Feed-Forward Weakly Supervised Deep Learning Models for Breast Cancer Diagnosis from Histological Images

by

Jérôme Rony

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR A MASTER'S DEGREE
WITH THESIS IN SYSTEM ENGINEERING
M.A.Sc.

MONTREAL, FEBRUARY 20, 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Eric Granger, memorandum supervisor
System Engineering Department, École de technologie supérieure

M. Ismail Ben Ayed, co-supervisor
System Engineering Department, École de technologie supérieure

M. Mohamed Cheriet, president of the board of examiners
System Engineering Department, École de technologie supérieure

M. Carlos Vàzquez, external examiner
Software and Information Technology Engineering Department, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON NOVEMBER 15, 2019

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Eric Granger and co-advisor Ismail Ben Ayed for sharing their wisdom and experience in conducting high-level research. I learned much from them, and I really appreciate the discussions and the progress we made during these two years that we worked together.

Thanks for all the colleagues at LIVIA for their help on my research, for teaching me new things and for being good friends. Special thanks to Luiz Gustavo Hafemann, Le Thanh Nguyen Meidine, Hoel Kervadec, Saypraseuth Mounsaveng, Karthik Gopinath, Imtiaz Ziko and Malik Boudiaf. I would also like to thank professors Marco Pedersoli, Christian Desrosiers, Robert Sabourin and Mohammed Cheriet for all the discussions.

Modèles d'Apprentissage Profond Faiblement Supervisés pour le Diagnostic du Cancer du Sein à partir d'images histologiques

Jérôme Rony

RÉSUMÉ

Le dépistage du cancer du sein est devenu une priorité afin de réduire le nombre de décès causés par cette maladie et la traiter le plus tôt possible. Le meilleur moyen de confirmer un diagnostic du cancer du sein est l'analyse d'images histologiques. Cependant, l'analyse de ces images requiert du temps et une forte expertise pour produire un diagnostic correct. La recherche en algorithmes de vision par ordinateur a montré un fort potentiel pour assister les médecins afin d'effectuer des diagnostics rapides et précis. Toutefois, ce type d'algorithmes requiert un volume important de données avec des annotations de qualité.

Dans ce mémoire, nous proposons d'approcher ce problème par l'apprentissage faiblement supervisé. Dans ce scénario, seuls des annotations faibles tels que des étiquettes au niveau de l'image sont disponibles. Avec ces annotations faibles, nous souhaitons entraîner des modèles pouvant prédire avec précision les étiquettes d'images (classification) et de pixels (segmentation).

Comme première contribution, nous avons identifié, analysé et évalué plusieurs méthodes d'apprentissage faiblement supervisé originalement proposées pour des images naturelles afin d'identifier celles qui sont les plus prometteuses pour notre application.

Comme seconde contribution, nous avons généralisé une technique initialement proposée pour de la classification binaire pour les cas plus généraux de classification mutli-classes et multi-étiquettes. Nous avons montré que cette technique fonctionne sur des datasets d'images naturelles et ensuite évaluée sur des images histologiques.

Comme troisième contribution, nous avons étudié l'impact d'ajouter des contraintes de tailles pour l'entraînement des modèles. Cela correspond à un scénario où un annotateur estimerait seulement la taille des régions cancéreuses dans les images histologiques, réduisant le temps d'annotations requis pour effectuer une segmentation complète. Nous avons proposé une formulation qui permet d'utiliser cette information de taille et montré que cela augmente significativement les performances en segmentation comparé à ce qui peut être obtenu lors d'un entraînement avec des étiquettes au niveau de l'image seulement.

Mots-clés: Diagnostic du Cancer du Sein, Apprentissage faiblement supervisé, Apprentissage Profond

Feed-Forward Weakly Supervised Deep Learning Models for Breast Cancer Diagnosis from Histological Images

Jérôme Rony

ABSTRACT

Breast cancer screening has become one of the top priorities to reduce the number of deaths from breast cancer and treat it as soon as possible. When making a diagnosis for breast cancer, the gold standard is histological image analysis. However, these images are very large and require both time and expertise to be correctly interpreted. Research in computer vision algorithms has shown great potential to assist practitioners in making faster and more accurate diagnoses. However, to use these kinds of algorithms, large amounts of data with high quality annotations are often required.

In this thesis, we propose to approach this problem from the perspective of weakly supervised learning. In this scenario, we consider that we only have weak annotations such as image-level labels for each sample. With these weak annotations, we seek to train accurate models to accurately predict both image-level (classification) and pixel-level labels (segmentation).

As a first contribution, we identify, analyze and evaluate several weakly supervised techniques proposed for natural images with the intent of identifying which one are the most promising for our application.

As a second contribution, we generalize a technique proposed for binary classification to the more general settings of multi-class and multi-label classification. We show that this technique scales well to widely used natural image datasets and further evaluate it on histological image datasets.

As a third contribution, we study the impact of adding size constraints to train a model. This corresponds to a scenario where an annotator would only estimate the size of the cancerous regions in histological images, reducing the time needed to annotate images when doing a segmentation. We propose a formulation to take advantage of the size information and show that it significantly improves the segmentation performance compared to training with image-level labels only.

Keywords: Breast Cancer Diagnosis, Weakly Supervised Learning, Deep Learning

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

CAM	Class Activation Map
CNN	Convolutional Neural Network
DL	Deep Learning
EB	Edge Boxes
FN	False Negative
FP	False Positive
GAP	Global Average Pooling
H&E	Haemotoxylin and Eosin
KL	Kullback-Leibler
RP	Region Proposal
SGD	Stochastic Gradient Descent
SPP	Spatial Pyramid Pooling
SS	Selective Search
TN	True Negative
TP	True Positive
WSDDN	Weakly Supervised Deep Detection Network
WSI	Whole Slide Image
WSL	Weakly Supervised Learning
WSOD	Weakl Supervised Object Detection
WSOL	Weakl Supervised Object Localization

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathcal{D}$	Training set
θ	Learnable parameters in a model
x	Input sample (Image)
y	Label for sample x
C	Number of classes
$\boldsymbol{y}$	One-hot label for sample x
H^{in}	Image height
W^{in}	Image width
$\boldsymbol{f}$	Feature vector
$\boldsymbol{\mathsf{F}}$	Feature maps extracted from x using a CNN
$\boldsymbol{\mathsf{M}}$	Activation maps
H	Feature and activation maps height
W	Feature and activation maps width
K	Dimensionality of a feature vector
S	Total stride of a convolutional model
$\boldsymbol{s}$	Class score vector
η	Learning rate
σ	Sigmoid function
$\mathbf{I}$	Indicator function

δ	Adversarial perturbation
$\tilde{x}$	Adversarial example

INTRODUCTION

Since 2012, Deep Learning (DL) has seen a rapid development thanks to the significant progress made mainly in computer vision with the widespread use of Convolutional Neural Networks (CNN). Since then, they are being deployed in more and more real-world applications. In the medical field, imaging is an important tool for many diagnoses but experts are needed for accurate diagnoses. For breast cancer, one of the main tools of diagnosis is histology imaging, in which a small sample of tissue is taken from a patient and observed with a microscope.

Nowadays, samples are digitized with electronic microscopes, producing images of large sizes ($\sim 10^9$ pixels) which can be time-consuming to analyze due to their sheer size. In order to reduce the time spent by an expert analyzing the image, we can use computer vision tools such as DL models to perform a pre-diagnosis: obtaining a first classification and regions of interest for an expert to look and confirm or infirm the diagnosis.

However, the performance of these DL models highly depends on the data used to train them. There are two major aspects to consider when training DL models: the number of samples and the quality of the annotation(s). The latter defines the type of model that can be used to solve a task: classification, detection, segmentation, *etc*. For a practical application, we wish to train models that can correctly classify histological images and identify regions of interest related to the prediction. This scenario usually falls under the task of semantic segmentation where an annotation specifying the class of every pixel in the image is used to train the model. This type of annotation is expensive to obtain as they can take minutes or hours to create depending on the size of the original image. Therefore, Weakly Supervised Learning (WSL) has attracted a lot of interest since it aims at learning the same kinds of models trained with pixel-level annotations but with much cheaper annotations. Cheaper annotations include image-level labels, point-wise locations, scribbles, regions size estimations, *etc*.

Problem statement

The problem of WSL for breast cancer diagnosis can be modeled as a pattern recognition problem. Specifically, it involves two tasks: classification and segmentation. Given a dataset $\mathcal{D}$ of histological images and their associated annotations, the goal is to learn a model that can correctly classify the images in different categories considered. The most common case is binary classification where the considered cancer grades are *benign* and *malignant* but some datasets consider more classes. If we consider only image-level annotations, we have $\mathcal{D} = \{x^i, y^i\}_{i=1}^N$ where $x^i \in \mathbb{R}^{D \times H^{in} \times W^{in}}$ is a histological image and y^i is its associated label. The input is fed to a model that outputs a prediction $\boldsymbol{s} \in \mathbb{R}^C$ with C being the number of classes, which is compared to the true label to train the model. In the evaluation phase, the model outputs both a score vector $\boldsymbol{s}$ and Class Activation Maps (CAMs) $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$ where H and W are usually smaller than H^{in} and W^{in} because of the architecture of CNNs designed for image-classification. These CAMs predict which part of the image are associated to a given label (Oquab, Bottou, Laptev et al., 2015) and are compared to a ground-truth pixel-level label if available to evaluate the segmentation performance. The difficulty of learning a good segmentation model from image-level label lies in the absence of local information in the labels, meaning that a model must be designed to promote good local classification abilities.

Literature overview

Using state-of-the-art deep learning models for the computer-assisted diagnosis of diseases like cancer presents several challenges related to the nature and availability of labeled histology images. In particular, cancer grading and localization in these images normally relies on both image- and pixel-level labels, the latter requiring a costly annotation process. In this survey, deep weakly-supervised learning (WSL) models are investigated to identify and locate diseases in histology images, without the need for pixel-level annotations. Given training data with image-level labels, these models allow to simultaneously classify histology images and yield pixel-wise localization scores, thereby identifying the corresponding regions of interest. These models are organized into two main approaches that differ in their mechanism for building attention maps to localize salient regions – (1) bottom-up approaches based on forward-pass information through a network, either by spatial pooling of representations/scores, or by detecting class regions; and (2) top-down approaches based on backward-pass information within a network, inspired by human visual attention. Since relevant WSL models have mainly been investigated within the computer vision community, and validated on natural scene images, we assess the extent to which they apply to histology images which have challenging properties, *e.g*. very large size, non-salient and highly unstructured regions, stain heterogeneity, and coarse/ambiguous labels.

The most relevant deep WSL models (*e.g*., CAM, WILDCAT and Deep MIL) are compared experimentally in terms of accuracy (classification and pixel-wise localization) on several public benchmark histology datasets for breast and colon cancer (BACH ICIAR 2018, BreakHis, CAMELYON16, and GlaS). Furthermore, to benchmark large-scale evaluations of WSL methods for histology, we propose a protocol to build WSL datasets from Whole Slide Imaging, with publicly available deterministic code and coordinates of the sampled patches. Results indicate that several deep learning models, and in particular WILDCAT and deep MIL, can provide a high level of classification accuracy, although pixel-wise localization of cancer regions remains an issue for such images.

Challenges

In this section, we review the main challenges in WSL for histological images:

- **Learning a good classification model**: One of the most important aspects of being able to identify regions of interest is to correctly classify the images in the first place. DL models are known to require large amounts of data for training to be able to produce good predictions on unseen images. However, the availability of a large number of histological images is not always assured as the annotation cost is much higher than for natural images.

- **Different image characteristics**: Most of state-of-the-art computer vision algorithms are designed and validated on natural images but histological images present different features. This is especially true in the WSL scenario, where the most recent algorithms all use priors on natural images to increase their performance. One of the typical prior is to suppose that edges in an image define objects and enforce that the local predictions made by a model are coherent with the edges. Such priors may not hold for histological images.
- **Weak annotations**: The main challenge lies in the relation between the weak annotation(s) available for training and the dense information to predict. A pixel-level annotation contains a large quantity of information that we want to predict with a model trained using only information given for the entire image.

Research Objectives and Contributions

The main question of this work is what level of supervision is needed to learn a model from data that can predict regions of interest in histological images. More specifically, the main research objectives are :

- to determine what is the current state-of-the-art for weakly-supervised learning applied to images;
- to evaluate the state-of-the-art methods on histological images;
- to propose improvements over existing methods;
- to investigate the levels of supervision needed to obtain close to full-supervision performances.

The core contributions of this thesis are:

- In Chapter 1, we do an in-depth review on the methods proposed for WSL for computer vision with image-level labels only. Specifically, we study their formulation and their limitations.

- In Chapter 2, we evaluate the performances of the methods studied in Chapter 1 on histological images datasets. We show that most of the methods that can be used perform poorly compared to their fully-supervised (*i.e.* trained with pixel-level annotation) counterparts.

  Related publication:

  Deep weakly-supervised learning methods for classification and localization in histology images: a survey (submitted to *Medical Image Analysis*)

- In Chapter 3, we propose to generalize one of the most promising approach studied in Chapter 1 to the multi-class and multi-label scenarios. We validate that the method improves segmentation performance on natural images and evaluate it on histological images datasets and conclude that the problem of WSL on histological images requires more supervision to significantly improve performance.

- In Chapter 4, we study the impact of adding more supervision and propose a formulation to take into account a size information on the regions of interest. We simulate a scenario where an annotator would only provide a rough size estimation by adding noise to the ground truth size.

An additional contribution with LIVIA collaborators was made in the field of adversarial machine learning, by developing a fast algorithm to generate adversarial examples, and its usage for training defense mechanisms (joint work with colleagues, described in appendix I). This method was used to win one of the competitions of the NIPS 2018 Adversarial Vision Challenge (Brendel, Rauber, Kurakin, Papernot, Veliqi, Salathé, Mohanty & Bethge, 2018; Brendel, Rauber, Kurakin, Papernot, Veliqi, Mohanty, Laurent, Salathé, Bethge, Yu et al., 2020):

Decoupling Direction and Norm for Efficient Gradient-Based L2 Adversarial Attacks and Defenses (accepted to the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2019)*).

CHAPTER 1

A SURVEY OF DEEP WEAKLY-SUPERVISED LEARNING METHODS FOR CLASSIFICATION AND LOCALIZATION IN HISTOLOGY IMAGES

1.1 Introduction

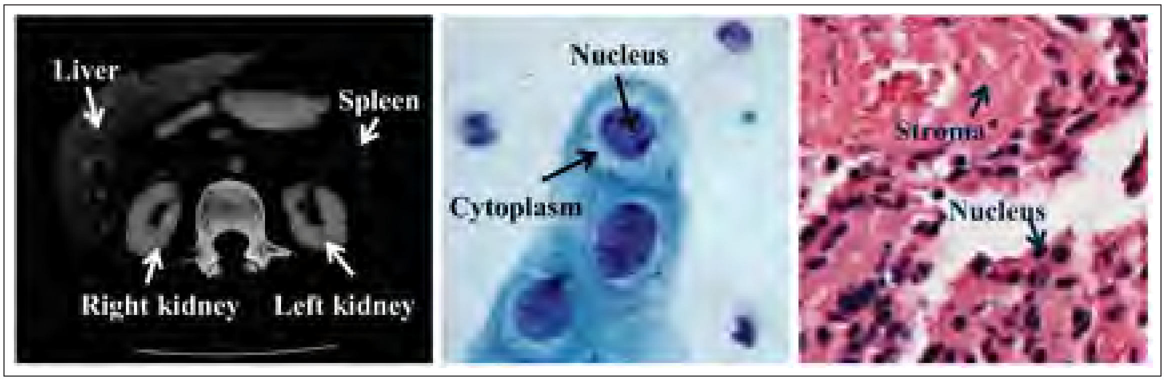


Figure 1.1 Image examples for radiology CT (left), cytology (middle), and histology (right) (Credit: (He *et al.*, 2010,1))

The advent of Whole Slide Imaging (WSI) scanners (He *et al.*, 2012), which can perform cost-effective and high-throughput digitization of histology slides, has opened new possibilities in pathology image analysis (He *et al.*, 2012; Madabhushi, 2009). Histology slides provide more comprehensive views of diseases and their effect on tissue (Hipp, Fernandez, Compton et al., 2011) since their preparation preserves the underlying tissue structure (He *et al.*, 2012). For instance, some disease characteristics (*e.g.* lymphatic infiltration of cancer) may be predicted using only histology images (Gurcan, Boucheron, Can et al., 2009). The analysis of a histology image remains the gold standard in diagnosing several diseases including most types of cancer (Gurcan *et al.*, 2009; He *et al.*, 2012; Veta, Pluim, Van Diest et al., 2014). Breast cancer is the most prevalent cancer in women worldwide, and medical imaging systems are a primary diagnosis tool for its early detection (Daisuke & Shumpei, 2018; Veta *et al.*, 2014; Xie, Liu, Joseph Luttrell et al., 2019).

Figure 1.1 shows examples of three different types of medical imaging: computed tomography (CT), cytology, and histology images. The different imaging techniques operate at different levels: CT and magnetic resonance image (MRI) at whole body and tissue level; histology at tissue and cell level; cytology at cellular level. Histology images differ from radiology images in having a large number of objects of interest (cells and cell structures, such as nuclei) widely distributed and surrounded by various tissue types (*e.g.* in the cervix, epithelium and stroma). In contrast, radiology image analysis usually focuses on a few organs in the image which tend to be more predictably located compared to histology images where objects constantly change location. A histology image usually has a size of $\sim 10^9$ pixels which is significantly larger than the size of a radiology image which typically has $\sim 10^5$ pixels. In addition, histology tissues are generally stained with different, but in the same palette, colors while radiology images usually contain only gray intensities. On the other hand, cytology images have some similarities to histology images; both have multiple cells distributed within the image. However, histology images are often taken at much lower level of magnification to allow analysis at the tissue level and identification of the boundary between tissue types. The level of magnification in histology images is sufficient for some analysis at the cell level such as nucleus counting but cannot provide the in-depth information of internal cell structure.

Cancer is mainly diagnosed by pathologists who must analyze WSIs, often identifying groups of cells organized in ducts or lobules within a heterogeneous stroma. Analyzing WSIs from digitized histology slides allows to facilitate and potentially automate Computer-Aided Diagnosis (CAD) in pathology, where the main goal is to confirm the presence or absence of disease and to grade or measure disease progression. The widespread use of CAD can be traced back to the emergence of digital mammography in the 1990s (Méndez, Tahoces, Lado et al., 1998). Since a large number of digitized exams have been collected, CAD has become a part of the routine clinical detection of breast cancer, for instance at many screening sites and hospitals (Tang, Rangayyan, Xu et al., 2009).

While the interpretation of histology images remains the standard for cancer diagnosis, current computer technology towards this task falls behind clinical needs. Manual analysis of histology tissues depends heavily on the expertise and experience of histopathologists. Such manual interpretation is time consuming and difficult to grade in a reproducible manner. Empirically, it is known that there are substantial intra- and inter-observation variations among experts (Meijer, Beliën, Van Diest et al., 1997). Such factors impede the development of effective computer-based histology analysis along other factors: (1) the large diversity and high complexity of histology traits make it difficult to develop a universal computer system to analyze images of different cancers; (2) the fact that advanced image processing systems for radiology and cytology applications cannot be directly adopted for histology images due to the different imaging technologies and image characteristics; and (3) the scarcity of annotation of cancerous tissue identification and classification, which makes algorithm evaluation largely subjective or only dependable to minimal confidence testing. Nonetheless, the growing demands on experts to inspect the images has driven interest in CAD systems.

Systems for CAD may reduce the workload of pathologists. For instance, it can automatically filter out obvious benign regions of the histology slide, so that the pathologist can focus on more difficult regions. Since current diagnosis relies on the subjective opinion of pathologists, it is clear that a quantitative image-based assessment of digital pathology slides is important from a diagnostic perspective, as well as a way to understand the underlying reasons for a specific diagnosis.

Automation of CAD systems can be traced back to the analysis of the spatial structure of histology images (Bartels, Thompson, Bibbo et al., 1992; Hamilton, Anderson, Bartels et al., 1994; Weind, Maier, Rutt et al., 1998). Techniques in image processing and machine learning (ML) have been leveraged to identify discriminative structures and classify histology images (He *et al.*, 2012). These techniques range from thresholding (Gurcan, Pan, Shimada et al., 2006; Petushi, Garcia, Haber et al., 2006), to active contours (Bamford & Lovell, 2001), Bayesian classifiers (Naik, Doyle, Madabhushi et al., 2007), graphs that model the spatial structure (Bilgin, Demir, Nagi et al., 2007; Tabesh, Teverovskiy, Pang et al., 2007), and ensemble methods based on SVMs and Adaboost (Doyle, Rodriguez, Madabhushi et al., 2006b; Qureshi, Sertel, Rajpoot et al., 2008).

An overview of techniques and their applications is provided in (Gurcan *et al.*, 2009; He *et al.*, 2012; Veta *et al.*, 2014).

Deep Learning (DL) models (Goodfellow, Bengio & Courville, 2016), and in particular Convolutional Neural Networks (CNNs), provide state-of-the-art performance in many visual recognition applications such as image classification (Krizhevsky, Sutskever & Hinton, 2012), object detection (Redmon, Divvala, Girshick et al., 2016) and segmentation (Dolz, Desrosiers & Ben Ayed, 2018). These supervised learning architectures are trained end-to-end with large amount of annotated (labeled) training data to encode a hierarchy of discriminant image features representing different levels of abstraction. More recently, the potential of DL models has begun to be explored in assisted pathology diagnosis (Daisuke & Shumpei, 2018; Janowczyk & Madabhushi, 2016; Li & Ping, 2018). Given the growing availability of histology slides, DL models for CAD have not only been proposed for disease prediction (Hou, Samaras, Kurc et al., 2016; Li & Ping, 2018; Sheikhzadeh, Guillaud & Ward, 2016; Spanhol *et al.*, 2016a; Xu, Luo, Wang et al., 2016), but also for related tasks like detection and segmentation of tumor regions within WSI (Kieffer, Babaie, Kalra et al., 2017; Mungle, Tewary, Das et al., 2017), scoring of immunostaining (Sheikhzadeh *et al.*, 2016; Wang, Foran, Ren et al., 2015), cancer staging (Shah, Wang, Rubadue et al., 2017; Spanhol *et al.*, 2016a), mitosis detection (Chen, Qi, Yu et al., 2016; Cireşan, Giusti, Gambardella et al., 2013; Roux, Racoceanu, Loménie et al., 2013), gland segmentation (Caie, Turnbull, Farrington et al., 2014; Gertych, Ing, Ma et al., 2015; Sirinukunwattana *et al.*, 2017), and detection and quantification of vascular invasion (Caicedo, González & Romero, 2011).

Histology images differ from natural images because the regions of interest do not have a common structure and they are not salient. ML techniques proposed to analyze histology images often require full supervision to address key tasks, such as classification, localization, and segmentation (Daisuke & Shumpei, 2018; Janowczyk & Madabhushi, 2016). Normally, learning to accurately localize regions of interest requires dense pixel-level annotations of images. In order to train a CNN for, *e.g.* pixel-wise localization of cancerous regions, one typically requires a large number of histology images with pixel-level labels to optimize the parameters of the model. Considering the size and complexity of such images, dense annotations of images come at a considerable cost and require highly trained experts. Outsourcing this task to standard workers such as Mechanical Turk Worker is not an option. As a result, histology datasets are often comprised of large images that are coarsely-annotated according to the diagnosis. Therefore, it is clear that training powerful DL models to predict the image class and the regions linked to this class *without* dense annotations is highly beneficial in histology image analysis.

In this paper, we focus on DL models that can be trained using data with image-level labels in order to classify a histology image, while yielding pixel-wise scores, thereby localizing the corresponding regions of interest within the image. Techniques for WSL are very promising for this purpose because they exploit unlabeled inputs, and coarse and ambiguous labels. They are applied in scenarios involving either (1) incomplete supervision (when only a small subset of training data has labels, although unlabeled data is abundant), (2) inexact supervision (when training with labeled data with coarse labels), and (3) ambiguous or inaccurate supervision (when labels may suffer from errors or noise) (Zhou, 2017). The inexact supervision scenario is relevant in this paper, where training datasets only require global image-level annotations. Under this scenario, powerful techniques for multiple-instance learning (MIL) (Carbonneau, Cheplygina, Granger & Gagnon, 2018; Cheplygina, de Bruijne & Pluim, 2019; Wang, Yan, Tang, Bai & Liu, 2018; Zhou, 2004) are generally considered, where individual instance labels (*e.g.* image pixels, segments or patches) are not observable or do not belong to well-defined classes – training instances are grouped into sets (*e.g.* images), and supervision is only provided for sets of instances.

