, Z., Cheng, G., Huang, X. & Wang, Z. (2022). Road and Car Extraction Using UAV Images via Efficient Dual Contextual Parsing Network. *IEEE Transactions on Geoscience and Remote Sensing*, 1-13. doi: 10.1109/TGRS.2022.3214246.

Tang, P., Wang, X., Bai, X. & Liu, W. (2017). Multiple Instance Detection Network with Online Instance Classifier Refinement. *CVPR*.

Tang, P., Wang, X., Bai, S., Shen, W., Bai, X., Liu, W. & Yuille, A. (2018a). PCL: Proposal Cluster Learning for Weakly Supervised Object Detection. *IEEE TPAMI*, 1-16.

Tang, P., Wang, X., Bai, X. & Liu, W. (2018b). Fast Visual Object Tracking with Rotated Bounding Boxes. *CVPR*.

Tang, Y., Chen, W., Luo, Y. & Zhang, Y. (2021). Humble Teachers Teach Better Students for Semi-Supervised Object Detection. *CVPR*.

Tarvainen, A. & Valpola, H. (2018). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*.

Tian, Z., Shen, C., Chen, H. & He, T. (2019). FCOS: Fully convolutional one-stage object detection. *ICCV*.

van de Sande, K., Uijlings, J., Gevers, T. & Smeulders, A. (2011). Segmentation as selective search for object recognition. *ICCV*.

van der Velden, B. H., Kuijf, H. J., Gilhuijs, K. G. & Viergever, M. A. (2022). Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 102470. doi: https://doi.org/10.1016/j.media.2022.102470.

van Engelen, J. E. & Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine Learning*, 109, 373 - 440.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is All You Need. *NeurIPS*, pp. 6000–6010.

Vilone, G. & Longo, L. (2020). Explainable Artificial Intelligence: a Systematic Review.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR*. doi: 10.1109/CVPR.2001.990517.

Vo, H., Siméoni, O., Gidaris, S., Bursuc, A., Pérez, P. & Ponce, J. (2022). Active Learning Strategies for Weakly-Supervised Object Detection.

Wan, F., Wei, P., Jiao, J., Han, Z. & Ye, Q. (2018). Min-Entropy Latent Model for Weakly Supervised Object Detection. *CVPR*.

Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J. & Shen, C. (2018). Repulsion Loss: Detecting Pedestrians in a Crowd. *CVPR*.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S. & Perona, P. (2010). The Caltech-UCSD Birds-200-2011 Dataset. 1-8.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y. & Girshick, R. (2019). Detectron2.

Xia, G., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M. & Zhang, L. (2018). Dota: A large-scale dataset for object detection in aerial images. *CVPR*.

Xie, Q., Luong, M., Hovy, E. & Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. *CVPR*.

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X. & Liu, Z. (2021). End-to-End Semi-Supervised Object Detection with Soft Teacher. *ICCV*.

Yang, C., Huang, Z. & Wang, N. (2022). QueryDet: Cascaded Sparse Query for Accelerating High-Resolution Small Object Detection. *CVPR*.

Yang, F., Fan, H., Chu, P., Blasch, E. & Ling, H. (2019). Clustered Object Detection in Aerial Images. *ICCV*.

Zang, Y., Zhou, K., Huang, C. & Loy, C. C. (2023). Semi-Supervised and Long-Tailed Object Detection with CascadeMatch. *IJCV*, 987–1001.

Zhang, C. & Kim, J. (2019). Object detection with location-aware deformable convolution and backward attention filtering. *CVPR*.

Zhang, D., Han, J., Cheng, G. & Yang, M. (2021). Weakly Supervised Object Localization and Detection: A Survey. *TPAMI*, 5866-5885.

Zhang, X., Wei, Y., Feng, J., Yang, Y. & Huang, T. (2018a). Adversarial complementary learning for weakly supervised object localization. *CVPR*.

Zhang, X., Wei, Y., Kang, G., Yang, Y. & Huang, T. (2018b). Self-produced guidance for weaklysupervised object localization. *ECCV*.

Zhang, X., Izquierdo, E. & Chandramouli, K. (2019). Dense and small object detection in UAV vision based on cascade network. *ICCVW*.

Zhang, Y., Bai, Y., Ding, M., Li, Y. & Ghanem, B. (2018c). W2F: A Weakly-Supervised to Fully-Supervised Framework for Object Detection. *CVPR*.

Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L. & Li, Y. (2022). Regionclip: Region-based language-image pretraining. *CVPR*.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A. & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *CVPR*.

Zhou, P., Ni, B., Geng, C., Hu, J. & Xu, Y. (2018). Scale-Transferrable Object Detection. *CVPR*.

Zhou, X., Girdhar, R., Joulin, A. & andd I. Misra, P. K. (2022). Detecting Twenty-thousand Classes using Image-level Supervision. *ECCV*.

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *NSR*, 5, 44–53.

Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C. & Liu, X. (2018). Visdrone-det2018: The vision meets drone object detection in image challenge results. *ECCVW*.

Zhu, X. (2008). Semi-supervised learning literature survey. 1-40.

Zitnick, C. L. & Dollar, P. (2014). Edge boxes: Locating object proposals from edges. *ECCV*.

Zou, Z., Chen, K., Shi, Z., Guo, Y. & Ye, J. (2023). Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*, 111(3), 257-276. doi: 10.1109/JPROC.2023.3238524.