While there has been different reviews of machine/deep learning models for medical image analysis, and in particular for histology slides analysis (Daisuke & Shumpei, 2018; Janowczyk & Madabhushi, 2016; Kandemir & Hamprecht, 2015; Litjens, Kooi, Bejnordi et al., 2017; Sudharshan *et al.*, 2019) and medical video analysis (Quellec, Cazuguel, Cochener & Lamard, 2017), they are focused on fully supervised or semi-supervised learning scenarios (Litjens *et al.*, 2017) for classification and segmentation. To our knowledge, this paper presents the first survey of deep weakly supervised learning (WSL) models for classification and pixel-wise localization of regions of interest in histology images. Most of the DL models in this survey rely on an MIL framework either explicitly, by using its formulation, or implicitly, by splitting the entire image into instances for learning.

Deep WSL techniques (Cheplygina *et al.*, 2019; Zhou, 2017) also provide the advantage of *interpretability* (Zhang & Zhu, 2018). Despite the success of deep neural networks in many different applications, they are often seen as black boxes that lack the ability to provide explanatory factors of their decisions (Lipton, 2018; Marcus, 2001,1; Ribeiro, Singh & Guestrin, 2016; Samek, Wiegand & Müller, 2017). The transparency issue (*i.e.* the absence of clear explanatory factors of a model's decision) is a potential liability for ML models applied in medical image analysis (O'Neil, 2016). *Interpretable ML* (Doshi-Velez & Kim, 2017; Molnar et al., 2018) is an emerging branch of ML that aims to promote the design of interpretable ML models, and provide new techniques to explain a model's decisions. In computer vision, visual attention maps represent one of such technique developed for *pixel-wise localization* of regions within the image used by the network to make its prediction (Zhang & Zhu, 2018). The deep WSL models investigated in this paper produce an attention map where high magnitude responses correspond to image regions of interest. One can thereby extract region locations without the need for pixel-level annotation (Zhou, Khosla, Lapedriza et al., 2016). From medical perspective, pixel-wise region localization can provide a more accurate and visual explanatory factor for the model's prediction of a cancer type, which is a highly desired property in a CAD system. For instance, regions of interest can later be discarded by the pathologist if they are predicted as benign, or be further inspected if they indicate cancerous regions.

This paper provides a survey of state-of-the-art deep WSL models that are suitable for the identification of diseases (*e.g.* type of cancer) in whole slide histology images, and pixel-wise localization of regions of interest that correspond to the predicted disease. Given a dataset of globally-annotated training images, these models allow to simultaneously classify images while localizing the corresponding regions of interest. Two types of WSL approaches have been proposed in the literature that build attention maps to localize regions – (1) bottom-up approaches – like WILDCAT (Durand, Mordan, Thome et al., 2017) and deep MIL (Ilse, Tomczak & Welling, 2018) – that use forward-pass information, either by spatial pooling of representations/scores, or by detecting class regions), and (2) top-down approaches – like Grad-CAM (Chattopadhyay, Sarkar, Howlader et al., 2018; Selvaraju, Cogswell, Das et al., 2017) – that use backward information, and are inspired by human visual attention. Most of these methods do not rely on any prior knowledge on the nature of images at hand.

The most relevant models are compared experimentally in terms of accuracy for image classification, pixel-wise localization of regions, and complexity on several public benchmark histology datasets for breast and colon cancer. Unfortunately, histology datasets with both image- and pixel-level labels are rare, and several benchmarks are private. In order to provide more histology image benchmarks for large scale evaluations of WSL methods, we also propose a protocol to build WSL datasets from WSIs. Our deterministic code and the coordinates of sampled patches from the CAMELYON16 dataset are publicly available. Models from literature have mainly been developed in computer vision community, and validated on natural scene images. Consequently, our experiments allow investigating the extent to which these methods can be applied to histology images which have different properties and challenges, including large size, non-salient and highly unstructured regions, stain heterogeneity, and coarse/ambiguous labels.

This survey is organized as follows. Section 1.2 provides some background on histology image production and analysis as well as key challenges. In Section 1.3, different models for deep weakly supervised localization are described and analyzed with histology image analysis in mind. Finally, Section 2.1 presents the experimental methodology for our comparative study (datasets, protocols and performance metrics), while Section 2.2 presents quantitative and qualitative results, interpretation, and future research directions.

1.2 Histology image analysis – background and challenges

Cytology imagery provides interesting characteristics that ease the visual analysis like isolated/clustered cells and the absence of complicated structures such as glands. Moreover, this type of image often results from the least invasive biopsies, contributing to their common use in disease screening and biopsy purposes (Gurcan *et al.*, 2009). Compared to cytology

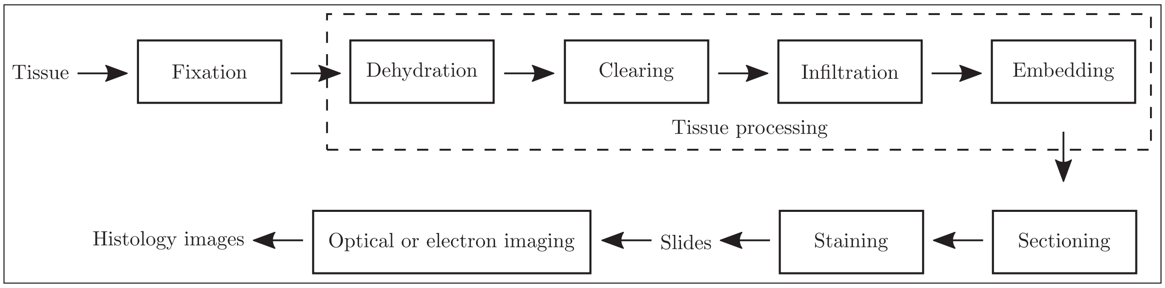


Figure 1.2 Histology tissue preparation and image production (He *et al.*, 2010,1)

imagery, histology slides provide a more comprehensive view of diseases and their effects on tissue (Hipp *et al.*, 2011) since their preparation preserves the underlying tissue structure (He *et al.*, 2012). Histology analysis is performed by inspecting a thin slice (*i.e.* section) of tissue under an optical or electron microscope (Gartner & Hiatt, 2006; Kiernan, 1990; Mescher, 2013; Murphy & Davidson, 2001; Sternberg, 1997). The study of histology images is considered as the gold standard for clinical diagnosis of cancer and identification of prognostic and therapeutic targets. Histopathology, the microscopic study of biopsies to locate and classify diseases, has roots in both clinical medicine and basic science (Sternberg, 1997). In this section, we first summarize the production of histology images, from tissue preparation to imaging technologies. Then, we briefly review histology image analysis, its relation to other types of medical imaging, and its main challenges.

1.2.1 Image production

Figure 1.2 presents an overview of the process of obtaining histology images. Fixation is the first stage of preparation for subsequent procedures, which should be conducted in real time to preserve the samples as well as possible. Different fixatives (*e.g.* precipitant and crosslinking) or methods (*e.g.* heat fixation and immersion) may be used. For example, the precipitant fixatives (*e.g.* methanol, ethanol, acetone, and chloroform) dehydrate the tissue samples, removing lipids and reducing the solubility of proteins. After fixation, the tissue must be adequately supported, *e.g.* frozen or embedded in a solid mold, to allow sufficiently thin sections to be cut for microscopic examination. Common treatments employ a series of reagents to process the fixed tissue and embed it in a stable medium such as paraffin wax, plastic, or resin. Such treatments include the main steps of dehydration[1], clearing, infiltration, and embedding (Chandler & Roberson, 2009; Nelson, Lehninger & Cox, 2008; Wootton, Springall, Polak et al., 1995).

The embedded tissue sample is finally cut into thin sections (*e.g.* $5\mu m$ for light microscopy and $80 - 100nm$ for electron microscopy). The transparent sections are usually produced with a microtome, an apparatus feeding the hardened blocks through a blade with high precision. After cutting, the sections are floated in warm water to smooth out any wrinkles. Then, they are mounted (by heating or adhesives) on a glass slide. Once they are attached on the slide, the process is reversed prior to staining. The wax is removed with a solvent (usually xylenes) and the tissue is re-hydrated through a series of solutions in which the alcohol - water ratio is changed. The gradual rehydration preserves tissue architecture. Now, the sections are ready for staining, which helps to enhance the contrast and highlight specific intra- or extra-cellular structures. A variety of dyes and associated staining protocols are used. The routine stain for light microscopy is hematoxylin and eosin (H&E); other stains are referred to as special stains for specific diagnostic needs. Each dye binds to particular cellular structures, and the color response to a given stain can vary across tissue structures. For example, hematoxylin stains the nuclear components of cells dark blue and eosin stains the cytoplasmic organelles varying shades of pink, red, or orange. (Kiernan, 1990; Ross, Kaye & Pawlina, 2003) provide a detailed description of common laboratory stains. After staining, the stained section on the slide is covered to protect the tissue and provide better visual quality for microscope examination.

After the tissue has been prepared, light microscope (Murphy & Davidson, 2001; Török & Kao, 2007) is used to take digital histology images of the stained sections. Additional details on different types of microscopes and image production are provided in (He *et al.*, 2010,1).

[1] The purpose of dehydration is to remove water so that the paraffin wax can infiltrate.

1.2.2 Image analysis

In histology image analysis for cancer diagnosis, histopathologists visually inspect the regularities of cell shapes and tissue distributions. Such histopathological study has been extensively employed for cancer detection and grading applications, including prostate (Doyle, Madabhushi, Feldman et al., 2006a; Doyle, Hwang, Shah et al., 2007), breast (Basavanhally, Agner, Alexe et al., 2008; Doyle, Agner, Madabhushi et al., 2008), cervix (Guillaud, Cox, Malpica et al., 2004; Guillaud, Adler-Storthz, Malpica et al., 2005), and lung (Jütting, Gais, Rodenacker et al., 1999; Kayser, Riede, Werner et al., 2002) cancer grading, neuroblastoma categorization (Gurcan *et al.*, 2006; Kong, Shimada, Boyer et al., 2007b), and follicular lymphoma grading (Cooper, Sertel, Kong et al., 2009; Kong, Sertel, Lozanski et al., 2007a).

Histopathology has attracted researchers from different disciplines including clinical medicine, biology, chemistry and machine learning. Computer-based image analysis has become an increasingly important field due to the high rate of production and the increasing reliance on these images by the biomedical community. Medical image processing and analysis in radiology (*e.g.* X-ray, ultrasound, CT, MRI) and cytology have been active research fields for several decades with numerous systems (Bankman, 2008; Greenberg, 1984; He, 2009; Yoo, 2004) and products[2][3][4] (Lamprecht, Sabatini & Carpenter, 2007; Schroeder, Ng & Cates, 2003) developed. However, the application of these systems in histology analysis is not straightforward due to the significant difference in the imaging techniques and image characteristics.

The complexity of histology images is defined by several factors including overlapping tissue types and cell boundaries and nuclei corrupted by noise; some structures, such as cell boundaries, may appear connected or blurred. These factors make it difficult to extract cell regions (*e.g.* nuclei and cytoplasm) by traditional image segmentation approaches. On the other hand, cytology images are taken at higher magnification level which results in clearly identified cell

[2] ImageJ (https://rsbweb.nih.gov/ij).

[3] Medical Image Processing, Analysis and Visualization (https://mipav.cit.nih.gov) (MIPAV).

[4] CellProfiler: Cell Image Analysis Software (https://www.cellprofiler.org).

compartments. Computer-based histology analysis systems generally exploit a much larger quantity of image features to derive clinically meaningful information than similar systems for radiology and cytology (He *et al.*, 2012). Nevertheless, the image analysis systems for these three domains generally consist of a common sequence of steps of image restoration, segmentation, feature extraction, and pattern classification.

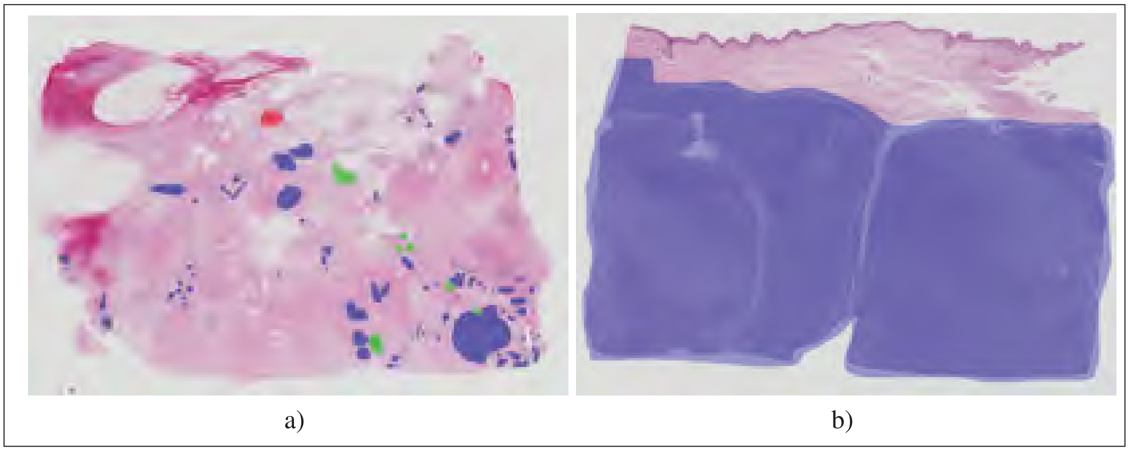


Figure 1.3 Segmentation of two WSI from the ICIAR 2018 BACH Challenge. Colors represent cancerous regions of different types: red for `Benign`, green for `In Situ Carcinoma` and blue for `Invasive Carcinoma`. We can see an important difference in the size and presence of regions (Aresta *et al.*, 2018).

1.2.3 Key challenges

Recently, histology image analysis has attracted much attention in the ML and computer vision communities (Daisuke & Shumpei, 2018; Litjens *et al.*, 2017; Spanhol *et al.*, 2016a; Sudharshan *et al.*, 2019) resulting in open competitions and public datasets such as GlaS (Sirinukunwattana *et al.*, 2017), TUPAC16 (Veta, Heng, Stathonikos et al., 2018), CAMELYON (Bándi, Geessink, Manson et al., 2019; Ehteshami Bejnordi *et al.*, 2017) and BACH 2018 (Aresta *et al.*, 2018). The following paragraphs describe the main difficulties of designing ML models for visual recognition using this type of images.

High resolution images

Pathology images come often in high resolution (WSI, Figure 1.3), leading to difficulties in terms of memory storage and processing time. A WSI has a higher resolution than the most common medical imaging types. For instance, the largest radiological image datasets obtained on a routine basis are high resolution chest CT scans comprising approximately $(512, 512, 512)$ spatial elements ($\sim$ 134 million voxels). In contrast, a single core of prostate biopsy tissue digitized at 40× magnification is approximately $(15,000, 15,000)$ elements ($\sim$ 225 million pixels). To put this in context, a single prostate biopsy procedure can contain anywhere between 12 and 20 biopsy samples or approximately 2.5–4 billion pixels of data generated per patient study (Gurcan *et al.*, 2009; Hipp *et al.*, 2011). In practice, this issue is addressed either by downsampling to lower resolution WSI, which results in a significant loss of image details, or by preserving such details and losing the spatial information of the entire WSI by sampling patches from the WSI. A potential issue in sampling patches is that the WSI labels are not transferred correctly to the patch. A sampled patch from a WSI with a cancerous label may contain only healthy tissue, however, it will be assigned the class of the WSI. This inconsistency in patches labeling can mislead ML models during the training process and decrease the model's performance (Frenay & Verleysen, 2014; Sukhbaatar, Bruna, Paluri et al., 2014; Zhang, Bengio, Hardt et al., 2017). Moreover, the high resolution of WSI makes pixel-level annotation impractical and extremely time consuming. In practice, the WSI annotation is coarse and scarce (*i.e*. the overall diagnosis) (Fig.1.3). This prevents from obtaining large corpora to accurately train ML models for pixel-wise localization and segmentation of images.

Heterogeneous data

Another key challenge in histology image analysis is related to the heterogeneity of data due to variations in staining. As described in Subsection 1.2.1, histology images are produced after many processing steps. Since they involve different chemical processes, many variables may affect the resulting histology image stain including the target diagnosis, the operator, the laboratory, the type of used chemicals, the duration of exposure to them, the microscope, and many other factors.

Figure 1.4 shows an example of such stain variation. While it is easy for pathologists to discard these variations, ML models can be heavily and negatively affected since they are sensitive to changes in the statistics of input signals (Shimodaira, 2000; Sugiyama & Kawanabe, 2012), in particular neural based models (Szegedy, Zaremba, Sutskever et al., 2013). In practice, this issue can be alleviated either by performing color normalization (Ciompi, Geessink, Bejnordi et al., 2017; Janowczyk, Basavanhally & Madabhushi, 2017) or color augmentation during training to improve robustness to stain variations. Among these strategies, color augmentation[5] is particularly relevant when training using small datasets.

Noisy annotations

Noisy or ambiguous annotations are a common practical issues in ML. In histology image analysis, this issue arises as a result of the way the pathologists grade WSIs. Often, such annotation is conducted by assigning the worst stage of cancer to the image. Therefore, a WSI that is labeled with a specific grade is more likely to contain most of the grades that are lower than the labeled grade. During the training of ML algorithms, sampling patches is a common strategy used to deal with large images. In this case, the WSI label is transferred to each patch. Such label transfer is not reliable and introduces noise and inconsistency in the patch label. Label inconsistency can degrade model performance and entangle learning (Frenay & Verleysen, 2014; Sukhbaatar *et al.*, 2014; Zhang *et al.*, 2017). Most of the time, a cancerous patch contains a relatively small cancerous region, while the rest is normal. The issue is aggravated when having many classes to characterize non-cancerous and cancerous lesions (*e.g.* `benign`, `in situ`, `invasive` along with the `normal` class).

1.3 A survey of deep weakly supervised learning techniques for classification and localization

This section presents a review of state-of-the-art deep WSL models that can be trained to *simultaneously* perform two tasks – image classification and pixel-wise localization – using only

[5] *E.g.* randomly modifying brightness, contrast, saturation and hue within chosen ranges

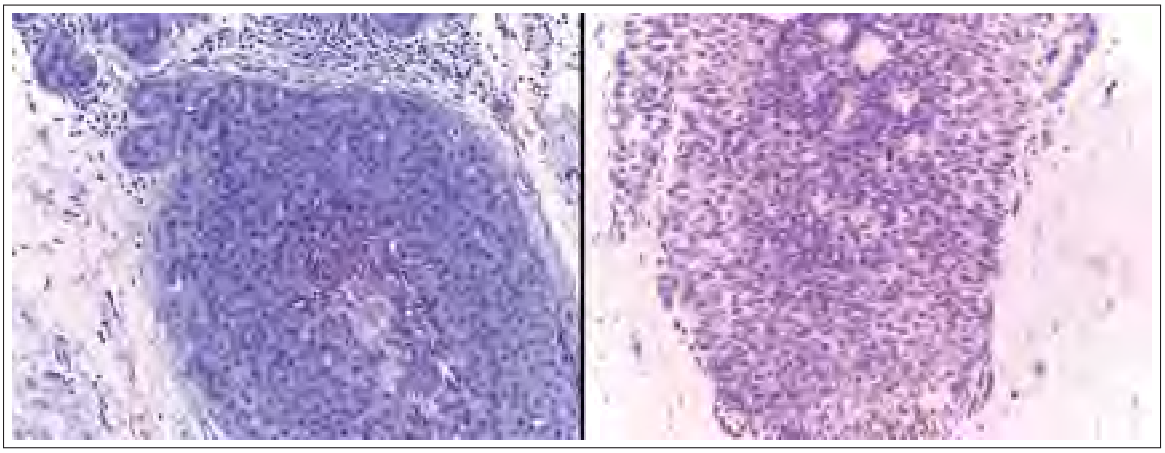


Figure 1.4 Difference in staining for two images labeled both as *In Situ Carcinoma* extracted from different WSI (Aresta *et al.*, 2018)

WSIs annotated with global labels. Most of these techniques have been originally proposed to process natural scene images and validated on well-known public benchmarks such as ImageNet (Deng, Dong, Socher et al., 2009), Pascal VOC (Everingham, Van Gool, Williams et al., 2010a) and MS-COCO (Lin, Maire, Belongie et al., 2014). Since histology images have different characteristics from natural scene images, we first present the main categories of models for deep WSL in natural scene images and then describe the models that are most relevant for our application. We end this section with a critical analysis and a selection of relevant models for experimental evaluation on histology datasets (Subsection 1.3.4).

1.3.1 Overall taxonomy

Among deep weakly supervised localization methods, we identify two main categories based on the way region localization is achieved (Figure 1.5): bottom-up methods that are based on the forward pass information within a network, and top-down methods that are based on the backward information. Figure 1.6 illustrates the overall taxonomy. All these methods employ **a localization mechanism** in order to isolate regions of interest. They rely on either: (1) an attention map where high magnitude responses correspond to salient regions within the image – *i.e*. regions of interest – or (2) a bounding box that encloses the region of interest. These

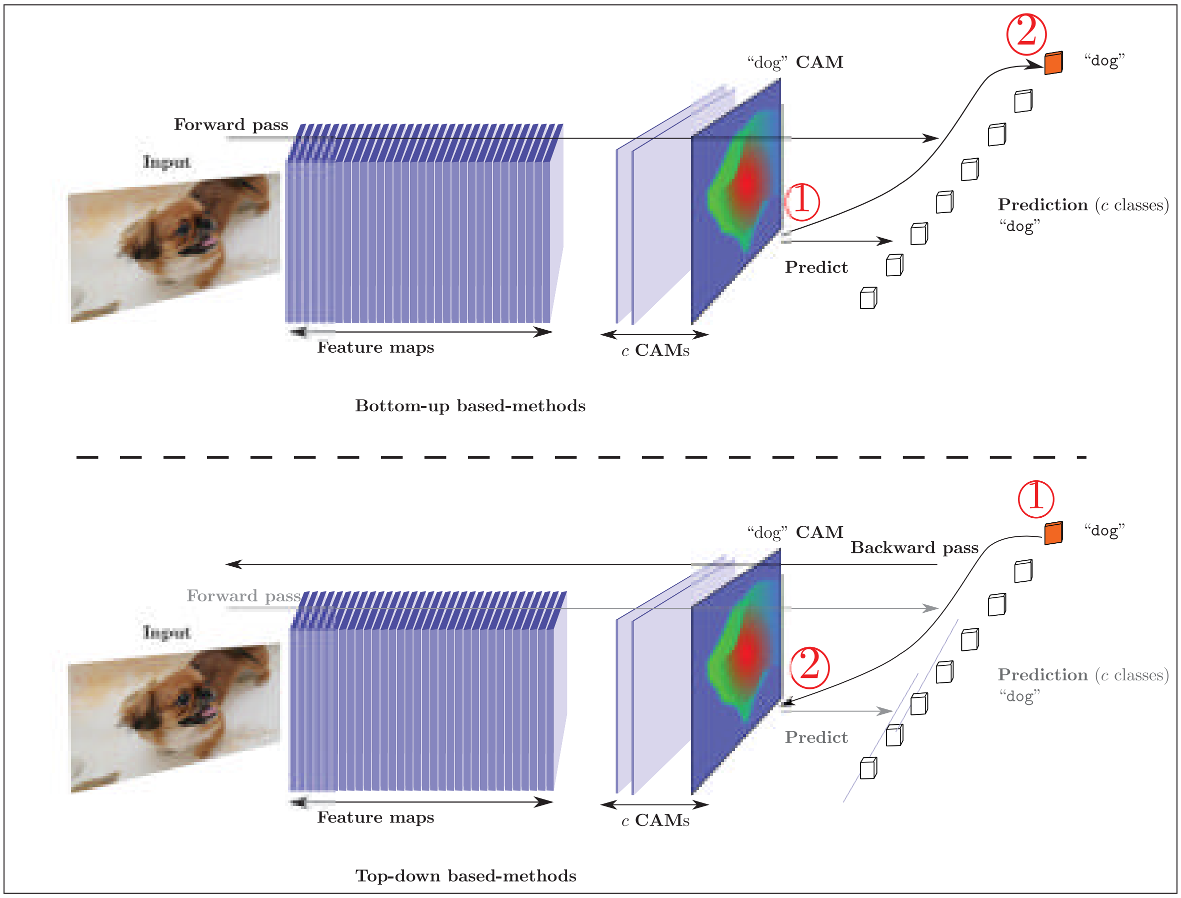


Figure 1.5 Illustration of the main difference between bottom-up (*top*) and top-down (*bottom*) WSL techniques. Both approaches provide CAMs, however, bottom-up techniques produces them during the forward pass, while top-down techniques require a forward, then a backward pass to obtain them.

methods also require **image-level annotation** in order to train DL model to classify an image while localizing the corresponding regions of interest within the image.

1.3.1.1 Bottom-up weakly supervised localization techniques

With these methods, the pixel-wise localization is based on the activation of the feature maps resulting from the standard flow of information within a network from the input signal into the output target (forward pass, Figure 1.5 (*top*)). Within this category, we identify two different subcategories of techniques to address weakly supervised localization. The first category

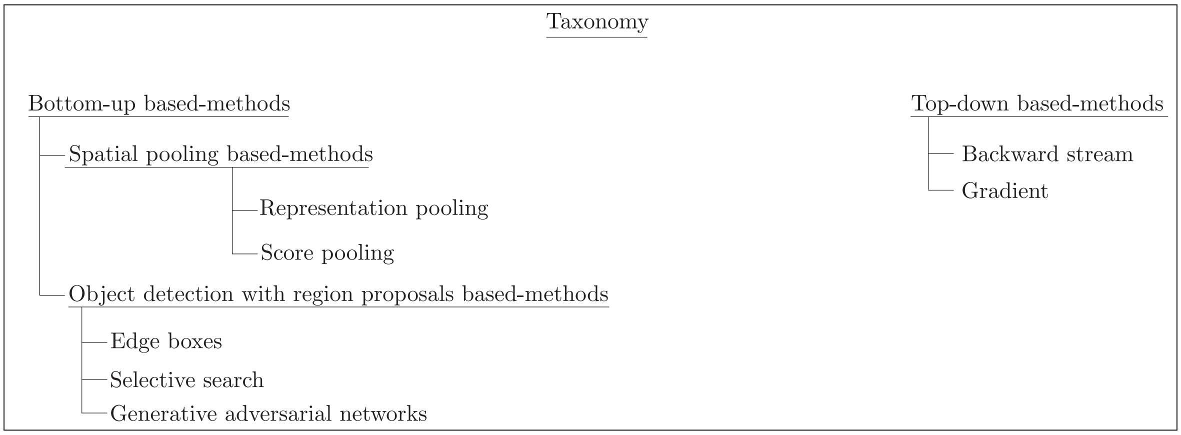


Figure 1.6 Overall taxonomy of deep weakly supervised localization models that rely on global image annotations

contains techniques that are based on spatial pooling of either representations or scores which aim at classifying a bag of instances while obtaining localization throughout the activation of the spatial maps (*i.e*. classifying instances). The second category contains techniques related to object detection which essentially aim to localize regions associated with classes.

Methods based on spatial pooling. This category of techniques are mainly based on learning a spatial representation that promotes the localization of the regions of interest, which is later pooled to classify the input. Within this category, we distinguish two main strategies.

- The first approach aims at building a global representation of the input and then classify it. This corresponds to the approach initially proposed by (Zhou *et al.*, 2016) in which the global representation is obtained by averaging the local representations. The class-specific activations are then obtained by a linear combination of the features using the weights of the classification layer. This strategy has been widely used for natural scene images as well as for medical images (Feng, Yang, Laine et al., 2017; Gondal, Köhler, Grzeszick et al., 2017; Izadyyazdanabadi, Belykh, Cavallo et al., 2018; Sedai, Mahapatra, Ge et al., 2018) where it often combines features from multiple levels (corresponding to different scales) to improve the performance. A more recent strategy proposed by (Ilse *et al.*, 2018) builds a representation as a weighted sum of the local representation where the weights are attention scores produced by a scoring function.

- The second approach aims at obtaining a global score for each class based on the local scores. The classification is done at the instance level and the scores are pooled using different strategies. The first approach proposed by (Oquab *et al.*, 2015) uses max pooling to obtain a score for the image, while the final score for a class is the maximum score of all the instances. However, this pooling technique tends to focus on small discriminative parts of objects (Zhou *et al.*, 2016). To alleviate this problem, Pinheiro & Collobert (2015b) propose to use a smoothed approximation of the max function to discover larger parts of the objects of interest. Finally, (Durand, Thome & Cord, 2016; Durand *et al.*, 2017) propose to use negative evidence to obtain a global score: instead of using only the maximum scoring instances, the pooling is based on both the maximum and minimum scoring instances which provides a strong regularization during training. This method has also been used in the medical field for histology image classification (Couture, Marron, Perou et al., 2018) and weakly supervised region localization and image classification in the same type of images (Courtiol, Tramel, Sanselme et al., 2018).

Methods based on object detection The second type of techniques within this category are related to Weakly Supervised Object Detectors (WSOD). The main goal of WSOD is to produce a region (or a set of regions) that are characteristic of one class (or a set of classes, not necessarily different). These regions are defined by rectangular bounding boxes and try to fit the object as much as possible (*i.e*. the bounding box is in contact with the outer edges of the object). The main difficulty of WSOD is to obtain an accurate placement of the bounding boxes. Most approaches rely on Region Proposal (RP) mechanisms such as Edge Boxes (EB) (Zitnick & Dollár, 2014) or Selective Search (SS) (Uijlings, Van De Sande, Gevers et al., 2013; Van de Sande, Uijlings, Gevers et al., 2011). The RP mechanism is used to generate a list of candidate regions that are likely to contain an object of interest. It can be introduced at different levels of the architecture and it was shown to heavily impact the performance of the overall algorithm. An early approach using this mechanism is used in (Teh, Rochan & Wang, 2016) where the content of each region is passed through attention and then scoring modules to obtain an average image feature which is a weighted average of the proposals. Bilen & Vedaldi (2016) propose a WSOD framework to address the task of multi-class object detection. More improvements of this work have been proposed since then (Kantorov, Oquab, Cho et al., 2016; Tang, Wang, Bai et al., 2017). Other approaches use multi-step training to first train a CNN for localization and then refine it for object detection (Diba, Sharma, Pazandeh et al., 2017; Ge, Yang & Yu, 2018; Sun, Paluri, Collobert et al., 2016). Wan, Wei, Jiao et al. (2018) propose to train a network to reduce its variance in terms of the proposals by reducing an entropy defined over the position of the proposals. Shen, Ji, Zhang et al. (2018) propose to use generative adversarial networks to generate the proposals.

1.3.1.2 Top-down weakly supervised localization techniques

This second main category is based essentially on the backward pass information within a network to build an attention map in order to localize regions with respect to a selected class. The main idea in this category is based on an optimization algorithm that aims at maximizing the posterior response of the network given an output target (*i.e.* class). This optimization scheme allows building an activation map where neurons that support the output target are activated. Different approaches have been used to build these activation maps including a probabilistic Winner-Take-All process that combines bottom-up and top-down information to compute the winning probability of each neuron (Zhang, Bargal, Lin et al., 2018a), a backward layer (Cao, Liu, Yang et al., 2015), or by computing the gradient of the output target with respect to the feature maps (Chattopadhyay *et al.*, 2018; Selvaraju *et al.*, 2017). In practice, these approaches are known to be computationally expensive.

1.3.2 Description of bottom-up techniques

Let us consider a set of training samples $\mathcal{D} = \{(x^{(t)}, y^{(t)})\}$ of images $x^{(t)} \in \mathbb{R}^{D \times H^{\text{in}} \times W^{\text{in}}}$ with H^{in}, W^{in} and D being the height, width and depth of the input image respectively; and its image-level label (*i.e.* class) is $y^{(t)} \in \mathcal{Y}$ with C possible classes. For simplicity, we refer to the input and its label as (x, y).

The training procedure aims at learning a neural network that models the function $f_{\boldsymbol{\theta}} : \mathbb{R}^{D \times H^{\text{in}} \times W^{\text{in}}} \rightarrow \mathcal{Y}$ where the input x has an arbitrary height and width and θ is the network parameters. Typically, in a multi-class scenario, given an input, the network outputs a vector of scores $\boldsymbol{s} \in \mathbb{R}^C$ which is then normalized to obtain a posterior probability using a softmax function,

$$\Pr(y = i|x) = \text{softmax}(\boldsymbol{s})_i = \frac{\exp(\boldsymbol{s}_i)}{\sum_{j=1}^{C} \exp(\boldsymbol{s}_j)} \ . \tag{1.1}$$

The predicted class is the one corresponding to the index of the maximum probability which is equivalent to taking the arg max of the score vector: $\underset{i}{\arg\max} \Pr(y = i|x) = \underset{i}{\arg\max}\, \boldsymbol{s}_i$.

Beside the classification of the input image, we are also interested in the pixel-wise localization of the region of interest within the image. The network can also output either a region of interest r related to the predicted class or a set of regions $\mathbb{R}_{\text{inter}} = \{r_i \mid i = 1, \ldots, t\}$, as well a set of C activation maps of height H and width W to indicate the location of the regions of each class. We note this set as a tensor of shape $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$; where $\mathbf{M}_c$ indicates the c^{th} map. $\mathbf{M}$ is commonly referred to as *Class Activation Maps* (CAM). In most practical cases, the height and the width of the CAM is *smaller* than the height and the width of the input image by a factor S called stride such that $H = H^{\text{in}}/S$ and $W = W^{\text{in}}/S$.

1.3.2.1 Spatial pooling

In this category, the beginning of the pipeline is usually the same for all techniques: a CNN extracts K feature maps $\mathbf{F} \in \mathbb{R}^{K \times H \times W}$, where K is the number of feature maps which is architecture-dependent. The feature maps $\mathbf{F}$ are then used to compute a score per class using a spatial pooling either on the representation or the scores of the instances.

We can distinguish two main approaches to compute the per-class score: spatial representation pooling and spatial score pooling.

Spatial representation pooling. In this first approach, the feature maps produced by the CNN are spatially pooled to form a single representation $\boldsymbol{f} \in \mathbb{R}^K$ of the whole input which is then classified.

Global average pooling (GAP). Lin, Chen & Yan (2013) propose a way of regularizing neural networks by adding GAP layers to avoid the use of fully connected layers that dramatically increase the number of parameters. The GAP module allows to obtain dense features $\boldsymbol{f} \in \mathbb{R}^K$ based on spatial features $\mathsf{F} \in \mathbb{R}^{K \times H \times W}$ by averaging the activations of each map,

$$\boldsymbol{f}_k = \frac{1}{H\,W} \sum_{i=1,j=1}^{H,W} \mathsf{F}_{k,i,j} \ , \tag{1.2}$$

where $\boldsymbol{f}_k$ is the k^{th} feature of the output of the GAP. Zhou *et al.* (2016) show that this pooling strategy can be used to obtain a localization ability in a CNN using only global labels. Typically, in a CNN, the last layer which classifies the representation $\boldsymbol{f}$ is a fully connected layer parametrized by $\boldsymbol{W} \in \mathbb{R}^{C \times K}$ such that $\boldsymbol{s} = \boldsymbol{W} \boldsymbol{f}$ (bias is omitted for simplicity). The CAMs, denoted as $\mathsf{M} \in \mathbb{R}^{C \times H \times W}$ are then obtained using a weighted sum of the spatial feature F,

$$\mathsf{M}_{c,i,j} = \sum_{k=1}^{K} \boldsymbol{W}_{c,k}\, \mathsf{F}_{k,i,j} \ . \tag{1.3}$$

The main advantage of Equation 1.3 is that it does not depend on the size of the input. This technique has been used extensively in medical domain (Feng *et al.*, 2017; Gondal *et al.*, 2017; Izadyyazdanabadi *et al.*, 2018; Sedai *et al.*, 2018), often combined with a multi-level feature maps. Combining feature maps from lower layers within the network can allow to obtain CAM with high resolution and more precision. Zhu, Zhou, Ye et al. (2017) propose soft proposal networks (SPNs) which are based on (Zhou *et al.*, 2016) with an extra module that generates a proposal map which highlights regions of the object in hand. Such proposal map is generated iteratively using random walk over a fully connected directed graph that connects each position within a feature map at a specific convolution layer. (Zhu *et al.*, 2017) can be also categorized as an object detection method with region proposals (subsubsection 1.3.2.2).

(Zhang, Wei, Feng et al., 2018b) propose to build the CAMs M by taking the maximum between two set of CAMs. The first set of CAMs is obtained in the same way as in (Zhou, 2017) and used to mask (or erase) a part of the feature maps based on thresholding. Using these masked features, a second set of CAMs is computed using a different layer. (Zhang *et al.*, 2018b) argue that it makes a CNN discover relevant regions more effectively.

Attention-based deep MIL. Ilse *et al.* (2018) propose to build an image (*i.e.* bag) representation using a weighted average of the instances representations based on an attention mechanism (Bahdanau, Cho & Bengio, 2014). Given a set of features $\mathsf{F} \in \mathbb{R}^{K \times H \times W}$ extracted for an image, the representation $\boldsymbol{f}$ of the image is computed as,

$$\boldsymbol{f} = \sum_{i=1,j=1}^{H,W} \boldsymbol{A}_{i,j} \mathsf{F}_{i,j} \qquad \text{with} \qquad \boldsymbol{A}_{i,j} = \frac{\exp(\psi(\mathsf{F}_{i,j}))}{\sum_{i=1,j=1}^{H,W} \exp(\psi(\mathsf{F}_{i,j}))}\,, \tag{1.4}$$

Where $\mathsf{F}_{i,j}$ is (by an abuse of notation) the feature vector of the location (*i.e.* instance) indexed by i and j. $\psi : \mathbb{R}^K \longrightarrow \mathbb{R}$ is a scoring function. The resulting representation $\boldsymbol{f}$ is then classified by a fully connected layer. Ilse *et al.* (2018) propose two scoring functions,

$$\psi_1(\boldsymbol{f}) = \boldsymbol{w} \tanh(\boldsymbol{V}\boldsymbol{f})\,, \tag{1.5}$$

$$\psi_2(\boldsymbol{f}) = \boldsymbol{w} \left[\tanh(\boldsymbol{V}\boldsymbol{f}) \odot \sigma(\boldsymbol{U}\boldsymbol{f}) \right], \tag{1.6}$$

where $\boldsymbol{w} \in \mathbb{R}^L$ and $(\boldsymbol{V}, \boldsymbol{U}) \in \mathbb{R}^{L \times K}$ are learnable weights. This approach is designed specifically for binary classification and produces a matrix of attention weights $\boldsymbol{A} \in [0, 1]^{H \times W}$ with $\sum \boldsymbol{A} = 1$. This means that for a positive bag, negative instances should have an attention weight close to 0 while positive instances should have a high attention weights. However, it is not possible to determine if an instance is actually predicted as positive or not except by fixing a threshold.

Spatial score pooling. In this second approach, the feature maps are used to produce the CAMs directly by classifying each instance. Then, the global per-class scores are obtained by pooling the instances' scores. The methods mainly differ by their strategy used to pool the instances' scores.

First, it is important to note that GAP (Zhou *et al.*, 2016) is designed to be used when the last classification layer of a model is linear. The consequence is that the pooling can also be done on scores which is equivalent. With GAP, Equation 1.2, and Equation 1.3 allow to compute the per-class scores $\boldsymbol{s}_c$ as,

$$\begin{aligned} \boldsymbol{s}_c &= \sum_{k=1}^{K} \boldsymbol{W}_{c,k} \boldsymbol{f}_k \;, \\ &= \frac{1}{H\,W} \sum_{k=1}^{K} \boldsymbol{W}_{c,k} \sum_{i=1,j=1}^{H,W} \mathbf{F}_{k,i,j} \;, \\ &= \frac{1}{H\,W} \sum_{i=1,j=1}^{H,W} \mathbf{M}_{c,i,j} \;, \end{aligned} \tag{1.7}$$

Equation 1.7 shows that the per-class score is computed by averaging the activations of the corresponding CAM. Instead of averaging the feature maps, taking the maximum value can be considered as well (Oquab *et al.*, 2015). However, such an operation tends to favor small discriminative regions (Zhou *et al.*, 2016). (Pinheiro & Collobert, 2015a; Sun *et al.*, 2016) consider using an approximation to the maximum function (Boyd & Vandenberghe, 2004) as an alternative where the score of each map can be computed as,

$$\boldsymbol{s}_c = \frac{1}{q} \log \Big[\frac{1}{H\,W} \sum_{i=1,j=1}^{H,W} \exp(q\, \mathbf{M}_{c,i,j}) \Big] \;, \tag{1.8}$$

where $q \in \mathbb{R}_+^*$ controls the smoothness of the approximation. A smaller value of q makes the approximation closer to the average function while a larger q makes it close to the maximum function. Thus, small values of q make the network consider large regions while large values consider only small regions.

Negative evidence mixing. Durand *et al.* (2016,1) compute m feature maps per-class using 1×1 convolution. Then, compute their average to obtain one feature map. While standard pooling methods are based on considering the maximum or the average of the activations within a map, Durand *et al.* (2016,1) consider using both maximum and minimum activations. A maximum activation indicates a positive evidence of the presence of the corresponding object. In the other hand, a minimum activation indicates a negative evidence (Durand, Thome & Cord, 2015) of its presence. The benefits of such mixing of both information provides a regularization mechanism that prevents the model from overfitting compared to learning only from maximum activation for instance (Durand *et al.*, 2016,1). Such pooling is computed over each map by considering the sum of the average of n^+ maximum activation and n^- of minimum activation. Specifically, the score for each class is computed as,

$$\boldsymbol{s}_c = \frac{Z_c^+}{n^+} + \alpha \frac{Z_c^-}{n^-} \, , \tag{1.9}$$

where Z_c^+ and Z_c^- correspond to the sum of the n^+ highest and n^- lowest activations of $\mathbf{M}_c$ respectively and α is a hyper-parameter that controls the importance of the minimum scoring regions. Such an operation consists in selecting for each class the n^+ highest activation and the n^- lowest activation within the corresponding map. In medical domain, Courtiol *et al.* (2018) show the benefit of mixing negative evidence.

1.3.2.2 Object detection with region proposals

The second category of techniques relevant for weakly supervised localization is related to object detection. Many techniques have been proposed in order to find the coordinates of a relevant region using only image-level labels. The pipeline of these detectors usually contains three operations presented in Figure 1.7. The order of the operations of this pipeline can be changed to accommodate different strategies of supervision but the principle of the operations remains the same. In the next paragraphs, we describe some of the related work that is based on WSOD.

Attention networks for WSOD. In a similar manner to (Ilse *et al.*, 2018), Teh *et al.* (2016) propose to use a fully connected network to generate attention score for each instance. The initial region proposals are generated using the Edge Boxes (EB) (Zitnick & Dollár, 2014) on the input image. Then for each region, a feature vector representation is extracted using a pre-trained CNN. A fully connected network produces a score for each feature vector, which is normalized using a softmax operation on the proposals. The image representation is the weighted sum of the

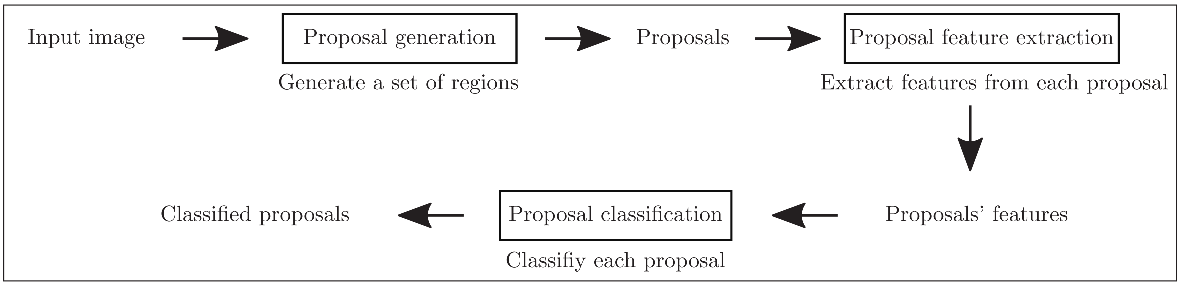


Figure 1.7 Standard pipeline for WSOD methods

feature vectors by the attention weights. The resulting representation is then classified by a fully connected layer. Bency, Kwon, Lee et al. (2016) propose an efficient way to extract top scoring regions from a CNN by performing a tree search on sub-regions of the feature maps. Given the feature maps produced by a CNN, four children regions of small size are extracted and each region is interpolated to produce a feature map of the same size as the parent. These four regions are classified by a fully connected layer. The top-scoring region for the class of the image then becomes the parent region. This process is iterated until it converges to a region with the maximum probability for the class of the image at train time.

Weakly supervised deep detection networks (WSDDN). (Bilen & Vedaldi, 2016) is one of the approaches that has achieved an important improvement compared to previous WSOD techniques (Bilen, Pedersoli & Tuytelaars, 2014,1; Cinbis, Verbeek & Schmid, 2017; Wang, Ren, Huang et al., 2014). Bilen & Vedaldi (2016) propose a modified CNN architecture with two streams: one focusing on recognition and the other one on localization. In this approach, the proposals are generated from the input image using Selective Search (SS) (Uijlings *et al.*, 2013; Van de Sande *et al.*, 2011) or Edge Boxes (EB). In parallel, a CNN produces feature maps for the input image. A Spatial Pyramid Pooling (SPP) (He, Zhang, Ren et al., 2014) is then used to extract the features corresponding to the region proposals from the feature maps. At this point, each region's features are processed by two different fully connected networks to produce classification scores $\boldsymbol{S}^{\text{class}} \in \mathbb{R}^{|\mathbb{R}_{\text{inter}}| \times C}$ and detection scores $\boldsymbol{S}^{\text{det}} \in \mathbb{R}^{|\mathbb{R}_{\text{inter}}| \times C}$. These scores are then normalized to obtain $\boldsymbol{\sigma}^{\text{class}}$ and $\boldsymbol{\sigma}^{\text{class}}$ using a softmax function,

$$\boldsymbol{\sigma}^{\text{class}}_{i,c} = \frac{\exp \boldsymbol{S}^{\text{class}}_{i,c}}{\sum_{j=1}^{C} \exp \boldsymbol{S}^{\text{class}}_{i,j}} \,, \tag{1.10}$$

$$\boldsymbol{\sigma}^{\text{det}}_{i,c} = \frac{\exp \boldsymbol{S}^{\text{det}}_{i,c}}{\sum_{j=1}^{C} \exp \boldsymbol{S}^{\text{det}}_{i,j}} \,. \tag{1.11}$$

The final region-level scores $\boldsymbol{R} \in \mathbb{R}^{|\mathbb{R}_{\text{inter}}| \times C}$ are obtained throughout an element-wise product of the normalized classification and detection scores, and the per-class image-level scores are obtained by summing the region-level score for each class,

$$\boldsymbol{R} = \sigma^{\text{class}} \odot \sigma^{\text{detect}} \,, \tag{1.12}$$

$$\text{and} \quad \boldsymbol{s}_c = \sum_{i=1}^{|\mathbb{R}_{\text{inter}}|} \boldsymbol{R}_{i,c} \,. \tag{1.13}$$

Kantorov *et al.* (2016) further improve this approach by adding more context information where $\boldsymbol{S}^{\text{det}}$ becomes the combination of the score of a region and its surrounding (*i.e.* spatial context). This allows to obtain a better discrimination of the proposed region, eliminating regions that do not tightly fit an object.

Online instance classifier refinement. Tang *et al.* (2017) propose another improvement of the WSDDN by refining the proposals multiple times based on the overlap of the proposals. When a proposal is a top-scoring region for a class that is present in the image, the refinement algorithm will look for all the other proposals that have a high overlap with it and set their label to one for this class. This process forces the model to gradually detect larger parts of objects as the training advances by fusing on high scoring and overlapping regions.

Weakly supervised region proposal network and object detection. Tang, Wang, Wang et al. (2018) suggest that the proposal generation has a great influence on WSOD performance and can benefit from the feature maps produced by a CNN instead of simply using SS or EB on the input image. Once the feature maps are obtained from the CNN, the proposals are generated in three steps. The first proposals are generated using a sliding window on the image. They are refined using the same principle as EB method. To further refine them, a fully connected network re-evaluates the objectness of the refined proposals. Finally, the different proposals are classified by extracting a feature vector representation using the same *region of interest* pooling algorithm as in Fast R-CNN (Girshick, 2015). This method presents the advantage of not relying on SS nor EB to generate the proposals which are known to be computationally expensive.

Region proposal filtering using top-down stream for multi-label recognition. Ge *et al.* (2018) propose to use multiple sources of information to obtain CAMs for better object detection. In this technique, the authors combine object heatmaps with top-down attention maps to obtain more accurate object instances. The object heatmaps are obtained using a pre-trained CNN, by adding the class probabilities of each proposal to its corresponding pixels when the class is present in the image. The attention maps are obtained using excitation backprop (Zhang *et al.*, 2018a) on a second pre-trained CNN. Object heatmaps obtained in a feedforward manner are usually too smooth to give precise information on the object boundaries. Combining them with attention maps allows to better filter the region proposals, and reduce the number of false positive regions.

Deep self-taught learning. Jie, Wei, Jin et al. (2017) suggest that training detectors with image-level supervision leads to poor-quality positive proposals. To start from good proposals, the authors propose to use a graph-based approach to refine the proposals initially generated by EB, by finding the dense sub-graph of proposals based on their spatial overlap. To improve the quality of the proposals, they propose to train a detector by *iteratively* selecting high-quality proposals based on their relative improvement compared to the previous training epoch. With this self-supervision, low-quality proposals obtain low improvement over the epochs, contrary to high-quality ones. This allows the detector to select high-quality regions.

Min-entropy latent model for WSOD. Wan *et al.* (2018) suggest that there is an inconsistency between the weak supervision (absence of labels of object localization) and the learned objectives (*i.e.* asking the model to learn the object location) which introduces randomness and uncertainty to object locations and object detectors. In order to decrease such uncertainty, the authors propose to minimize the entropy of a latent model which aims to reduce the variance of the proposals (*i.e*. the locations of the objects). The proposals are initially generated by SS and a feature representation is extracted for each proposal. They are further refined using a graph-based approach to fuse them when they present the high confidences for a class and a significant spatial overlap. Then, the model is trained to minimize the classification error on the most confident proposals. This produces sparse predictions on the most confident proposals which reduces the randomness of selected proposals during learning.

Generative adversarial networks for WSOD. A main criticism of the WSOD techniques is that they usually follow a multi-step pipeline (Figure 1.7), leading to expensive computational cost, hence, a slow running time. Shen *et al.* (2018) propose to improve the speed by training a WSOD within a generative adversarial framework. Three models are used in this approach: a generator G which is a one-stage detector (Liu, Anguelov, Erhan et al., 2016) that outputs bounding boxes with associated probabilities, a discriminator D predicting the quality of bounding boxes for an image, and a surrogator F which is a modified version of WSDDN used to estimate image proposals. The intuition is that G, which is fast, will learn to generate the same proposals as F, which is slow in comparison, with the supervision of D. D learns to distinguish bounding boxes generated by G from accurate estimated ones generated by F.

1.3.3 Description of top-down techniques

In this second main category, the weakly localization of objects is determined based on information obtained from the stream that goes from the output (top) toward the input (down). We can distinguish two related methods: backward based-methods, and gradient based-methods. We note that such approaches are computationally expensive compared to bottom-up methods (Subsection 1.3.2).

Backward stream based-methods. Excitation backprop for top-down attention (Zhang *et al.*, 2018a) is one of the illustrative examples of such approach. In its formulation, the authors propose a probabilistic winner-take-all formulation in the backward pass of the model to determine which units are active with respect to a selected output class. By defining a prior distribution over the output classes, the winner neurons of lower layers can be sampled recursively in a top-down fashion. Given a neuron a_z at some layer, its probability can be determined using the probability of its parent neurons $a_t \in \mathbb{P}_z$ at the previous layer as follows,

$$\Pr(a_z) = \sum_{a_t \in \mathbb{P}_z} \Pr(a_z | a_t) \Pr(a_t) \, , \tag{1.14}$$

where $\Pr(a_z|a_t)$ is simply a normalized energy that flows from the neuron a_z to a_t with respect to all the neurons that share the same parent as a_z. Using such approach, it is possible to obtain a CAM at each convolutional layer. Since an image may contain many objects, the dominant neurons may belong to different classes. Thus, a CAM may contain the activations of more than one object. To deal with this, Zhang *et al.* (2018a) propose a *contrastive* attention that builds a highly discriminative CAMs by keeping only one class and suppressing the rest. Cao *et al.* (2015) propose a related work where an attention map is built with respect to a selected class using the backward information throughout a *feedback layer*. Neurons in feedback layers are updated iteratively to maximize the confidence of the output target. The selectivity in such neurons is controlled using a binary mask obtained throughout an optimization procedure.

Gradient based-methods. These approaches are based on computing the gradient of any output target with respect to the feature maps to determine the main locations that contribute to the prediction of the selected target. Such approaches are mainly used as *visual tools* to explain a network's decision. (Selvaraju *et al.*, 2017) is an illustrative example of this approach. In order to compute the CAMs, the authors propose to use GAP (Equation 1.3), where the coefficient of each feature map is computed using the gradient of the score of the selected target class with respect to that map. Therefore, a CAM for the class c is a linear combination of the feature maps, similar to Equation 1.3,

$$\mathbf{M}_c = \mathrm{ReLU}\left(\sum_{k=1}^{K} \boldsymbol{A}_{c,k}\ \mathbf{F}_k\right), \tag{1.15}$$

$$\text{where} \quad \boldsymbol{A}_{c,k} = \frac{1}{H\,W} \sum_{i=1,j=1}^{H,W} \frac{\partial \boldsymbol{s}_c}{\partial \mathbf{F}_{k,i,j}}, \tag{1.16}$$

where $\boldsymbol{s}_c$ is the score for class c. This approach is a generalization of the the method proposed by (Zhou *et al.*, 2016) where the derivative of the score with respect to the feature map is used. In the case where the last classification is linear, both formulation are equivalent. This approach has been improved in (Chattopadhyay *et al.*, 2018) to obtain better object localization, as well as explaining occurrences of multiple object instances in a single image.

1.3.4 A critical analysis

Our first observation is that all the deep weakly supervised localization methods have been proposed and validated on natural images. Their application on histology images can be problematic due to the heterogeneous nature of these images. The second observation is that bottom-up techniques have attracted much more attention compared to top-down ones. A possible explanation to this is the simplicity of bottom-up methods which follows classical flow of information within a neural network. In contrast, top-down methods, which are inspired from human visual attention, are more complex in terms of implementation and inference. For these reasons, most of the techniques selected for our experimental evaluation are mainly from the bottom-up family.

Among bottom-up methods, we find weakly supervised localization methods based on a spatial pooling allowing localization of regions after being trained using global labels only. Often, this category of techniques is straightforward to use on histology images and does not rely on prior knowledge on the nature of the image at hand. Among the spatial pooling methods, we evaluate the work in (Zhou *et al.*, 2016) which has shown promising results in terms of classification and weak localization. We consider using three different pooling techniques: average pooling, max pooling, and log-sum-exponential pooling (Equation 1.8), WILDCAT pooling (Durand *et al.*, 2017) which has shown interesting results in terms of classification and pixel-wise localization, and deep multi-instance learning (Deep MIL) (Ilse *et al.*, 2018). All these methods have shown great potential for localization while maintaining high levels of classification accuracy. The major drawback of these methods is that the resolution of the CAMs is small due the stride of the backbone network used to extract features. When pixel-level evaluation is required (typically to evaluate the Dice index), we interpolate the CAMs to match the input size using a bilinear interpolation.

The Deep MIL method proposed by Ilse *et al.* (2018) has two major limitations. Firstly, it is restricted to binary classification. Therefore, for datasets with more than two classes we adapt this method by replicating the pooling and scoring module to match the number of predicted classes. Secondly, this method produces attention scores for each instance which sum to 1. Therefore, when we evaluate at pixel-level, an instance is predicted as belonging to the positive class if its attention weight is superior to $\frac{1}{HW}$. We acknowledge that this is not a perfect criteria as edge cases are not covered: if all instances are predicted with the same score, all attention weights are equal to $\frac{1}{HW}$ which does not indicate whether the initial score was high (positive instance) or low (negative instance). However, we observe that this works well in practice, showing the potential of this method.

In bottom-up category, we also find WSOD techniques based on region proposals, mostly used for object detection. For our experiments, we use the work in (Tang *et al.*, 2018) which shows a large improvement compared to other WSOD methods, and (Bilen & Vedaldi, 2016) which limits the use of SS or EB methods which are known to be computationally expensive. The main limitation presented in WSOD methods is that they are tailored to natural scene images. Therefore, most of them encode in their algorithms some priors on the object in such images. We recall that objects within natural scene images have usually a standard structure/shape, and they tend to be smooth; while a *cancerous region* within histology images, for instance, does not have any prior structure nor appearance. As a consequence of such adaptation for natural scene images, region proposal methods such as SS and EB rely on the fact that edges are likely to delimit an object and pixels inside an object are more likely to be similar. In histology images, the first fact is not clear to be true. Moreover, edges are expected to be very noisy due to high variation in the texture of the microscopic tissue and intensive presence of cell boundaries. For these reasons, we were unable to obtain concluding results with WSOD methods despite our best efforts, and therefore did not include them in our experiments.

For top-down methods, Grad-CAM and Average pooling are equivalent when the last layer (*i.e*. classification layer) of a model is linear. Since we use ResNet models in our study, this is always verified, meaning that both methods are equivalent. Therefore, we only report results for the Average pooling when evluating classification performance. We also initially wanted to evaluate Excitation Backprop (Zhang *et al.*, 2018a) but we were not able to obtain a working code for this method using the PyTorch framework[6].

[6] https://pytorch.org

CHAPTER 2

EXPERIMENTAL COMPARISON OF STATE-OF-THE-ART WEAKLY-SUPERVISED LEARNING METHODS IN HISTOLOGY IMAGES

2.1 Experimental methodology

In this section, we present an experimental evaluation of several deep weakly supervised methods for classification and localization from the previous chapter that are relevant in histology image analysis. The aim of our experiments is to assess the ability of the selected methods to accurately classify histology images, and localize cancer regions of interest. The experiments are conducted on four public datasets of histology images which are described in Subsection 2.1.1. Most of the public datasets were made exclusively for classification or segmentation purposes (Daisuke & Shumpei, 2018). Very few datasets have image-level and pixel-level annotation simultaneously. The only dataset that we found that has both types of annotation is GlaS (subsubsection 2.1.1.3) which is not enough for our evaluation. For this reason, we created a dataset with the required annotations by using a protocol (subsubsection 2.1.2.3) to sample patches from WSIs of the CAMELYON16 dataset (subsubsection 2.1.1.4). Subsection 2.1.3 provides a brief description of the training setup of the relevant techniques that we selected in our comparative study.

2.1.1 Datasets

We describe in this section the four public datasets of histology images that we have used in our experiments. A brief description of the datasets is presented in Table 2.1.

2.1.1.1 BreaKHis dataset

BreaKHis is a publicly available[1] dataset for microscopic biopsy images of benign and malignant breast tumor (Spanhol *et al.*, 2016b). The images were collected through a clinical study from

[1] https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis

Table 2.1 Brief description of the used datasets in our experiments (seg.: segmentation)

Dataset	Medical aspect	Type of image	#Images	Number of classes	Image-level labels	Pixel-level labels
BreakHis	Breast cancer	Patches	7, 909	2 classes: benign, malignant.	Yes	No
BACH (Part A)	Breast cancer	Patches	400	4 classes: normal, benign, in Situ, invasive	Yes	No
GlaS	Colon cancer	Patches	165	2 classes: benign, malignant	Yes	Yes (gland seg.)
CAMELYON16	Cancer metastases in Lymph Nodes	WSIs	399	2 classes: normal, metastases	Yes	Yes (tumor seg.)

January 2014 to December 2014, in which all patients referred to the P&D Laboratory[2] with a clinical indication of breast cancer were invited to participate. The institutional review board approved the study and all patients provided their written consent. All the data were anonymized. Samples were generated from the breast tissue biopsy slides stained with H&E. The samples were collected by surgical open biopsy, prepared for histological study and labeled by pathologists of the P&D Lab. The diagnosis of each case was produced by experienced pathologists and confirmed by complementary exams such as immunohistochemestry analysis.

The original images were acquired in three-channel red-green-blue color space (RGB, 24-bit color depth, 8 bit per channel) with resolution of 752×582 using magnifying factors of 40×, 100×, 200× and 400×. The images were then cropped into size 700×460 and saved in Portable Network Graphics format (PNG) with no compression, nor normalization or color standardization. Figure 2.1 shows these four magnification on a single image. The dataset is composed of 7, 909 images divided into benign and malignant tumors. Table 2.2 summarizes the dataset distribution in terms of number of images per class, magnification factor and patient. The classes benign and malignant are subdivided into different categories. However, in our experiments, we limit ourselves to the two main classes, *i.e.* benign against malignant. For more details on the dataset, we refer to (Spanhol *et al.*, 2016b).

[2] Pathological Anatomy and Cytopathology, Parana, Brazil: http://www.prevencaoediagnose.com.br

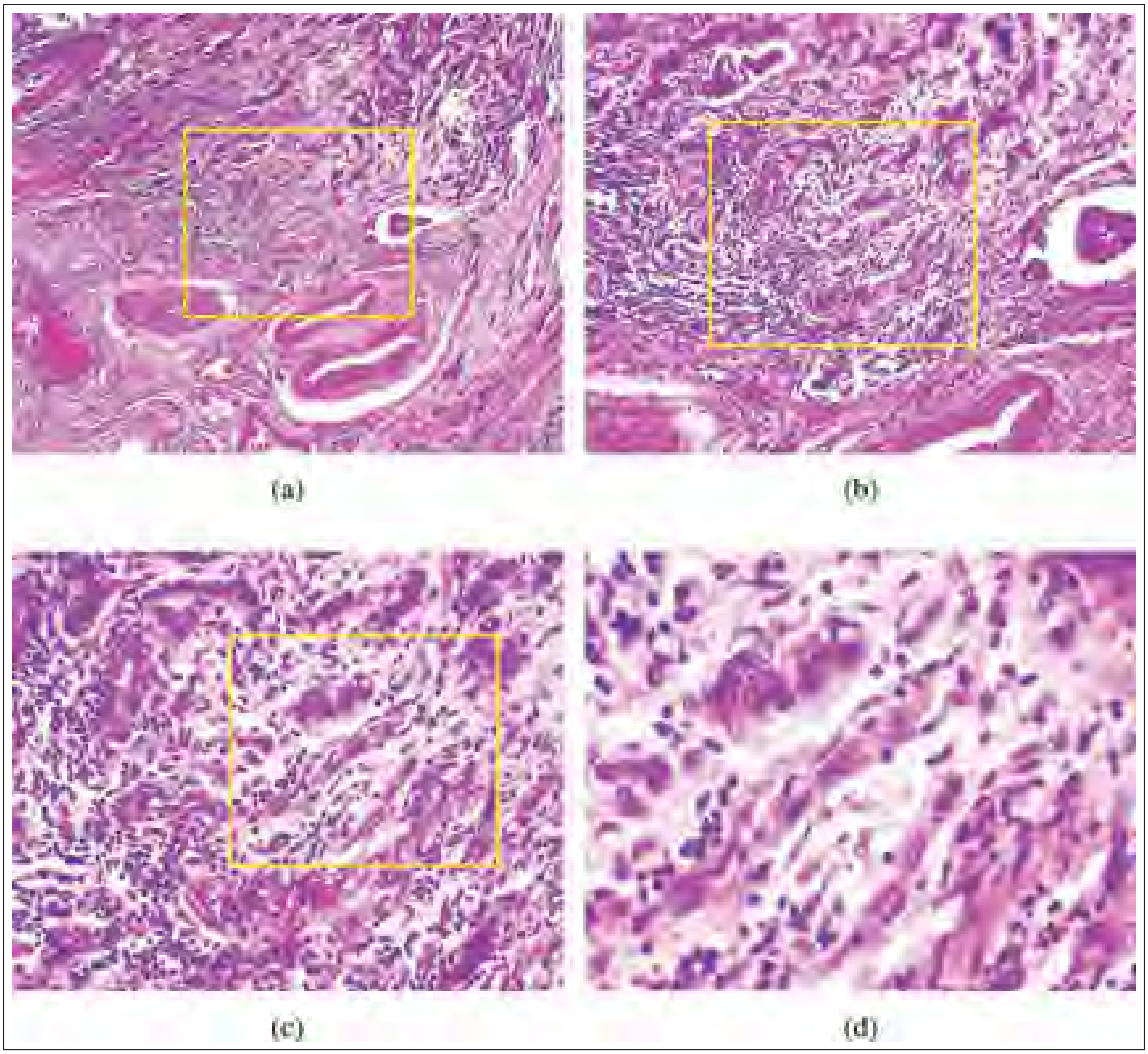


Figure 2.1 Slides of breast malignant tumor (stained with H&E) seen in different magnification factors: (a) 40×, (b) 100×, (c) 200×, and (d) 400×. Highlighted rectangle, which is manually added for illustration purposes only, is the area of interest selected by pathologist to be detailed in the next higher magnification factor (Credit: (Sudharshan *et al.*, 2019)).

2.1.1.2 BACH challenge dataset (Part A) 2018

The Grand Challenge on BreAst Cancer Histology images (BACH)[3] (Aresta *et al.*, 2018), which is a follow-up of the Bioimaging challenge of 2015[4], was organized in 2018 in the aim of

[3] https://iciar2018-challenge.grand-challenge.org/

[4] http://www.bioimaging2015.ineb.up.pt/challenge_overview.html

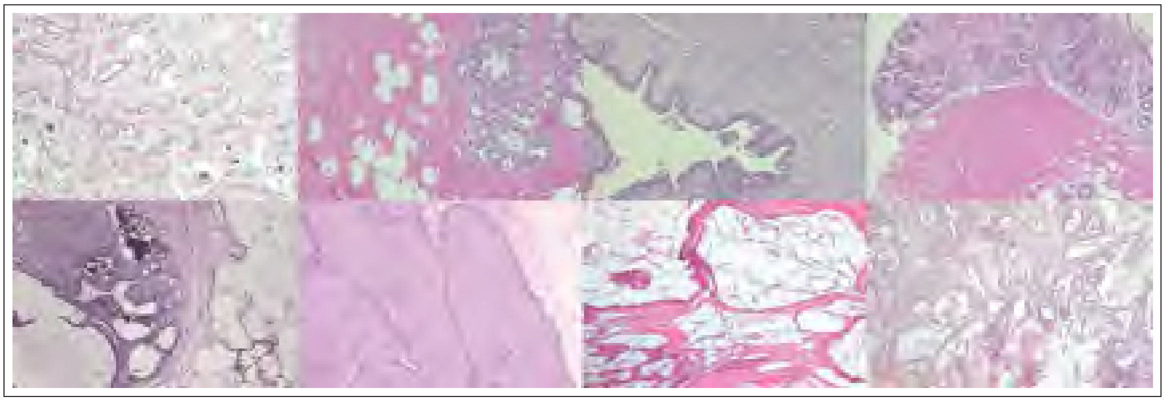


Figure 2.2 Samples from test set of fold 1 (Spanhol *et al.*, 2016a) from BreaKHis dataset. *Row 1*: Benign. *Row 2*: Malignant.

Table 2.2 BreakHis dataset (Spanhol *et al.*, 2016b) distribution by class, magnification factor, and patient

Magnification	Benign	Malignant	Total
40×	625	1,370	1,995
100×	644	1,437	2,081
200×	623	1,390	2,013
400×	588	1,232	1,820
Total	2480	5,429	7,909
#Patients	24	58	82

advancing state-of-the-art in automatic classification of histology images. A large annotated dataset of H&E stained breast histology images, composed of both microscopy and WSIs, was specifically compiled and made publicly available for the challenge. The challenge is composed of two parts. Part A is based on the microscopy images and dedicated for image classification task, while Part B is based on WSI and considered for image segmentation task. In our experiments, we consider only Part A since working directly on WSIs adds a heavy complexity to the learning algorithms in terms of memory and running time (Subsection 1.2.3).

The microscopy dataset is composed of 400 training images and 100 test images distributed evenly between four classes (image level labels): normal, benign, in Situ, and invasive. Figure 2.3 illustrates some examples from different classes. All images were acquired in 2014, 2015, and 2017 using Leica DM 2000 LED microscope and Leica ICC50 HD camera. All patients are from the Covilhã and Porto regions (Portugal). The annotation was performed by two medical experts. Images where there was disagreement between the normal and benign classes were discarded. The remaining doubtful cases were confirmed via immunohistochemical analysis. The provided images are in RGB Tagged Image File Format (TIFF). All the images have the same size (2,048, 1,536) pixels and a pixel scale of $(0.42\mu m, 0.42\mu m)$. (Aresta *et al.*, 2018) provide more details on the challenge and the provided data.

In our experiments, we consider a classification task with the four classes of the dataset. The challenge made public images of the train and test sets. However, only train labels are provided. A model prediction must be uploaded to the website of the challenge for evaluation on the test set. Only three trials are allowed per day. Therefore, we limit ourselves to use only the train set for training and evaluation using a cross-validation scheme. We take half of the samples of each class to build the test set and we apply k-fold over the left samples to build the validation and train set.

2.1.1.3 GlaS dataset

Colorectal adenocarcinoma originating in intestinal glandular structures is the most common form of colon cancer (Sirinukunwattana *et al.*, 2017). The morphology of intestinal glands, including architectural appearance and glandular formation is used in clinical practice by pathologists to inform prognosis and plan treatment of individual patients. Achieving good inter-observer as well intra-observer reproducibility of cancer grading is a major challenge in the pathology domain. The Gland Segmentation in Colon Histology Images Challenge Contest[5] (Sirinukunwattana *et al.*, 2017) was held in 2015 in the aim to advance automated approaches for quantifying the morphology of glands.

[5] https://warwick.ac.uk/fac/sci/dcs/research/tia/glascontest

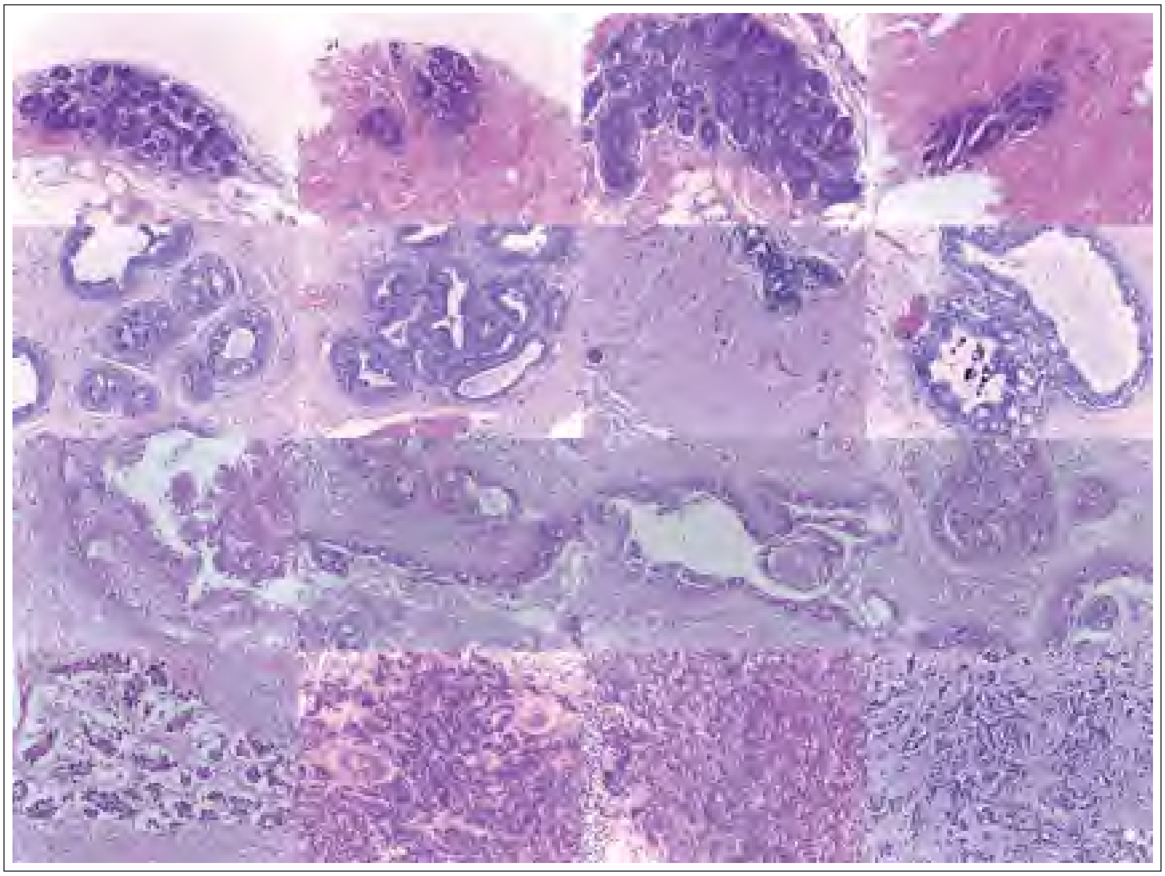


Figure 2.3 Samples from BACH public train dataset. *Row 1*: Normal. *Row 2*: Benign. *Row 3*: In Situ. *Row 4*: Invasive.

The challenge provides a dataset, GlaS, composed of 165 images derived from 16 H&E histological sections of two grades (classes): benign, and malignant (Figure 2.4). The digitization of these histological sections into WSI was accomplished using a Zeiss MIRAX MIDI Slide Scanner with a pixel resolution of $0.465\mu m$. The WSI were subsequently rescaled to a pixel resolution of $0.620\mu m$ (equivalent to 20× objective magnification). Table 2.3 summarizes the partitioning of the dataset with the images size details. (Sirinukunwattana *et al.*, 2017) provide more information on the dataset.

Since the challenge was primarily made for segmentation, a ground truth of the glandes segmentation is provided (pixel-level annotation). Aside the segmentation labels, image-level labels are also provided with two classes: benign, or malignant.

In this dataset, the glandes are the regions of interest that the pathologists use to prognosis the image grading of being benign or malignant. Therefore, in our later experiments (Section 2.2), we are interested in measuring how well the model relies on such medically-valid regions of interest to predict the global class of the image. Therefore, the localized regions by the model are considered as a visual interpretability tool to justify the model's decision. We note that only image-level labels are used during the training, while pixel-level labels are used to evaluate the accuracy of localizing regions of interest (*i.e*. glandes). In our experiments, Test Part A, B are mixed (Table 2.3).

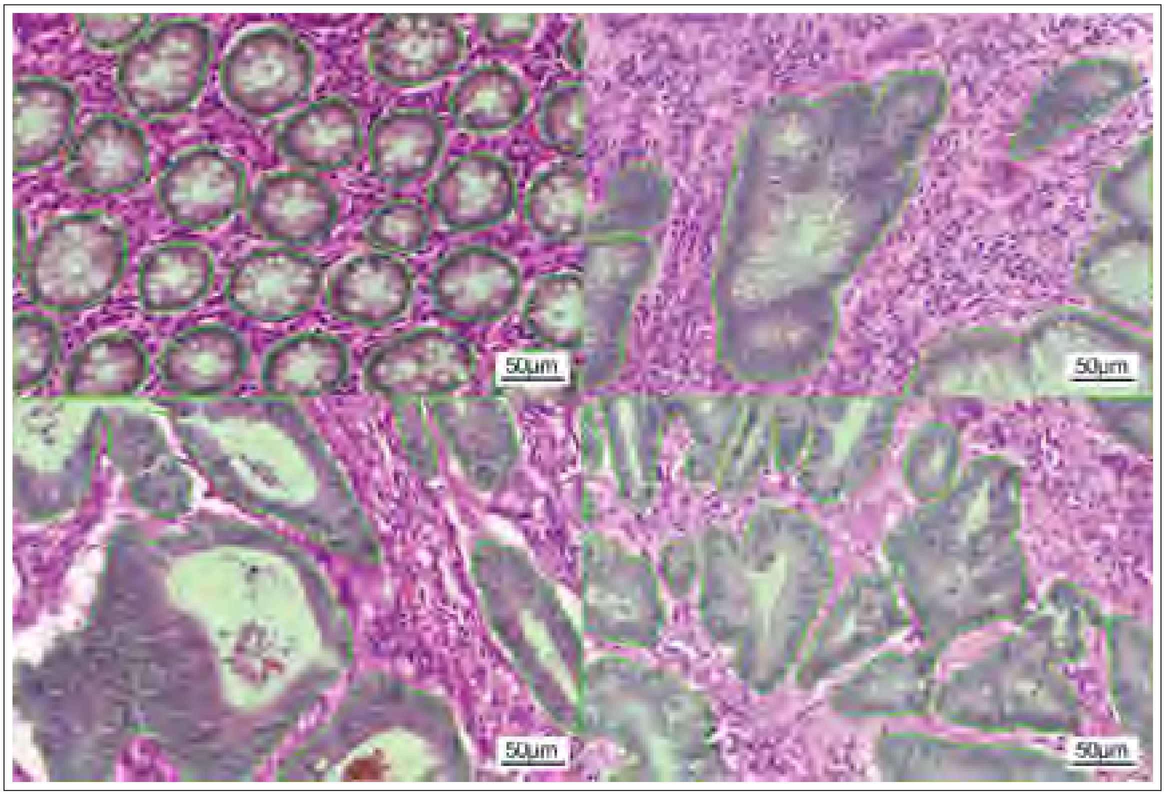


Figure 2.4 Example of images of different classes with their segmentation from GlaS dataset (Credit: (Sirinukunwattana *et al.*, 2017)). *Row 1*: Benign. *Row 2*: Malignant.

Table 2.3 Details of the GlaS dataset (Sirinukunwattana *et al.*, 2017)

Histologic Grade	Number of Images ((Width, Height) in Pixels)						
	Training Part			Test Part A			Test Part B
Benign	37	1 1 35	(574, 433) (589, 453) (775, 522)	33	1 4 28	(574, 433) (589, 453) (775, 522)	4 (775, 522)
Malignant	48	1 3 44	(567, 430) (589, 453) (775, 522)	27	1 2 24	(578, 433) (581, 442) (775, 522)	16 (775, 522)

2.1.1.4 CAMELLYON16 dataset

The Cancer Metastases in Lymph Nodes Challenge 2016 (CAMELLYON16)[6] competition (Ehteshami Bejnordi *et al.*, 2017) was organized to investigate the potential of machine learning algorithms for detection of metastases in H&E stained tissue sections of sentinel auxiliary lymph nodes (SNLs) of women with breast cancer.

The organizers of the challenge collected 399 WSIs of SNLs during the first half of 2015. SNLs were retrospectively sampled from 399 patients that underwent surgery for breast cancer at 2 hospitals in the Netherlands: Radbound University Medical Center (RUMC) and University Medical Center Utrecht (UMCU). The need for informed consent was waived by the institutional review board of RUMC. The WSIs were acquired at two different centers using two different scanners. RUMC images were produced with a digital slide scanner (Pannoramic 250 Flash II; 3DHISTECH) with a 20× objective lens (specimen-level pixel size, $0.243\mu m \times 0.243\mu m$). UMCU images were produced using a digital slide scanner (NanoZoomer-XR Digital slide scaner C12000-01; Hamamatsu Photonics) with a 40× objective lens (specimen-level pixel size, $0.226\mu m \times 0.226\mu m$). The WSIs are annotated globally to normal or metastases. The WSIs with metastases are further annotated at pixel-level to indicate regions of tumors. The annotations were first drawn by two students (one from each hospital), and then every slide was checked in details by one of the two expert pathologists (one from RUMC and the second from UMCU). In

[6] https://camelyon16.grand-challenge.org/Home

case of uncertainty, pathologist opt to use immunohistochemistry to resolve the diagnostic. An example of a WSI is provided in Figure 2.5. Among the provided 399 WSIs, 270 are used for training (111, 159 with and without nodal metastases), and 129 for test (49, 80 with and without nodal metastases)[7]. (Ehteshami Bejnordi *et al.*, 2017) provide more details on the dataset, the challenge, and its final results.

Two tasks were defined in the challenge: identification of individual metastases in WSIs (task 1), and classification of every WSI as either containing or lacking SNL metastases (task 2). The WSIs are extremely large (many gigabytes per image and a resolution of $\sim 100,000^2$ pixels for each image) which makes it inconvenient to conduct our experiments for the purposes of this survey. Therefore, we consider neither of the two tasks. However, we design a concise protocol to assess the different models' capacity in localizing regions of interest at pixel-level (Subsection 2.1.2). Our protocol consists in building a weakly supervised learning scenario for pixel-wise localization through a binary classification task (normal against metastases) where we have both pixel-level and image-level labels and only image-level labels are used for training. In order to build train, validation, and test sets, we sample a set of patches from the WSIs where a patch is given an image-level label of metastases or normal depending on whether it contains cancerous pixels or not. If it is a metastatic patch, a binary mask that indicates the cancerous pixels is constructed based on the WSI pixel-level annotation.

2.1.2 Experimental protocol

2.1.2.1 Performance metrics

In our experiments, we seek to evaluate a model's capacity to accurately classify an image, and locate regions of interest at pixel-level. As described below, we consider two types of evaluation metrics: Evaluation A and Evaluation B.

[7] Sample `test_114` is discarded since the pixel level annotation was not provided. Therefore, the test set is composed of 128 samples with 48 samples with nodal metastases.

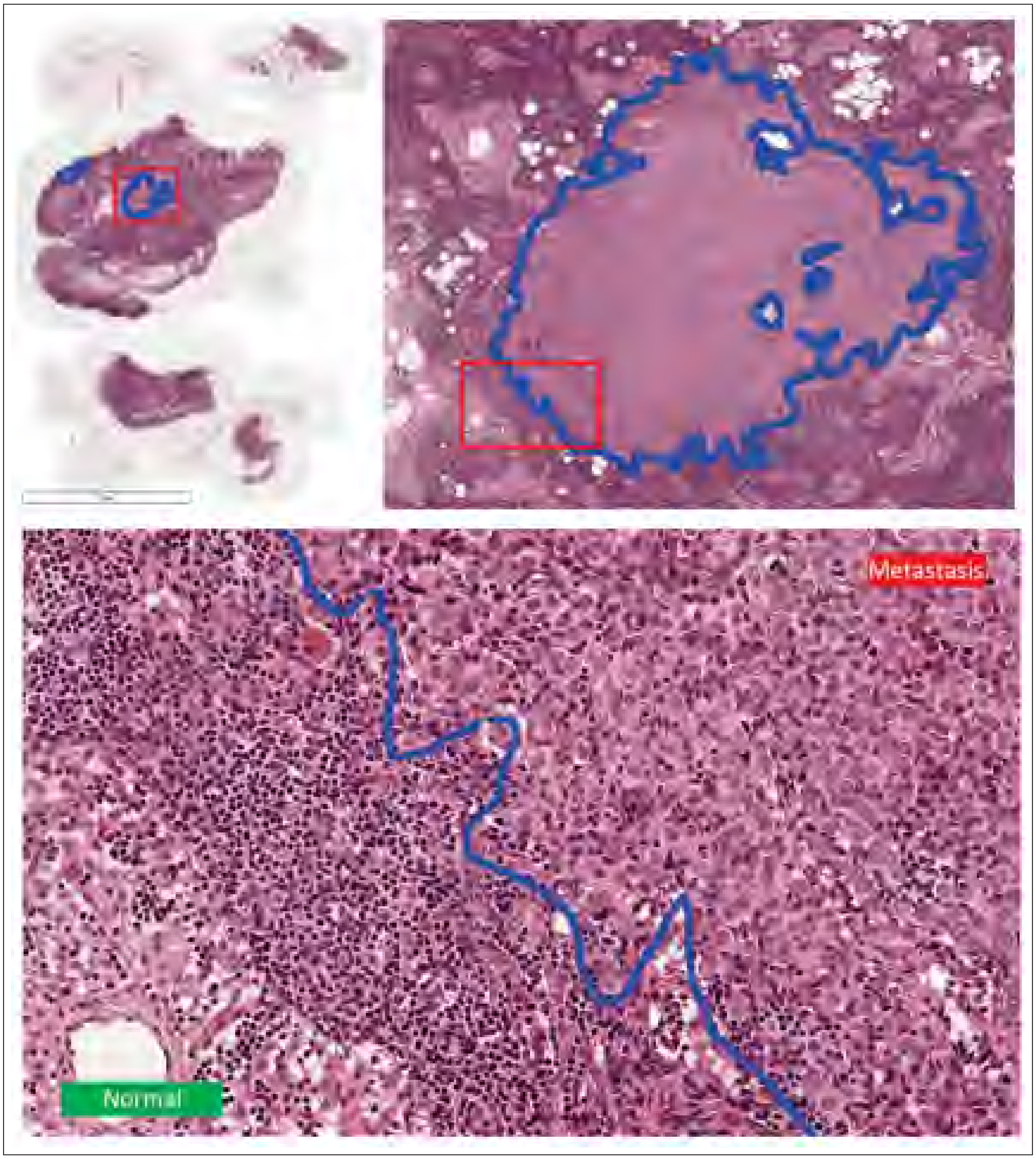


Figure 2.5 Example of metastatic regions in a WSI from CAMELYON16 dataset (Credit: (Sirinukunwattana *et al.*, 2017)). *Top left*: WSI with tumor. *Top right*: Zoom to one of the metastatic regions. *Bottom*: Further zoom into the frontier between normal and metastatic regions.

Evaluation A. In this evaluation setup we focus only on a model's performance in terms of classification. As evaluation metric, we consider using the accuracy measure:

$$\text{accuracy} = 100 \, \frac{\#\text{correctly classified samples}}{\#\text{samples}} \, (\%) \,, \tag{2.1}$$

where *#correctly classified samples* is the total number of correctly classified samples, and *#samples* is the total number of samples; and the mean Average Precision measure (Su, Yuan & Zhu, 2015) (mAP),

$$\text{mAP} = \frac{1}{c}\sum_{k=1}^{c} \text{AP}_k \; 100 \quad (\%) \,, \tag{2.2}$$

where c is the total number of classes, and AP_k is the average precision of the class k. A perfect model has 100% accuracy and 100% mAP. In this setup, all the relevant models are evaluated on the BreaKHis (Spanhol *et al.*, 2016b) and BACH (Part A) (Aresta *et al.*, 2018) datasets. The image-level labels are the only required supervised annotation for the training and evaluation of the models.

Evaluation B. In this evaluation setup we focus mainly on the performance of a model in terms of pixel-wise localization of regions of interest. To this end, we rely on standard segmentation metrics. In our experiments, we consider Dice index metric (Dice, 1945) which is a measure of agreement or similarity between two sets of samples. Give $\mathbb{G}$ a set of pixels belonging to a ground truth object, and $\mathbb{S}$, a set of pixels belonging to a segmented object. Dice index is defined as follows,

$$\text{Dice}(\mathbb{G}, \mathbb{S}) = \frac{2|\mathbb{G} \cap \mathbb{S}|}{|\mathbb{G}| + |\mathbb{S}|} \,, \tag{2.3}$$

where $|\cdot|$ denotes set cardinality, $\mathbb{G}\cap\mathbb{S}$ is the set of overlapped pixels between $\mathbb{G}$ and $\mathbb{S}$. Dice index ranges in the interval [0, 1], where the higher the value, the more concordant the segmentation and the ground truth. A Dice index of 1 indicates a perfect segmentation. To compute the Dice index, the pixel-level annotation is required. In the context of evaluating weakly supervised localization models, such annotation is exclusively used for evaluation –*i.e.* it is not used for training–. Only image-level annotation is used for training. GlaS (Sirinukunwattana *et al.*, 2017), and a variant of CAMELYON16 (Ehteshami Bejnordi *et al.*, 2017) (subsubsection 2.1.2.3) datasets are used for this evaluation. Classification performance at image-level is reported as well.

The aim of this type of evaluation is to measure how well a model, that is trained over a weakly supervised localization task –*i.e*. using only image-level annotation and deprived from pixel-level annotation–, can localize regions of interest that practically require pixel-level supervision. Such weakly supervised models are compared with an ideal model that is trained for segmentation only using pixel-level annotation without image-level annotation –*i.e*. segmentation task–. In this context, we consider using the model U-Net (Ronneberger, Fischer & Brox, 2015) which is a reference model in medical image segmentation. Such model is trained exclusively on pixel-level annotation: In the case of GlaS we train the model to segment the glands; while in the case of CAMELYON16 we train it to segment cancerous regions.

We note that in the case of the GlaS dataset (subsubsection 2.1.1.3), and the case of weakly supervised model that outputs two features maps to indicate regions of interest, Dice index is computed using the heat map corresponding to the true image-level label. In the case of CAMELYON16 dataset (subsubsection 2.1.1.4), Dice index is computed with respect to the heat map that corresponds to the metastatic class. This implies that we perform the evaluation only on the samples with metastatic image-level label. We also compute average Dice index over both metastatic and normal classes.

2.1.2.2 Datasets organization

In our experiments, the test set is fixed in all the datasets, and only train and validation sets are changed using a k-fold scheme. The only exception to the fixed test set rule is BreakHis where we use the provided folding (Spanhol *et al.*, 2016a) where each fold, and each magnification has its own test set. However, we apply the k-fold over the provided train set to obtain the train and validation sets. In our experiments, given the provided train set, we take 20% of the samples for validation, and 80% for actual training. This leads to 5-folds partitioning. We report the mean and standard deviation of each metric over the trials in the following form: mean ± standard deviation. We note that BreaKHis and BACH (Part A) datasets are used for **Evaluation A** (subsubsection 2.1.2.1) while GlaS and CAMELYON16 are used for **Evaluation B** (subsubsection 2.1.2.1). The results of our experiments on BreaKHis, BACH (Part A), GlaS, and CAMELYON16 are presented in Tables 2.5, 2.6, 2.7, 2.8 and 2.9. We note that in order to compute Dice index for a set, we average Dice index of each image, unless stated otherwise.

The deterministic code used to create the folds of all the datasets, sampling from CAMELYON16, the coordinates of the sampled patches from CAMELYON16, and the code of all the experiments is publicly available[8].

2.1.2.3 CAMELYON16 protocol for weakly supervised localization

We describe in this section our protocol of creating a weakly supervised localization dataset from CAMELYON16 dataset (Ehteshami Bejnordi *et al.*, 2017). Samples are patches from WSIs, and each patch has two levels of annotation:

- Image-level label y: the class of the patch, where $y \in \{\texttt{normal}, \texttt{metastatic}\}$.
- Pixel-level label $\boldsymbol{Y} = \{0, 1\}^{H^{\text{in}} \times W^{\text{in}}}$: a binary mask where the value 1 indicates a `metastatic` pixel, and 0 a `normal` pixel. For `normal` patches, this mask will contain 0 only.

First, we split CAMELYON16 dataset into train, validation, and test set at *WSI-level* as described in subsubsection 2.1.2.1. This prevent patches from the same WSI to end up in different sets. All patches are sampled with the highest resolution from WSI –*i.e*. level = 0 in WSI terminology–. We present in the following our methodology of sampling metastatic and normal patches.

Sampling metastatic patches. Metastatic patches are sampled only from metastatic WSIs around the cancerous regions. Sampled patches will have image-level label, and a pixel-level label. The sampling follows these steps:

1. Consider a metastatic WSI.
2. Sample a patch $\mathbf{x}$ with size (H, W).
3. Binarize the patch into $\boldsymbol{x}^{\text{b}}$ mask using OTSU method (Otsu, 1979). Pixels with value 1 indicate tissue.

[8] https://github.com/jeromerony/survey_wsl_histology

4. Let $p_t^{\boldsymbol{x}^{\mathrm{b}}}$ be the tissue percentage within $\boldsymbol{x}^{\mathrm{b}}$. If $p_t^{\boldsymbol{x}^{\mathrm{b}}} < p_t$, discard the patch.
5. Compute the metastatic binary mask $\boldsymbol{Y}$ of the patch $\mathbf{x}$ using the pixel-level annotation of the WSI (values of 1 indicate a metastatic pixel).
6. Compute the percentage $p_m^{\boldsymbol{x}}$ of metastatic pixels within $\boldsymbol{Y}$.
7. If $p_m^{\mathbf{x}} < p_0$, discard the patch. Else, keep the patch $\mathbf{x}$ and set $y = \texttt{metastatic}$ and $\boldsymbol{Y}$ is its pixel-level annotation.

We note that we sample *all* possible metastatic patches from CAMELYON16 using the above approach. Sampling using such approach will lead to a large number of metastatic patches with high percentage of cancerous pixels (patches sampled from the center of the cancerous regions). These patches will have their binary annotation mask $\boldsymbol{Y}$ full of 1. Using these patches will shadow the performance measure of localization of cancerous regions. To avoid this issue, we propose to perform a calibration of the sampled patches in order to get rid of most of such patches. We define two categories of metastatic patches:

1. **Category 1**: Contains patches with $p_0 \leq p_m^{\mathbf{x}} \leq p_1$. Such patches are rare, and contain only small region of cancerous pixels. They are often located at the edge of the cancerous regions within a WSI.
2. **Category 2**: Contains patches with $p_m^{\mathbf{x}} > p_1$. Such patches are extremely abundant, and contain a very large region of cancerous pixels (most of the time the entire patch is cancerous). Such patches are often located inside the cancerous regions within a WSI.

Our calibration method consists in keeping all patches within **Category 1** and throwing most of the patches in **Category 2**. To this end, we apply the following sampling approach:

1. Assume we have n patches in **Category 1**. We will sample $n\ p_n$ patches from **Category 2**, where p_n is a predefined percentage.
2. Compute the histogram of the frequency of the percentage of cancerous pixels within all patches. Assuming a histogram with b bins.
3. Among all the bins with $p_m^{\mathbf{x}} > p_1$, pick uniformly a bin.
4. Pick uniformly a patch within that bin.

This procedure is repeated until we sample $n\ p_n$ patches from **Category 2**. Table 2.4 presents the number of sampled patches from the entire CAMELYON16 dataset, before and after calibration. We note that the sampling of metastatic patches is done separately on the original provided train, and test sets of WSIs.

In our experiments, patches are not overlapping. We use the following configuration: $p_0 = 20\%$, $p_1 = 50\%$, $p_t = 10\%$, $p_n = 1\%$. The number of bins in the histogram is obtained by dividing the interval $[0, 1]$ with a delta of 0.05. We investigate the following patch sizes: $(512, 512)$, $(768, 768)$ and $(1024, 1024)$. In one experiment, only one patch size is used –*i.e.* patches with different sizes are not mixed within the same set–. Figure 2.6 illustrates an example of metastatic patches and their corresponding masks.

We note that metastatic patches are sampled then calibrated only once from the original train, and test WSI. Therefore, each WSI has a *unique* and *unchanged* set of metastatic patches.

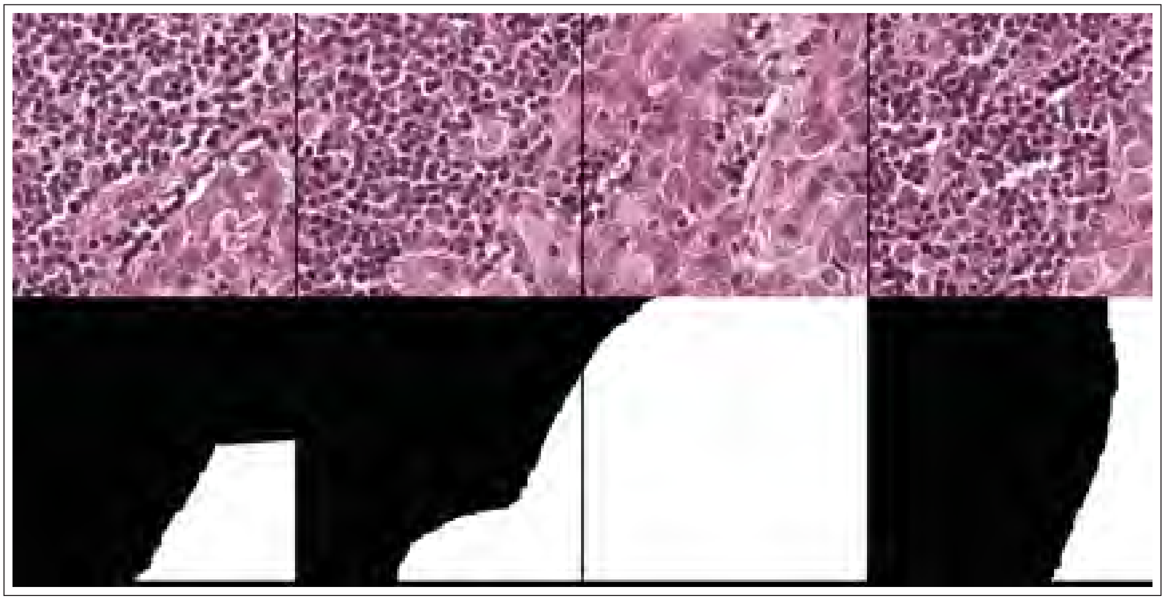


Figure 2.6 Example of metastatic patches with size $(512, 512)$ sampled from CAMELYON16 dataset (WSI: `tumor_001.tif`). *Top row*: Patches. *Bottom row*: Masks of metastatic regions (white color).

Sampling normal patches. Normal patches are sampled only from normal WSI. A normal patch is sampled randomly and uniformly from the WSI (without repetition nor overlapping).

Table 2.4 Total number of metastatic patches sampled from the entire CAMELYON16 dataset (Ehteshami Bejnordi *et al.*, 2017) using our sampling approach; and different patch sizes. Patches are not overlapping.

	#Patches: Before calibration			#Patches: After calibration	
Patch size	Total	$p_0 \leq p_m^{\mathbf{x}} \leq p_1$	$p_m^{\mathbf{x}} > p_1$	Total	$p_m^{\mathbf{x}} > p_1$
512×512	137,769	14,912	122,857	24,435	9,523
(768, 768)	64,127	9,512	54,615	15,377	5,865
(1,024, 1,024)	37,598	6,988	30,610	11,470	4,482

If the patch has enough tissue ($p_t^{\boldsymbol{x}^{\mathrm{b}}} \geq p_t$), the patch is accepted. The measure of tissue mass is performed at level = 6 where it is easy for the OTSU binarization method to split the tissue from the background. We double-check the tissue mass at level = 0.

Let us consider a set (train, validation, or test) at patch level within a specific fold. We first pick the corresponding metastatic patches from the metastatic WSI, assuming n_m is their total number. Assuming there is h normal WSIs in this set, we sample the same number of normal patches as the total number of metastatic ones. In order to mix the patches from all the normal WSI, we sample $\frac{n_m}{h}$ normal patches per normal WSI. In our experiment, we use the same setup as in the case of sampling metastatic patches: $p_t = 10\%$. Figure 2.7 illustrates an example of normal patches. This sampling procedure implies that metastatic patches are fix in all the metastatic WSI (also, they are the same across folds), while normal patches within a normal WSI change across folds.

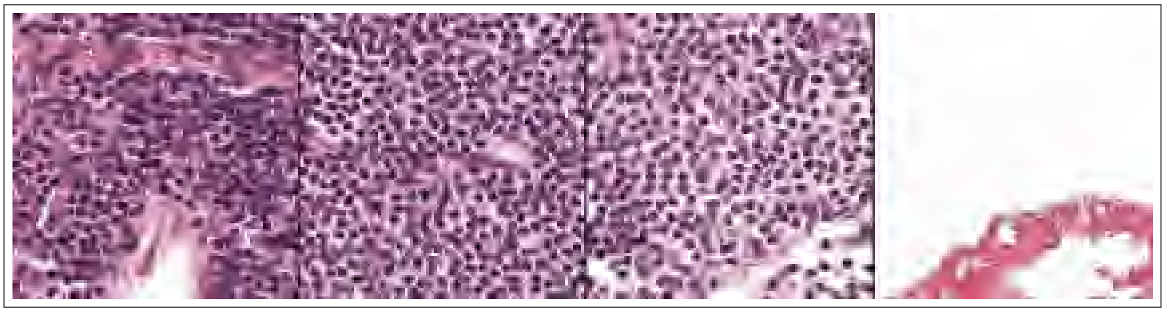


Figure 2.7 Example of normal patches with size (512, 512) sampled from CAMELYON16 dataset (WSI: `normal_001.tif`)

2.1.3 Training setup

We present in this section the training setup that we used in our experiments for both learning methods: weakly supervised, and fully supervised training.

2.1.3.1 Weakly supervised training

For all methods, we use an ImageNet-pretrained ResNet-18 (He, Zhang, Ren et al., 2016) architecture as a feature extractor. The optimization algorithm used is Stochastic Gradient Descent (SGD) with Nesterov acceleration with a momentum of 0.9 and a weight decay of 0.0001. The learning rate is set to 0.01 for the first half of the training and decayed to 0.001 for the second half. The minibatch size is set to 64 for all datasets except GlaS where it is set to 32. For all datasets, we randomly flip the images during training. We also perform random color jittering on the images with parameters brightness, contrast and saturation at 0.5 and hue at 0.05 (from the PyTorch framework).

To train models using mini-batches, we must scale images to a common size. Since the images have different sizes between datasets and within some datasets, the cropping and resizing strategies differ between datasets. Even though we may use only a part of an image to train, cropped patches inherit the same label of the entire image.

For BACH dataset, the image size is large $(2048, 1536)$. Therefore, we train on patches extracted from the images rather than on the full images; and each patch receives the image label. The extracted patches have a size of $(512, 512)$ at random locations and random rotations while ensuring that no empty zone is included (which happens when sampling too close to the borders of the image depending on the angle of rotation). We train all the models during 20 epochs. For the BreakHis and GlaS datasets, we extract patches of size $(448, 448)$ and $(416, 416)$ respectively at random locations and rotate them with a random angle in $\{0°, 90°, 180°, 270°\}$. For the BreakHis dataset, the models are trained during 80 epochs and for GlaS dataset, they are trained for 160 epochs since the number of samples is very small. For CAMELYON16 dataset, the images are simply rotated with a random angle in $\{0°, 90°, 180°, 270°\}$. The models are trained for 20 epochs.

In the case of the LSE and WILDCAT pooling, the hyper-parameters are chosen depending on the recommended values in their original papers. For the LSE pooling, we use $q = 10$. For WILDCAT, we set n^+ and n^- to correspond to 10% of highest and lowest scoring instances each and $\alpha = 0.6$.

For BreakHis and BACH datasets, we also study the impact of the number of training samples by training with only a fraction of each dataset. For both datasets, we do not change the size of the validation set nor the test set. To reduce the size of the training set, we randomly sample a given fraction of the examples in each class to keep the same balance between the classes. For BACH, we use the following percentages: $10\%, 25\%, 50\%, 75\%, 100\%$ which corresponds to the following number of training samples *per class*: $4, 10, 20, 30, 40$, respectively. For BreakHis, we use the following percentages: $4\%, 10\%, 25\%, 50\%, 75\%, 100\%$. Since each magnification has a different number of samples, and for clarity, we prefer not to mention the per class number of samples. However, they can be easily computed based on Table 2.2.

2.1.3.2 Fully supervised training

We also train a fully supervised U-Net architecture (Ronneberger *et al.*, 2015) on GlaS and CAMELYON16 datasets to obtain an upper bound performance in terms of pixel-wise localization in a fully supervised setting. For both datasets, we train using SGD with Nesterov acceleration with a momentum of 0.9 and a weight decay of 0.0001. The learning rate is set to 0.1 and decayed during training depending on the number of epochs to reach 0.001 at the end of the training. For both datasets, the mini-batch size is set to 16.

For GlaS dataset, the model is trained for 960 epochs and the learning rate is divided by 10 every 320 epochs. For CAMELYON16 dataset, the model is trained for 90 epochs and the learning rate is divided by 10 every 30 epochs. We use the same augmentations as in the weakly-supervised training for both datasets (subsubsection 2.1.3.1).

2.2 Results, interpretation, and future directions

2.2.1 Evaluation A – classification performance

Table 2.5 **Evaluation A**: Accuracy and AP over test folds of BreakHis (Spanhol *et al.*, 2016b) dataset using different magnification factors and different models

Magnification	40×		100×		200×		400×	
Method	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**
CAM - Average	92.19 ± 3.54	97.80 ± 2.30	89.64 ± 2.93	98.10 ± 0.91	91.03 ± 1.33	98.36 ± 0.54	85.09 ± 2.09	96.04 ± 0.99
Max	90.09 ± 2.89	97.64 ± 2.01	88.11 ± 3.08	97.75 ± 1.22	90.41 ± 2.66	98.21 ± 0.70	84.00 ± 1.95	95.44 ± 1.16
LSE	89.52 ± 3.68	97.04 ± 2.96	89.57 ± 3.24	97.92 ± 1.00	90.15 ± 1.96	98.08 ± 0.79	84.86 ± 1.98	95.16 ± 1.74
WILDCAT	92.40 ± 2.82	97.90 ± 2.41	90.22 ± 2.48	97.99 ± 1.55	90.75 ± 2.00	98.49 ± 0.59	85.85 ± 3.05	96.41 ± 1.40
Deep MIL	91.80 ± 2.70	98.38 ± 1.64	89.54 ± 3.14	97.69 ± 1.16	91.61 ± 1.34	98.71 ± 0.55	85.98 ± 2.28	96.29 ± 0.85

Table 2.6 **Evaluation A**: Accuracy and mAP over test folds on the BACH (Part A) dataset (Aresta *et al.*, 2018) using different models

Method	**Accuracy (%)**	**mAP (%)**
CAM - Average	84.10 ± 2.51	93.23 ± 1.27
Max	76.10 ± 3.60	87.34 ± 4.44
LSE	78.90 ± 4.29	88.73 ± 2.77
WILDCAT	84.80 ± 1.25	93.04 ± 1.00
Deep MIL (adapted)	83.30 ± 3.90	92.68 ± 2.71

Table 2.7 **Evaluation A and B**: Dice index, accuracy over the test folds of GlaS (Sirinukunwattana *et al.*, 2017) dataset using different models. AP is not shown since weakly supervised localization methods always achieve 100%.

Method	**Dice index**	**Accuracy (%)**
CAM - Average	68.43 ± 0.73	99.75 ± 0.56
GradCAM	68.48 ± 0.72	99.75 ± 0.56
Max	67.51 ± 2.48	99.75 ± 0.56
LSE	65.61 ± 3.73	99.50 ± 0.68
WILDCAT	68.62 ± 0.61	100 ± 0
Deep MIL (adapted)	72.13 ± 1.78	99.25 ± 1.12
U-Net	90.54 ± 0.88	-

Tables 2.5, 2.6, 2.7 and 2.8 present the obtained results in terms of classification performance for the different models on the four datasets in terms of accuracy and average precision (AP) or mean

Table 2.8 **Evaluation A**: Accuracy and AP obtained with weakly supervised localization models for test folds of CAMELYON16 (Ehteshami Bejnordi *et al.*, 2017) dataset using growing patch size

Patch size	(512, 512)		(768, 768)		(1,024, 1,024)	
Method	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**
CAM - Average	98.54 ± 0.36	99.80 ± 0.04	98.37 ± 0.45	99.89 ± 0.03	99.20 ± 0.28	99.92 ± 0.02
Max	98.51 ± 0.22	99.80 ± 0.04	98.62 ± 0.27	99.90 ± 0.04	87.07 ± 11.59	92.45 ± 12.29
LSE	98.62 ± 0.22	99.80 ± 0.04	98.77 ± 0.22	99.92 ± 0.02	93.41 ± 0.67	98.44 ± 0.27
WILDCAT	98.37 ± 0.65	99.84 ± 0.05	98.62 ± 0.36	99.92 ± 0.03	99.16 ± 0.17	99.95 ± 0.01
Deep MIL	98.15 ± 0.59	99.82 ± 0.04	98.34 ± 1.16	99.82 ± 0.23	99.17 ± 0.11	99.95 ± 0.01

average precision (mAP) in the case of a multi-class dataset. For this evaluation, the performance of Grad-CAM is not reported since the method is identical to CAM for classification.

We observe that all the studied methods achieved similar high classification performance on GlaS (Table 2.7) and CAMELYON16 (Table 2.8) datasets (~ 100%) with low variation. This suggests that both datasets are easy for a classification task. For BreakHis (Table 2.5) and BACH (Table 2.6), we observe that the classification performance is high and similar for CAM, WILDCAT and Deep MIL (even adapted to the multi-class scenario) and low for LSE and Max with generally high deviations. This confirms the observation made by Zhou *et al.* (2016) that LSE and Max pooling strategies tend to overfit more, especially when training on few samples. From Table 2.5, we can also observe that the classification performance (both in terms of accuracy and AP) is the best at 200× and 40× magnifications. This suggests that both level of magnification are ideal for BreakHis dataset with respect to the studied methods. This is further confirmed by the accuracy obtained when training only on a fraction of the dataset Figure 2.9. When using 25% of the training set, the accuracy of most methods is still at ~ 90% at magnification of 200× and 40× as opposed to 100× and 400× magnification. However, the magnification 200× seems to be more stable and robust toward the variation of the number of training samples. The magnification 400× shows to be the worse. These results suggest that the visual discriminative features for cancer grading using the studied approach are better observed at low magnification (zoom-out) –*i.e.* between 40× and 200×–. Zooming-in further into the histology image –*e.g.* 400×–, makes it difficult to discriminate between the different classes. This may also suggest that when zooming-in further, this type of imaging provides almost similar images independently from the cancer grade.

We also note that Deep MIL has high variation in terms of classification performance when the number of training samples is heavily reduced in the case of BreakHis dataset Figure 2.9 which suggests that such method needs a larger number of training sample to achieve a higher accuracy.

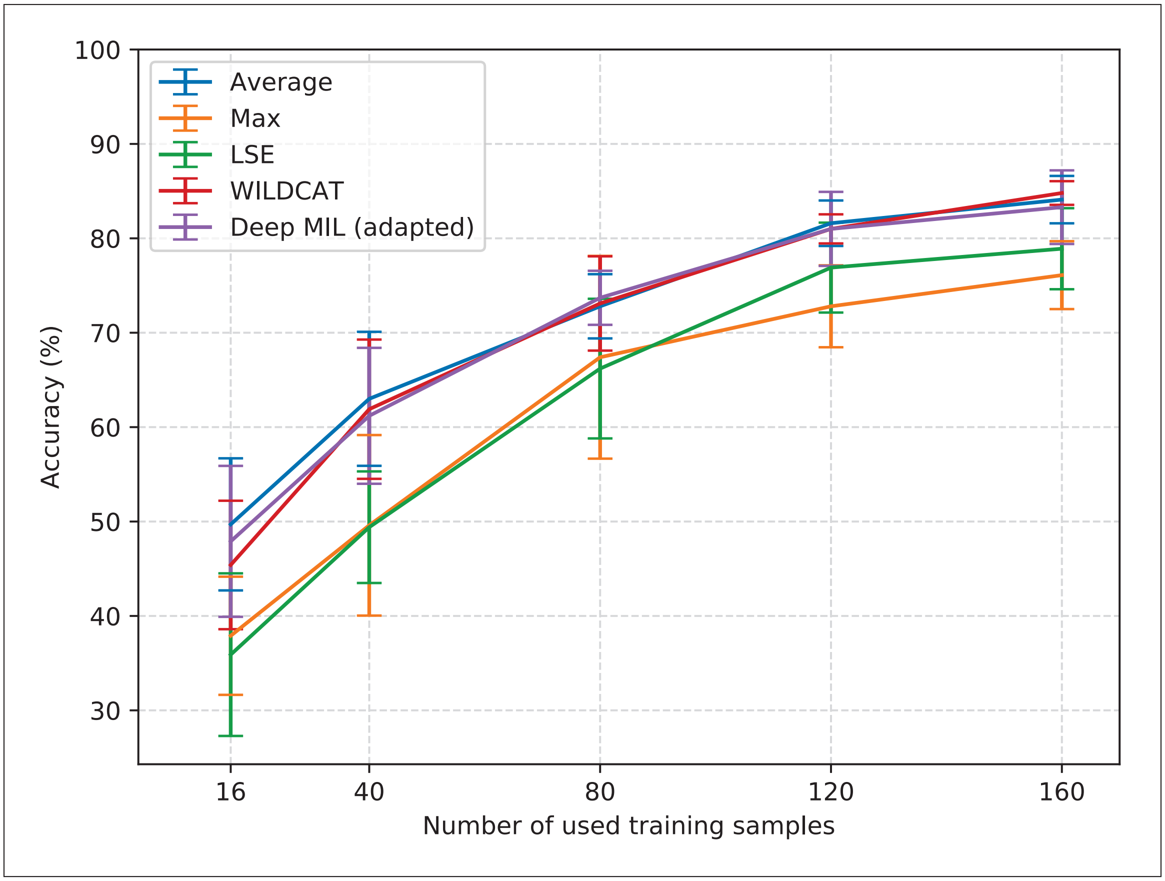


Figure 2.8 Influence of the training set size for the BACH dataset

In the case of BACH dataset, increasing the size of the training samples leads to an improvement in classification performance across all the methods (Figure 2.8) which is a typical behavior in ML algorithms.

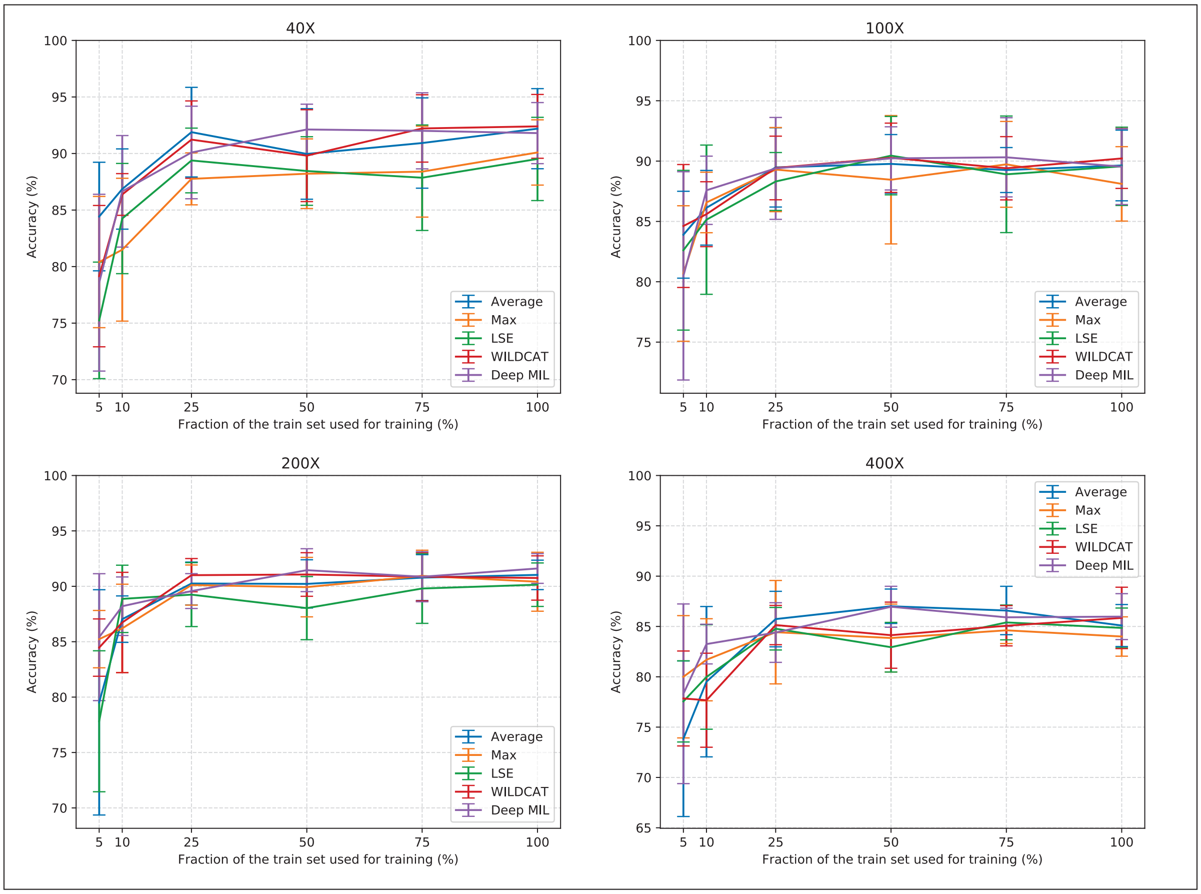


Figure 2.9 Influence of the training set size for the BreakHis dataset at magnification levels of 40× (top left), 100× (top right), 200× (bottom left) and 400× (bottom right)

Figures 2.8 and 2.9 also suggest that in order to obtain a good classification accuracy of ∼ 80%, models require at least ∼ 40 samples per class for BACH dataset, and ∼ 40 samples per class for BreakHis (∼ 10%).

2.2.2 Evaluation B – localization performance

Tables 2.7 and 2.9 correspond to the evaluation of different deep weakly supervised localization techniques over GlaS and CAMELYON16 datasets, respectively, in terms of localization performance. In terms of Dice index, we observe that Deep MIL obtains the best performance on GlaS dataset while Max and LSE pooling have worse results and higher variation. This suggests that both methods have a tendency to overfit more and they are more sensitive to data variations

Table 2.9 **Evaluation B**: Dice index over all pixels and averaged over all metastatic images (mDice metastatic) obtained with weakly supervised localization models for test folds of CAMELYON16 (Ehteshami Bejnordi *et al.*, 2017) dataset using a growing patch size.

Patch size	**(512, 512)**		**(768, 768)**		**(1,024, 1,024)**	
Method	**Dice index**	**mDice metastatic**	**Dice index**	**mDice metastatic**	**Dice index**	**mDice metastatic**
CAM - Average	65.06 ± 1.00	62.47 ± 1.17	64.16 ± 0.78	61.45 ± 0.67	66.70 ± 2.77	63.98 ± 2.93
GradCAM	65.18 ± 1.02	62.51 ± 1.19	64.14 ± 0.86	61.35 ± 0.75	66.75 ± 2.76	63.98 ± 2.95
Max	67.08 ± 1.33	62.31 ± 1.10	68.50 ± 1.43	63.51 ± 1.02	50.23 ± 12.45	55.71 ± 5.73
LSE	67.04 ± 1.30	62.78 ± 0.87	64.96 ± 3.96	61.71 ± 1.81	62.97 ± 7.34	60.19 ± 8.15
WILDCAT	66.06 ± 0.84	62.87 ± 0.86	67.24 ± 1.29	63.74 ± 1.33	66.05 ± 2.96	65.52 ± 1.28
Deep MIL	49.26 ± 2.81	69.76 ± 0.73	47.93 ± 2.66	68.49 ± 2.16	46.61 ± 2.81	65.97 ± 1.67
U-Net	77.68 ± 1.47	70.90 ± 2.37	79.90 ± 1.30	73.03 ± 2.05	80.72 ± 0.83	72.79 ± 1.47

through cross-validation. For the CAMELYON16 dataset, we notice that all the methods except Deep MIL achieve relatively similar Dice index over the entire test set. Max and LSE have much lower performance on large patches. This is due to the fact that these methods tend to overfit on small discriminative regions which represent a much small fraction of the images in a large patch which results in a low Dice index. The performance in terms of Dice index of Deep MIL is however not representative due the way we have adapted it. It tends to predict positive –*i.e.* metastatic– regions all over the images. For this reason, we also report the mean Dice index over metastatic images to measure how well it predicts positive regions. This measure shows that Deep MIL is able to correctly identify positive regions compared to other techniques.

In Chapter I, we provide visual examples of the predicted masks for pixel-wise localization produced by the different studied deep weakly supervised localization techniques over GlaS and CAMELYON16 test sets of the first split. From Figures I-1, I-2, I-3 and I-4, the main observation is the high false positive rate. The models are unable to correctly spot the right regions of interest, and, they tend to be active all over the image.

The deep WSL models have been developed, validated, and improved over the years mainly over natural scene images which reinforce many implicit priors of such type of images in their conception. We believe that applying them directly to histology images for weakly supervised localization tasks can lead to poor and unexpected results in terms of localization of regions of interest as illustrated visually in Figures I-1, I-2, I-3 and I-4. This behavior is mainly a direct result of the nature of histology images where regions of interest are highly unstructured, variable in size, multi-instance presence, and more importantly, non-salient. Often, in a histology image, regions of interest have similar visual appearance in terms of texture/color with respect to the background. This may potentially mislead the models and result in a high false positive rate. Therefore, applying the studied weakly supervised localization techniques in histology may require an adaptation to improve the selectivity of regions of interest and reduce false positive rate. A promising approach has been proposed recently where modeling irrelevant regions within the image is taken in consideration which allows to reduce false positives with a large gap (Belharbi, Rony, Dolz, Ben Ayed, McCaffrey & Granger, 2019). High false positive rate damages the interpretability aspect of weakly supervised localization techniques, and lower their usefulness in a weak localization task in histology images.

Another explanation to the high false positive rate is related to the pooling function. All the models are required to perform a spatial pooling to be able to classify the input image. A model is trained to predict either benign or malignant cancer grade by maximizing the probability of the target class. For a method such as CAM with Average pooling, this will attempt to maximize the probability of the target class by maximizing scores over every location in the image in order to maximize the global probability for that class since all locations in histology images are practically similar in terms of visual perception. Adding to this the issue of noisy labels of patches during training discussed in Subsection 1.2.3. As a result, this allows to consider non-discirminative regions as discriminative –*i.e.* consider noise/background as regions of interest–. Therefore, high false positive rate is increased. This problem may be addressed by adding more supervision in the form of size constraints (Jia, Huang, Eric, Chang & Xu, 2017) for instance. However, this will face the challenge of the high variation of the size of object of interest in histology images that goes from tiny regions to almost cover an entire image. Adding noise –*i.e.* uncertainty– to the target label at patch level may help reducing the issue of the inconsistency that may raise when transferring the image label to the patch label (Szegedy, Vanhoucke, Ioffe et al., 2016a).

In our evaluation, we only considered the fully-supervised U-Net model as an upper bound. It might be interesting to evaluate more recent state-of-the-art segmentation architectures such as DeepLabV3+ (Chen, Zhu, Papandreou, Schroff & Adam, 2018b) and HRNet (Sun, Zhao, Jiang, Cheng, Xiao, Liu, Mu, Wang, Liu & Wang, 2019) to get a stronger baseline, as these architectures combine the benefits of both ResNet-like architectures (which perform well in classification) and segmentation-specific architectural improvements.

2.2.3 Future directions

Based on our results, the application of deep WSL models for classification and localization in histology images showed that, in terms of classification, these techniques can achieve a high level performance. However, in terms of pixel-wise localization of regions of interest, these models lack accuracy, leading to localization with a high false positive rate, and potentially limiting the interpretability of a model's prediction. This is mainly due to the complex nature of histology images.

While the localization accuracy obtained with full pixel-level supervision comes at a high cost in terms of labeling, our results suggest that learning to localize without pixel-level labels can result in poor localization in histology images. As a potential compromise and a future direction, few-shot learning (Rakelly, Shelhamer, Darrell et al., 2018; Wang & Yao, 2019), where only very few samples are labeled at pixel-level, can be a promising research direction for histology image analysis. In this case, a pathologist labels only few relevant samples at pixel-level. However, such scarce but valuable annotation may provide a strong hint about the nature of regions of interest during learning, which in turn, will potentially reduce the false positive rate of localization. Given some limited interactions with a pathologist, active learning methods (Settles, 2009) will allow to selectively increase the number of annotated samples. This can be very helpful to deal with high resolution images. In such scenario, the model requests the pathologist to annotate at pixel-level the most relevant region at a training step. This prevents annotating irrelevant regions and reduce the pathologist's workload.

Finally, it is worth noting the challenges related to cancer grading, and its impact on image labels and regions localization. As mentioned in the key challenges of histology images, many grades may be present in the image, yet only the worse grade is provided as image label. Therefore, in order to improve region localization and reduce false positives, the learning process may also leverage the presence of multiple grades by exploiting multi-label learning scenario (instead of considering one single label in the image). This will improve DL model awareness, and prevent it from associating the entire image into one single label.

2.3 Conclusion

Training deep learning models for cancer grading and localization in histology images normally requires both image- and pixel-level labels. Given the high resolution of histology images, pixel-level labels require a costly and time consuming annotation process. Motivated by this issue, we explore the application of several state-of-the-art deep WSL models –initially proposed in the computer vision community (for natural images)– in histology image analysis, *without* pixel-level annotation.

This paper provides a survey on deep WSL models that are suitable for classification of histology images, and pixel-wise localization of regions of interest that correspond to class predictions. First we describe the process of histology image production, and outline the key challenges for their analysis. Then, we describe a taxonomy of suitable deep WSL techniques in the literature composed of bottom-up and top-down methods, where the former represents the more active in the research community. These methods are analysed with histology image analysis in mind. Promising methods are evaluated and compared experimentally in terms of accuracy (classification and pixel-wise localization) on four different public histology image datasets for breast and colon cancer – BreakHis, BACH, GlaS, and CAMELYON16. In order to provide more histology image benchmarks for large scale evaluation, we propose a concise protocol to build WSL datasets from Whole Slide Images (WSI). This protocol is used to create a new weakly supervised localization benchmark from the CAMELYON16 dataset. The results of our experimental study [9] show that the deep WSL models can provide a very high level of classification accuracy, but also suffer from a high false positive rate for pixel-wise localization.

[9] Public code: https://github.com/jeromerony/survey_wsl_histology

The latter suggests that specialized deep WSL models (e.g., (Belharbi *et al.*, 2019)) are required for pixel-wise localization in such large heterogeneous images, with highly non-salient and unstructured regions. Future research directions include improving performance by leveraging few relevant pixel-level annotations, through few-shot and active learning.

CHAPTER 3

DEEP MIL FOR MULTI-CLASS AND MULTI-LABEL CLASSIFICATION

3.1 Introduction

As shown in Chapter 1, most of the proposed feed-forward techniques that do not include prior on the content of the images have similar results in term of classification performance (except Max pooling) with a little advantage for Deep MIL in term of localization performance. However, Deep MIL was designed (and is intrinsically limited to) binary classification. Most real world applications have more than two classes so the goal of this chapter is to adapt the work of Deep MIL to the multi-class and multi-label scenarios.

As previously presented in Chapter 1, several methods have been proposed in the literature to obtain a segmentation model while training only with image-level labels. In particular, feed-forward methods are typically based on the Multiple Instance Learning (MIL) framework. In this framework, each image is represented as a bag of instances representing the different regions of the image. For each region of the image, a CNN is used to obtain a representation. This is typically done by feeding the entire input image to the CNN which downsizes it by a certain factor (called stride) depending on the architecture (*e.g*. 32 for ResNet (He *et al.*, 2016)). For classification tasks, we need a single representation for the whole image. This is usually achieved by computing the average of all the representations. However, to obtain a segmentation, the model must perform a prediction at each location. Therefore, a spatial average of the representations might not be the best strategy to force the network to be able to identify local features when learning from global image-level labels. In fact, the main contributions of the different feed-forward techniques that have been proposed is the strategy to go from a bag of instance representations to a classification by performing a pooling either on the instance representations or the instances scores.

3.2 Overview of the techniques

The different proposed techniques can be divided in two categories: the ones performing a pooling on the representations of the instances in the bag to obtain a single representation $\boldsymbol{f} \in \mathbb{R}^K$ of the bag and classify it; and the ones classifying each instance and pooling the scores to obtain a single score vector $\boldsymbol{s} \in \mathbb{R}^C$ for the image. In some cases, these two strategies are equivalent. Both strategies aim to obtain Class Activation Maps (CAMs) which are a segmentation of the images in the C classes. These CAMs are usually of lower resolution than the input but typically upsampled to match the input size. We will denote these CAMs as $\mathsf{M} \in \mathbb{R}^{C\times H\times W}$ before applying any upsampling. More formally, let us consider that we have an input image $x \in \mathbb{R}^{H^{\text{in}}\times W^{\text{in}}\times D}$ with H^{in}, W^{in} and D being the height, width and number of dimensions respectively. Given a model with stride S, we can extract features $\mathsf{F} \in \mathbb{R}^{K\times H\times W}$ where $H = H^{\text{in}}/S$, $W = W^{\text{in}}/S$ and K is the number of dimensions of the representations. To classify the image, we need to obtain a vector of score $\boldsymbol{s} \in \mathbb{R}^C$ with C being the number of classes of the problem.

One of the first work on WSL for image segmentation was done by Zhou *et al.* (2016) where they used the Global Average Pooling layer proposed in (Lin *et al.*, 2013). They noticed that using this strategy to spatially pool the features, one could obtain CAMs from the last classification layer. In this approach, f is obtained by performing an average (GAP) on the features:

$$\boldsymbol{f} = \frac{1}{HW}\sum_{i,j}\mathsf{F}_{i,j} \tag{3.1}$$

Where $\mathsf{F}_{i,j} \in \mathbb{R}^{K\times H\times W}$ is (by an abuse of notation) the representation of the instance at the index i and j from the spatial axes (of size H and W respectively). Then this representation $\boldsymbol{f}$ is classified through a classification function $\phi : \mathbb{R}^K \longrightarrow \mathbb{R}^C$: $\boldsymbol{s} = \phi(\boldsymbol{f})$. For most architectures (*e.g*. ResNet(He *et al.*, 2016), DenseNet(Huang, Liu, Van Der Maaten & Weinberger, 2017), ResNeXt(Xie, Girshick, Dollár, Tu & He, 2017), *etc*.), this classification function is a single fully connected layer. In that case, the score for the class c is:

$$\boldsymbol{s}_c = \boldsymbol{W}_c\boldsymbol{f} + \boldsymbol{b}_c \tag{3.2}$$

where $\boldsymbol{W} \in \mathbb{R}^{C\times K}$ and $\boldsymbol{b} \in \mathbb{R}^{C}$ are the weight and bias of the layer. The CAMs are then obtained by linearly combining the feature maps F:

$$\mathsf{M}_{c,i,j} = \boldsymbol{W}_c \mathsf{F}_{i,j} + \boldsymbol{b}_c \tag{3.3}$$

We can notice that since the classification layer is linear, performing the GAP directly on the CAMs to obtain the final classification scores is equivalent:

$$\begin{aligned} \boldsymbol{s}_c &= W_c(\frac{1}{HW}\sum_{i,j}\mathsf{F}_{i,j}) + \boldsymbol{b}_c \\ &= \frac{1}{HW}\sum_{i,j}\boldsymbol{W}_c\mathsf{F}_{i,j} + \boldsymbol{b}_c \\ &= \frac{1}{HW}\sum_{i,j}\mathsf{M}_{c,i,j} \end{aligned} \tag{3.4}$$

Usually, this results in small implementation difference where the fully connected layer is replaced by a convolution with a kernel of size 1×1. The CAMs obtained by this convolution are then spatially averaged to obtain the final image scores. This approach has also been generalized in (Selvaraju *et al.*, 2017) where $\boldsymbol{W}_{c,k}$ is replaced by $\boldsymbol{A}_{c,k}$ which is the derivative of the class score function w.r.t. the feature map:

$$\boldsymbol{A}_{c,k} = \frac{1}{HW}\sum_{i,j}\frac{\partial \boldsymbol{s}_c}{\partial \mathsf{F}_{k,i,j}} \tag{3.5}$$

This formulation is also equivalent when the final classification layer is linear.

Other works have considered replacing the final average pooling by a max pooling $\boldsymbol{s}_c = \max \mathsf{M}_c$ (Oquab *et al.*, 2015) however this pooling tends to focus the high scoring on very discriminative regions (Zhou *et al.*, 2016). To alleviate this problem, Pinheiro & Collobert (2015b) considered using a smooth approximation of the max function called Log-Sum-Exp

(LSE) (Boyd & Vandenberghe, 2004):

$$s_c = \frac{1}{q} \log \Big[\frac{1}{HW} \sum_{i,j} \exp(q \times \mathbf{M}_{c,i,j}) \Big] \tag{3.6}$$

where q is a hyper-parameter that controls the smoothness of the approximation. Small values (close to 0) of q will make the approximation close to the average and higher values will make it closer to the max function. However, this formulation is not numerically stable as it can cause both overflow and underflow because of the exponential. To make it more numerically stable:

$$\begin{aligned} s_c &= \frac{1}{q} \log \Big[\frac{1}{HW} \sum_{i,j} \exp(q(\mathbf{M}_{c,i,j} - \max \mathbf{M})) \exp(q \max \mathbf{M}) \Big] \\ &= \frac{1}{q} \log \Big[\exp(q \max \mathbf{M}) \frac{1}{HW} \sum_{i,j} \exp(q(\mathbf{M}_{c,i,j} - \max \mathbf{M})) \Big] \\ &= \frac{1}{q} (\log \big[\exp(q \max \mathbf{M}) \big] + \log \Big[\frac{1}{HW} \sum_{i,j} \exp(q(\mathbf{M}_{c,i,j} - \max \mathbf{M})) \Big]) \\ &= \max \mathbf{M} + \frac{1}{q} \log \Big[\frac{1}{HW} \sum_{i,j} \exp(q(\mathbf{M}_{c,i,j} - \max \mathbf{M})) \Big] \end{aligned} \tag{3.7}$$

Another approach that has obtained much more success in getting CAMs of better quality is the one proposed by Durand *et al.* (2017). In this formulation, the feature maps are first classified into C classes through a 1×1 convolution and then pooled using a novel strategy. Only a part of the instances are used in the pooling which considers both high and low scoring instances. Specifically, the final score for a given class will be:

$$\begin{gathered} s_c = \frac{1}{k^+} Z_c^+ + \frac{\alpha}{k^-} Z_c^- \\ \text{where} \quad Z_c^+ = \max_{Z \subset \mathbf{M}_c,\ |Z| = k^+} \sum_{z \in Z} z \quad \text{and} \quad Z_c^- = \min_{Z \subset \mathbf{M}_c,\ |Z| = k^-} \sum_{z \in Z} z \end{gathered} \tag{3.8}$$

Here, Z_c^+ and Z_c^- are the sum of the k^+ and k^- maximum and minimum scoring instances for that class and α is a weight for the negative scoring regions. This pooling selects only a part of the maximum scoring regions which alleviates the problem the max pooling and provides a regularization effect through the minimum scoring regions. In the original formulation, Durand *et al*. propose a class-wise pooling, in which, instead of classifying the feature maps in C classes, they are classified in $C \times M$ classes and averaged over the M modalities. We show that the choice of M has no impact on the results in Section 3.4.

Finally, a more recent approach proposed by Ilse *et al.* (2018) focuses on obtaining a representation of the bag using an attention mechanism to obtain weights for each instance and then classifying it. Specifically, the representation $\boldsymbol{f}$ is computed as:

$$\boldsymbol{f} = \sum_{i,j} \boldsymbol{A}_{i,j}\mathbf{F}_{i,j} \quad \text{where} \quad \boldsymbol{A}_{i,j} = \frac{\exp(\psi(\mathbf{F}_{i,j}))}{\sum_{i,j}\exp(\psi(\mathbf{F}_{i,j}))} \tag{3.9}$$

where $\psi : \mathbb{R}^K \longrightarrow \mathbb{R}$ is a scoring function. The final score s is obtained by classifying this representation of the bag: $s = \phi(\boldsymbol{f})$. In this formulation, $\boldsymbol{A}$ represents the importance of an instance in the final representation of the bag. This can be seen as a weighted average ($\sum \boldsymbol{A} = 1$ because of the softmax normalization) of the representations of the instances. However, this work was originally done for binary classification and is not straightforward to adapt to the multi-class scenario as there is no way to obtain CAMs.

3.3 Proposed Method

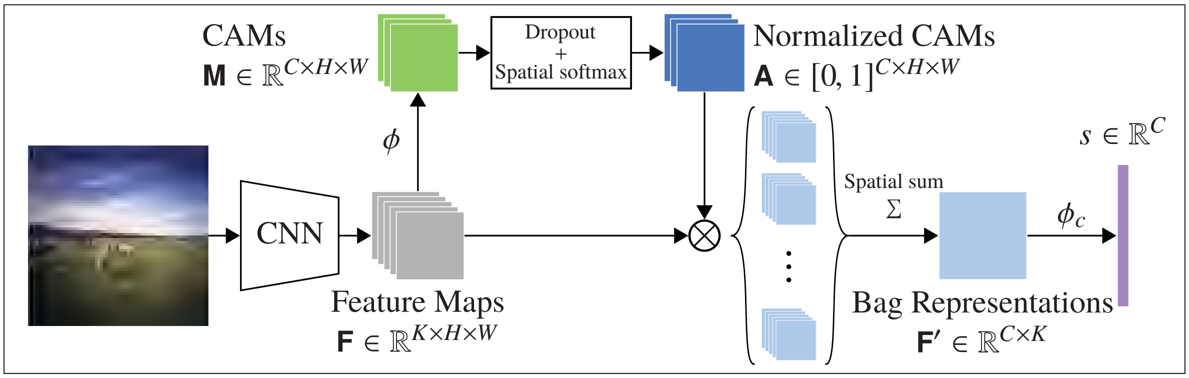

CAMs
$\mathbf{M} \in \mathbb{R}^{C\times H\times W}$
Dropout
+
Spatial softmax
Normalized CAMs
$\mathbf{A} \in [0,1]^{C\times H\times W}$
ϕ
CNN
Feature Maps
$\mathbf{F} \in \mathbb{R}^{K\times H\times W}$
Spatial sum
Σ
$s \in \mathbb{R}^C$
ϕ_c
Bag Representations
$\mathbf{F}' \in \mathbb{R}^{C\times K}$

Figure 3.1 Proposed pooling strategy. $\otimes$ denotes the tensor product done between the class axis of the normalized CAMs and the channel axis of the feature maps. ϕ_c denotes the c-th component of the output of ϕ

Building on the method of Ilse *et al.* (2018), we propose to compute a representation of the bag for each class of the problem illustrated in Figure 3.1. Specifically, instead of computing one score for each instance through a function $\psi : \mathbb{R}^K \longrightarrow \mathbb{R}$, we use the final classification function $\phi : \mathbb{R}^K \longrightarrow \mathbb{R}^C$ to compute a score for each instance of the bag. Then we use this score to obtain C weighted averages of the representations of the instances. Therefore, we obtain C representations of the input denoted as $\boldsymbol{F'} \in \mathbb{R}^{C \times K}$ with:

$$\begin{aligned} \boldsymbol{F}'_c &= \sum_{i,j} \mathbf{A}_{c,i,j} \mathbf{F}_{i,j} \\ \text{where} \quad \mathbf{A}_{c,i,j} &= \frac{\exp(\phi(\mathbf{F}_{i,j})_c)}{\sum\limits_{i,j} \exp(\phi(\mathbf{F}_{i,j})_c)} \end{aligned} \tag{3.10}$$

Then the score for class c is $\boldsymbol{s}_c = \phi(\boldsymbol{F}'_c)_c$. In that formulation, if ϕ is linear, we obtain:

$$\begin{aligned} \boldsymbol{s}_c &= \phi(\sum_{i,j} \mathbf{A}_{c,i,j} \mathbf{F}_{i,j})_c \\ &= \sum_{i,j} \mathbf{A}_{c,i,j} \phi(\mathbf{F}_{i,j})_c \\ &= \sum_{i,j} \mathbf{A}_{c,i,j} \mathbf{M}_{c,i,j} \end{aligned} \tag{3.11}$$

This means that the CAMs are used to compute the weights $\mathbf{A}$ to perform the weighted average over the instances of a bag.

In practice, this method tends to overfit quickly even with augmentation. To add more regularization to the pooling strategy, we randomly set a fraction d of the instance scores to 0 before the softmax normalization. This regularization randomly gives a lower weight to high scoring instances (which would have a score superior to 0) and gives a higher weights to lower scoring instances (which would have a score inferior to 0). We notice that this regularizes training and allows the network to discover larger parts of the objects. It also sets a fixed reference for activations at 0, leading to better foreground vs background performance.

3.4 Experiments

3.4.1 Dataset and Evaluation

We evaluate this method on the Pascal VOC 2012 Image Segmentation dataset Everingham, Van Gool, Williams, Winn & Zisserman (2010b). This dataset features 20 varied classes (*e.g.* aeroplane, dog, table, *etc*.) and a background class. We train a model using the augmented train set provided in (Hariharan, Arbelaez, Bourdev, Maji & Malik, 2011) which contains 10,582 images and test on the validation set containing 1,449 images. The performance is measured in term of mean Intersection over Union (mIoU) on the pixels averaged on the 21 classes (20 foreground classes and one background class) defined as:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{3.12}$$

where A and B are binary vectors representing the presence of a class for a pixel in our case. This measure, also known as the Jaccard index relates directly to the Dice index which becomes apparent when writing it in term of True Positives (TP), False Positives (FP) and False Negatives (FN):

$$IoU = \frac{TP}{TP + FP + FN} \qquad \text{and} \qquad Dice = \frac{2TP}{2TP + FP + FN} \tag{3.13}$$

thus:

$$\frac{2IoU}{1 + IoU} = \frac{2TP}{TP + FP + FN} \times \frac{1}{1 + \frac{TP}{TP+FP+FN}} = \frac{2TP}{2TP + TP + FP + FN} = Dice \tag{3.14}$$

The consequence of this relation is that the Dice index is always superior to the IoU as illustrated in Figure 3.2. We also report the mean Average Precision (mAP) over the 20 foreground classes of Pascal VOC.

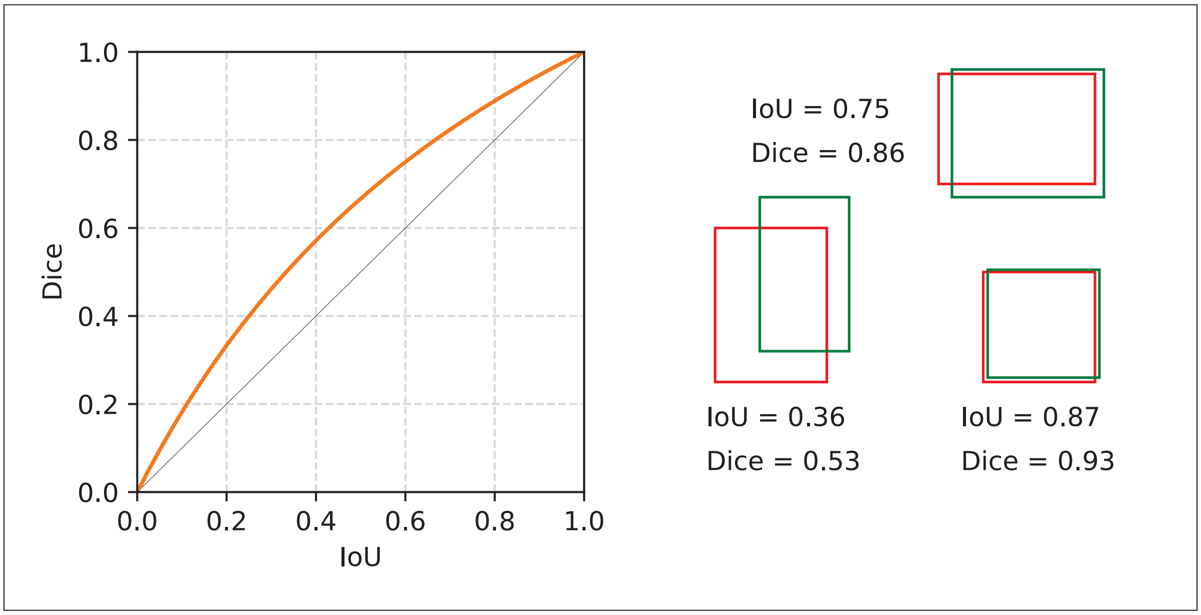


Figure 3.2 Relation between the IoU and the Dice index (left) with examples on rectangles (right).

3.4.2 Training and Inference

As a feature extractor we use the ResNet architecture (He *et al.*, 2016) pretrained on ImageNet-1000. Unless specified otherwise, we use a ResNet-18. This means that the stride of the CNN is 32 resulting in a segmentation of size 8×8 for an input image of size 256×256. Since we train using minibatches, we need to have images of the same sizes. Therefore, we resize all the training images to 448×448. We augment the training images with various transformations:

- random color jittering with parameters: brightness, contrast and saturation at 0.3 and hue at 0.1 (from the PyTorch framework).
- random cropping: we extract a patch at a random location with a random size between 224×224 and 448×448 and resize it to 448×448. This results in a random scaling which helps for the varying scales of the objects in the dataset.
- Gaussian blurring with a random radius in $[0, 2]$
- random horizontal flipping

These transformations typically do not change the classes present in an image except for the random cropping. The models are trained by minimizing a binary cross-entropy loss for 20 epochs with a minibatch size of 16 using SGD optimization algorithm with Nesterov momentum of 0.9 and a weight decay of 0.0001. The learning rate follows a cosine annealing decay (see Chapter II) starting at $\eta_0 = 0.01$. We set the probability of the dropout at $d = 0.3$.

Since we do not have access to the strong supervision provided by the segmentation masks, the model is not trained to predict the background class which is present in almost all images. Instead, we train the model to predict the 20 foreground classes. The background class is attributed to a pixel if no foreground class has a score higher than 0. In practice, this amounts to adding a 0 corresponding to the background to the score vector of each pixel and taking the $\arg\max$ of that vector to get the predicted label for each vector. This is also equivalent to applying a ReLU as in (Selvaraju *et al.*, 2017) in which a score of zero means that the pixels is predicted as a background pixel.

We also perform image augmentations at test time. More specifically, for each image, we compute the CAMs on the flipped image and at 4 different scales (0.5, 1, 1.5 and 2). This means we obtain 8 different CAMs that we resize to the input image original size using bilinear upsampling. Then we perform an average over these 8 sets of CAMs to obtain a single prediction.The evaluation of the mIoU is done on the entire images at the original resolution. We also perform this averaging over the image-level predictions obtained for each scale to obtain a single score per class and evaluate the mAP over the 20 foreground classes.

3.4.3 Modalities of WILDCAT

Table 3.1 Impact of the number of modalities on the performance of WILDCAT

Modalities	1	2	4	8	12	16	20
mAP (%)	92.1	92.6	92.4	92.1	91.7	91.8	91.6
mIoU (%)	37.9	38.3	38.3	38.7	38.7	37.8	38.2

To show that the number of modalities does not significantly impact the results of the WILDCAT method, we experiment with $M \in \{1, 2, 4, 8, 12, 16, 20\}$. Table 3.1 shows the impact of the number of modalities for WILDCAT with a ResNet-18 as a backbone on the Pascal VOC 2012 validation set. We observe that the results have low variation for a small number of modalities (< 10) and start to decrease as the number of modalities increases which may come from over-parametrization of the classification layer. The small differences might also come from the initialization (*i.e.* random seed used for each training). This is an expected result as averaging the results of a linear classification layers is equivalent to averaging the weights of this layer and computing the result. The only benefit of having multiple modalities is to reduce the probability of having a bad initialization for each class as it reduces the probability that a weight starts at an extreme value. On the other hand, having too many modalities will initiliaze the resulting weight (*i.e.* the average of the weights) to a value much closer to 0 (when initialized with a normal distribution) which can lead to poor performance (Glorot & Bengio, 2010).

3.4.4 Hyper-parameters

For the different methods, several hyper-parameters can be tuned and greatly impact the performance. For the LSE pooling, we experimented with values of $q \in \{0.1, 0.3, 0.5, 1, 3, 5, 10\}$. The best results (reported in Table 3.2) in term of mIoU were obtained for $q = 0.5$. For WILDCAT, the original hyper-parameters used to perform the segmentation on Pascal VOC were not made public [1], we performed a grid-search on the hyper-parameters. The WILDCAT method has three hyper-parameters: k^+, k^- and α (four with modalities, but we fixed $M = 4$). However, as considered in the original article, we set $k^+ = k^-$ which reduces the search to 2 hyper-parameters. We investigated the results of WILDCAT with values of $\alpha \in \{0, 0.1, 0.2, \ldots, 1.0\}$ and $k^+ \in \{1, 2, 3, 5, 10, 20, 40, 60\}$. The best results were obtained for $k^+ = 20$ and $\alpha = 0.2$. We can notice that the mIoU obtained is better than claimed in the original paper with a significantly smaller model (ResNet-18 instead of ResNet-101).

[1] We did not get any answer from the first author after contacting him.

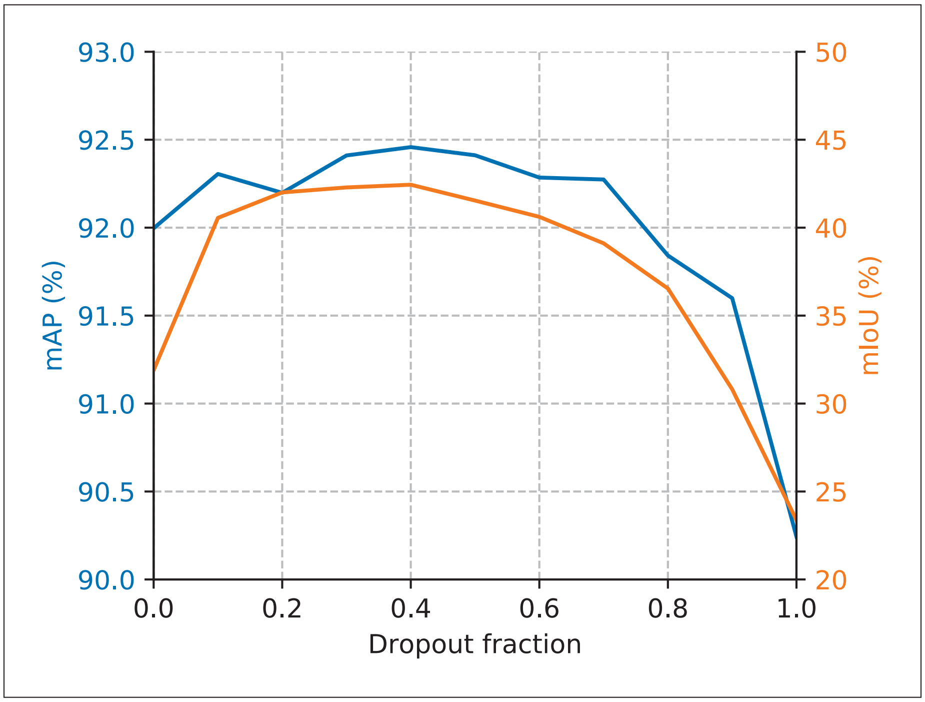


Figure 3.3 Impact of the dropout fraction on the mAP and mIoU on the Pascal VOC 2012 validation set

For our method, we investigate the impact of the dropout fraction before the normalization step. We experiment with values in $\{0, 0.1, 0.2, \ldots, 1.0\}$. Figure 3.3 shows the impact on the mAP and mIoU for the dropout fraction. We can see that a dropout fraction between 0.3 and 0.4 yields the best results. For the rest of the experiment we use a value of 0.4. It is also important to note that a dropout fraction of 1 essentially gives a zero-score to every regions. After the normalization steps, each instance will have a weight of $\frac{1}{HW}$, which makes it equivalent to the Average pooling.

3.4.5 Results

The results of this evaluation are presented in Table 3.2. We obtain a significant improvement in both mIoU and mAP when considering the same architecture compared to existing methods such as WILDCAT and LSE. We also notice that LSE performs in fact well compared to WILDCAT (both claimed and reproduced) when carefully tuned. Not surprisingly, increasing

Table 3.2 Pascal VOC 2012 validation performance (* indicates reported results)

Method	mAP (%)	mIoU (%)
Average pooling (Zhou *et al.*, 2016)	89.1	23.2
Max pooling (Oquab *et al.*, 2015)	91.9	24.6
LSE pooling (Pinheiro & Collobert, 2015b) ($q = 0.5$)	92.5	41.4
WILDCAT (Durand *et al.*, 2017) (ResNet-18)	92.4	40.0
WILDCAT* (Durand *et al.*, 2017) (ResNet-101)	93.4	39.2
Ours (ResNet-18)	92.5	42.4
Ours (ResNet-50)	94.0	43.6
Ours (ResNet-101)	94.4	44.1

model capacity increases performance with a $\sim 2\%$ improvement in both mAP and mIoU when using a ResNet-101 compared to a ResNet-18 at the cost of a five times higher inference time Bianco, Cadene, Celona & Napoletano (2018) with more than four times as many parameters.

3.5 Application to Histology Images and Discussion

After validating the proposed method on Pascal VOC, we test it on histology images. We use the same evaluation scenarios as in Chapter 1 and evaluate on BreakHis and ICIAR to study the impact in term of classification performance and on CAMELYON16 for the localization performance. For all datasets, we do not report results for Max and LSE poolings as their performance is significantly lower than Average, WILDCAT and Deep MIL.

Table 3.3, Table 3.4 and Table 3.5 report the results of our method on BreakHis, BACH and CAMELYON16 datasets for classification and localization performance.

We can observe that our method degrades the classification performance on both BreakHis and BACH which is surprising considering that it performs significantly better on Pascal VOC. One thing that could explain this discrepancy is the number of classes involved in the problem. On the Pascal VOC, images can have multiple objects from different classes in the same image which forces the model to focus on smaller discriminative parts. In our settings, we consider multi-class problems which can only have one class per image. Typically, this leads to predicting the whole image as belonging to the winning class. On camelyon, the difference in classification performance is not as important and we can see that our method somewhat improves the Dice for large images.

Overall, it seems that all studied methods (as well as the proposed one) plateau at the same performance level, being unable to get higher than $\sim 67\%$ of Dice. This suggests that image-level labels might be insufficient as a supervision to learn good segmentation models.

Table 3.3 Accuracy and AP over test folds of BreakHis dataset using different magnification factors and different models

Magnification / **Method**	**40×**		**100×**	
	Accuracy (%)	**AP (%)**	**Accuracy (%)**	**AP (%)**
CAM - Average	92.19 ± 3.54	97.80 ± 2.30	89.64 ± 2.93	98.10 ± 0.91
WILDCAT	92.40 ± 2.82	97.90 ± 2.41	90.22 ± 2.48	97.99 ± 1.55
Deep MIL	91.80 ± 2.70	98.38 ± 1.64	89.54 ± 3.14	97.69 ± 1.16
Ours (d = 0.03)	89.68 ± 4.22	97.00 ± 2.76	89.73 ± 2.70	98.02 ± 0.86

Magnification / **Method**	**200×**		**400×**	
	Accuracy (%)	**AP (%)**	**Accuracy (%)**	**AP (%)**
CAM - Average	91.03 ± 1.33	98.36 ± 0.54	85.09 ± 2.09	96.04 ± 0.99
WILDCAT	90.75 ± 2.00	98.49 ± 0.59	85.85 ± 3.05	96.41 ± 1.40
Deep MIL	91.61 ± 1.34	98.71 ± 0.55	85.98 ± 2.28	96.29 ± 0.85
Ours (d = 0.03)	90.30 ± 2.49	97.88 ± 0.94	85.12 ± 2.53	95.75 ± 1.25

Table 3.4 Accuracy and mAP over test folds on the BACH (Part A) dataset using different models

Method	**Accuracy (%)**	**mAP (%)**
CAM - Average	84.10 ± 2.51	93.23 ± 1.27
WILDCAT	84.80 ± 1.25	93.04 ± 1.00
Deep MIL (adapted)	83.30 ± 3.90	92.68 ± 2.71
Ours (d = 0.03)	79.50 ± 2.37	89.07 ± 1.61

Table 3.5 Accuracy, AP, Dice and mean Dice over metastatic patches obtained with weakly supervised localization models for test folds of CAMELYON16 dataset using growing patch size

Patch size	**512×512**		**768×768**		**1,024×1,024**	
Method	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**	**Accuracy (%)**	**AP (%)**
CAM - Average	98.54 ± 0.36	99.80 ± 0.04	98.37 ± 0.45	99.89 ± 0.03	99.20 ± 0.28	99.92 ± 0.02
WILDCAT	98.37 ± 0.65	99.84 ± 0.05	98.62 ± 0.36	99.92 ± 0.03	99.16 ± 0.17	99.95 ± 0.01
Deep MIL	98.15 ± 0.59	99.82 ± 0.04	98.34 ± 1.16	99.82 ± 0.23	99.17 ± 0.11	99.95 ± 0.01
Ours (d = 0.1)	98.07 ± 0.93	99.76 ± 0.09	98.28 ± 0.75	99.81 ± 0.19	98.67 ± 0.99	99.81 ± 0.32
Method	**Dice index**	**mDice metastatic**	**Dice index**	**mDice metastatic**	**Dice index**	**mDice metastatic**
CAM - Average	65.06 ± 1.00	62.47 ± 1.17	64.16 ± 0.78	61.45 ± 0.67	66.70 ± 2.77	63.98 ± 2.93
WILDCAT	66.06 ± 0.84	62.87 ± 0.86	67.24 ± 1.29	63.74 ± 1.33	66.05 ± 2.96	65.52 ± 1.28
Deep MIL	49.26 ± 2.81	69.76 ± 0.73	47.93 ± 2.66	68.49 ± 2.16	46.61 ± 2.81	65.97 ± 1.67
Ours (d = 0.1)	65.06 ± 0.73	61.97 ± 0.90	64.56 ± 2.15	61.69 ± 0.99	68.14 ± 2.00	64.66 ± 1.60
U-Net	77.68 ± 1.47	70.90 ± 2.37	79.90 ± 1.30	73.03 ± 2.05	80.72 ± 0.83	72.79 ± 1.47

CHAPTER 4

WEAK SUPERVISION USING REGION SIZE INFORMATION

4.1 Introduction

Despite several methods proposed in the literature, obtaining a good segmentation performance solely from image-level label supervision compared to full pixel-level supervision remains an open problem. For natural images, including priors into the model and/or the training phase allowed to dramatically increase the performance of weakly supervised methods. At the time of writing, the best performance for weakly supervised segmentation on Pascal VOC has reached 65% of mIoU (Lee, Kim, Lee, Lee & Yoon, 2019) compared to 44% without any prior and 89% with full pixel-level supervision (Chen, Zhu, Papandreou, Schroff & Adam, 2018a).

Since obtaining full pixel-level annotations on medical images is expensive because of the need for experts compared to natural images, we consider a new scenario with an intermediate level of supervision. Instead of having access to a full ground-truth segmentation mask, we assume that we have access to the size of the different classes present in the image. This annotation is much less expensive to obtain but still requires an expert annotator in the context of medical images. This kind of supervision using size information has been proposed in previous works as inequality constraints either on each sample (Jia *et al.*, 2017) or to include priors about the size of the target object (Kervadec, Dolz, Tang, Granger, Boykov & Ayed, 2019) present in all images. In our case, the regions of interest can have any size and shape so priors on object size proposed by Kervadec *et al.* (2019) are not suitable for our application.

In this chapter we study the impact of adding size supervision for histology images classification and segmentation. More specifically, we proposed to use the label distribution learning framework (Geng, 2016) to constrain the size of the segmented regions to match the true size distribution. We also study the impact of adding noise to the true size distribution provided by annotators to evaluate the robustness of different approaches.

4.2 Related Work

To the best of our knowledge, only Jia *et al.* (2017) considered using the size information in histology images to improve the weakly supervised segmentation performance. In their work, they consider a two-class scenario (*i.e*. normal *vs*. cancerous regions). They combine a classification loss with a constraint on the size if the size of the predicted positive region is larger than the size estimated by an expert (which we will call *true size*). More specifically, the probability of the positive class for the classification is obtained through a generalized mean which approximates the max function:

$$p = P(y = 1|x) = \left(\frac{1}{HW}\sum_{i,j}\sigma(\boldsymbol{M}_{i,j})^r\right)^{1/r} \tag{4.1}$$

where $\boldsymbol{M}_{i,j}$ is the positive class activation map, σ is the sigmoid function and r is a hyper-parameter controlling the sharpness of the approximation. The classification loss l_c for a sample x with binary label $y \in \{0, 1\}$ is a binary cross-entropy:

$$l_c = -(y \log p + (1 - y) \log(1 - p)) \tag{4.2}$$

Given the true size a of the positive region for a sample, the size loss l_s is defined as:

$$l_s = y \max(0, \hat{v} - a)^2 \qquad \text{with} \qquad \hat{v} = \frac{1}{HW}\sum_{i,j}\sigma(\boldsymbol{M}_{i,j}) \tag{4.3}$$

The total loss l is the sum of the two losses: $l = l_c + \lambda_s l_s$ where λ_s is a weight balancing the two terms of the loss.

4.3 Proposed Method

Let us consider a training set $\mathcal{D}$ containing samples x from the input space $\mathcal{X}$, with their associated label y from a set of possible labels $\mathcal{Y}$ and region sizes $\boldsymbol{a} \in [0, 1]^C$ with $\sum \boldsymbol{a} = 1$. In the case of multi-label classification, y is replaced the one-hot vector $y \in \{0, 1\}^C$. In this work we consider a CNN with stride S that produces CAMs $\mathbf{M} \in \mathbb{R}^{C \times H \times W}$ for an input x, meaning the last layer is a classification layer. Typically, this is a 1 × 1 convolution. The produced CAMs are then pooled using a spatial pooling function to produce image-level scores $\boldsymbol{s} \in \mathbb{R}^C$ which can be transformed into image-level probabilities $\boldsymbol{p} \in [0, 1]^C$ using a softmax function for multi-class or sigmoid function for multi-label classification scenarios. For the following study, we use an averaging pooling such that: $s_c = \frac{1}{HW} \sum_{i,j} \mathbf{M}_{c,i,j}$ Our goal is to obtain both good classification and segmentation performance in a single training procedure using image-level labels and size information.

For the classification task, we minimize the cross-entropy loss l_{mc} for the multi-class scenario:

$$l_{mc} = -\sum_{c}^{C} \boldsymbol{I}(c = y) \log \boldsymbol{p}_c \tag{4.4}$$

where $\boldsymbol{I}$ is the indicator function. For the case of multi-label classification, we minimize the binary cross-entropy loss l_{ml} over all the classes:

$$l_{ml} = -\sum_{c}^{C} \boldsymbol{y}_c \log \boldsymbol{p}_c + (1 - \boldsymbol{y}_c) \log(1 - \boldsymbol{p}_c) \tag{4.5}$$

In the following equations, we denote by l_c the classification loss independently of the scenario.

One of the main problems of the methods studied in Chapter 1 is their high false positive rate. To reduce that, we consider using the label distribution framework to constrain the spatial size of the regions predicted by the networks. This constraint is applied on the CAMs directly after being normalized, either by a softmax or a sigmoid function depending on the scenario. Before computing the size of the regions predicted by the network, we also upsample the CAMs to match the input size. We denote the upsampled and normalized CAMs $\hat{\mathbf{M}} \in [0, 1]^{C \times H^{in} \times W^{in}}$. To obtain a segmentation of the input, we apply an $\arg\max$ function for both multi-class and multi-label classification as one pixel typically belongs to only one class. From there, obtaining the relative size of the predicted regions for each class $\boldsymbol{v} \in [0,1]^C$ is straightforward:

$$\boldsymbol{v}_c = \frac{1}{H^{in}W^{in}} \sum_{h,w}^{H^{in},W^{in}} \boldsymbol{I}(\arg\max_i \hat{\mathbf{M}}_{i,h,w} = c) = \frac{\left\|\mathbf{I}(\arg\max_i \hat{\mathbf{M}}_{i,h,w} = c)\right\|_1}{H^{in}W^{in}} \tag{4.6}$$

Where $\mathbf{I}$ is the indicator function. However, the arg max function is not differentiable, meaning that we cannot use the predicted size for training. Therefore, we instead of computing the L_1 norm on the binary predictions, we compute the L_1 on the probabilities to obtain an estimate of the predicted relative size of each class within an image:

$$\hat{\boldsymbol{v}}_c = \frac{1}{H^{in}W^{in}} \sum_{h,w}^{H^{in},W^{in}} \hat{\mathbf{M}}_{c,h,w} = \frac{\left\|\hat{\mathbf{M}}_{c,h,w}\right\|_1}{H^{in}W^{in}} \tag{4.7}$$

Since we have access to the size of the regions corresponding to the different classes of our problem, we add a loss penalizing the difference between the estimated predicted sizes $\hat{\boldsymbol{v}}$ and the true sizes $\boldsymbol{a}$. The choice of the penalty is arbitrary: in this work we consider using the Kullback-Leibler divergence, the L_2 distance or an adaptive variant of the L_2 loss proposed by Barron (2019).

Since we estimate our size from the normalized activation maps, we still have $\hat{\boldsymbol{v}} \in [0,1]^C$ and $\sum \hat{\boldsymbol{v}} = 1$, meaning we can use the Kullback-Leibler (KL) divergence, leading to size penalty loss l_s:

$$l_s = \sum_c^C \boldsymbol{a}_c \log\left(\frac{\boldsymbol{a}_c}{\hat{\boldsymbol{v}}_c}\right) \tag{4.8}$$

In the case of L_2 loss, we obtain:

$$l_s = \sum_c^C (\hat{\boldsymbol{v}}_c - \boldsymbol{a}_c)^2 \tag{4.9}$$

However, both of these losses are not well suited to outliers. Indeed, the cost for being far from the ground-truth prediction grows linearly for the L_2 loss and is unbounded for the KL divergence. This is especially important if we consider that we only have an estimation of the true sizes $\boldsymbol{a}$, meaning that there are outliers. To alleviate this problem, many loss functions have been proposed to be robust to outliers such as the Charbonnier loss (Charbonnier, Blanc-Feraud, Aubert & Barlaud, 1994) (also called the pseudo-Huber loss (Huber, 1964) or the L1-L2 loss (Zhang, 1995)), the Cauchy loss (Black & Anandan, 1996), the Geman-McClure loss (Ganan & McClure, 1985) or the Welsch loss (Dennis Jr & Welsch, 1978). Barron (2019) noticed that all these losses can be generalized by a single loss $\rho(\cdot, \alpha, \gamma)$ parametrized by a shape parameter $\alpha \in \mathbb{R}$ and scale parameter $\gamma \in \mathbb{R}_+^*$:

$$\rho(x, \alpha, \gamma) = \begin{cases} \frac{1}{2}\left({}^{x}\!/\!_{\gamma}\right)^2 & \text{if } \alpha = 2 \\ \log\left(\frac{1}{2}\left({}^{x}\!/\!_{\gamma}\right)^2 + 1\right) & \text{if } \alpha = 0 \\ 1 - \exp\left(-\frac{1}{2}\left({}^{x}\!/\!_{\gamma}\right)^2\right) & \text{if } \alpha = -\infty \\ \frac{|\alpha-2|}{\alpha}\left(\left(\frac{(x/\gamma)^2}{|\alpha-2|} + 1\right)^{\alpha/2} - 1\right) & \text{otherwise} \end{cases} \tag{4.10}$$

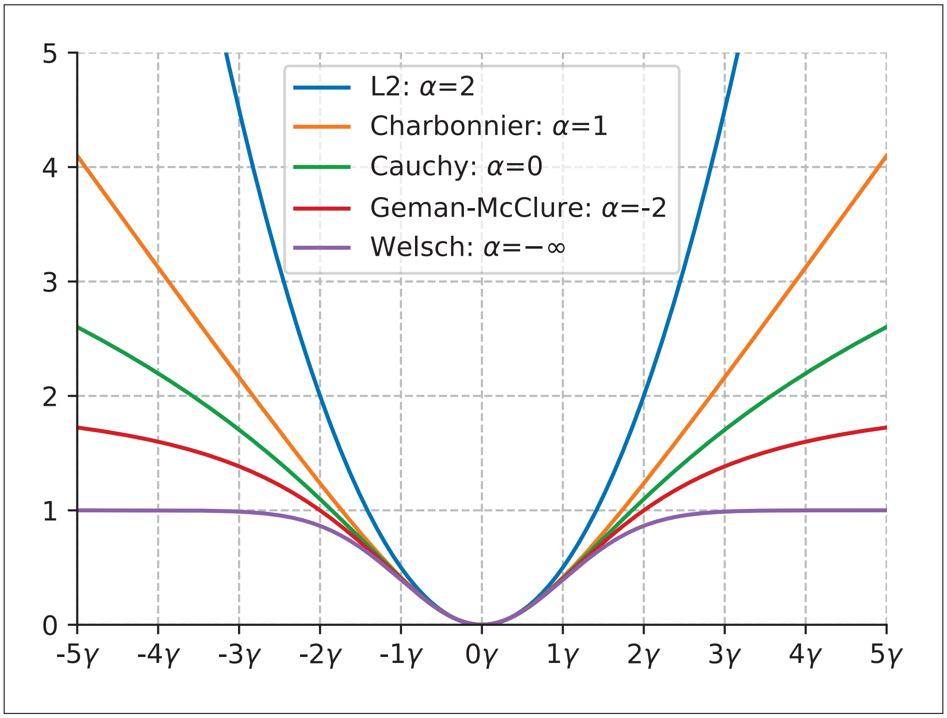


Figure 4.1 General loss function proposed by Barron (2019)

This loss shown in Figure 4.1 is C^∞ with respect to x, α and $\gamma > 0$ which makes it suitable for gradient-based optimization. Instead of fixing α and γ based on heuristics or trial-and-error (*i.e.* grid-search), Barron (2019) proposes to learn α and γ along the model parameters. However, setting α as a free parameter while minimizing ρ does not work since ρ is monotonous w.r.t. α so alpha will become as small as possible. Instead, we need to minimize the negative log-likelihood (NLL) of the probability distribution that corresponds to the loss defined by:

$$\begin{aligned} p(x|\alpha, \gamma) &= \frac{1}{\gamma Z(\alpha)} \exp(-\rho(x, \alpha, \gamma)) \\ Z(\alpha) &= \int_{-\infty}^{\infty} \exp(-\rho(x, \alpha, 1)) \end{aligned} \tag{4.11}$$

$Z(\alpha)$ is a normalization term required to have $\int_{-\infty}^{\infty} p(x|\alpha, \gamma) = 1$. In this case, minimizing the NLL makes it possible to learn α. If α decreases, it gives less weight to outliers, but at the cost of an increased loss for the inliers. Note that $p(x|\alpha, \gamma)$ is only defined for $\alpha \geq 0$ since $Z(\alpha)$ is divergent for $\alpha < 0$. Therefore, we can only learn this loss function for positive values of α. This limitation is balanced by the fact that γ is learned as well.

Our equality constraint on the sizes of the predicted regions suppose that the estimation of the sizes is close to the predicted sizes, meaning that the probabilities are either close to 0 or 1. To encourage that property, we add an unsupervised entropy loss:

$$l_e = -\sum_{c}^{C} \frac{1}{H^{in} W^{in}} \sum_{h,w}^{H^{in}, W^{in}} \hat{\mathbf{M}}_{c,h,w} \log \hat{\mathbf{M}}_{c,h,w} \tag{4.12}$$

Our total training l_t is expressed as:

$$l_t = \frac{l_c + \lambda_s l_s + \lambda_e l_e}{1 + \lambda_s + \lambda_e} \tag{4.13}$$

where λ_s and λ_e are hyper-parameters controlling the importance of each term. Note that the loss is normalized to avoid having to tune the learning rate for each combination of hyper-parameters.

4.4 Experiments

4.4.1 Dataset and Evaluation

We evaluate our method on the GlaS and CAMELYON16 dasasets presented in Chapter 1. On both of these datasets, we evaluate the performance of our algorithm in terms of accuracy for the classification performance and Dice over all the pixels of the test set for the segmentation performance.

For GlaS, we consider three different classes (*i.e.* background, benign gland and malignant gland) for the segmentation task. To obtain a single metric, we compute the Dice index over all the pixels of the test set for the two foreground classes (*i.e.* benign gland and malignant gland)in one. This means that we have:

$$Dice_{GlaS} = \frac{2 * (TP_1 + TP_2)}{2 * (TP_1 + TP_2) + FP_1 + FP_2 + FN_1 + FN_2} \tag{4.14}$$

This is different from averaging the Dice of the benign gland and malignant together since they have different cardinalities.

4.4.2 Training and Inference

As a feature extractor, we use the same model as in Chapter 1, *i.e.* a ResNet-18 He *et al.* (2016) pretrained on ImageNet-1000. The stride of that CNN is by default 32 but since we have access to more supervision than for the experiments where we only use image-level supervision, we also experiment with a total stride of 16. This is achieved by changing the stride of the first convolution in the last block of the ResNet. In a convolution changing the stride does not change the shape of the underlying weight (and bias if applicable) tensor meaning that we can still use the pretrained weights.

We use the same augmentation strategies as in Chapter 1. For GlaS, these correspond to:

- random cropping: we extract a patch of size 448 × 448 at a random location,
- random color jittering with parameters: brightness, contrast and saturation at 0.5 and hue at 0.05 (from the PyTorch framework),
- random horizontal flipping,
- random discrete rotation: the image is rotated randomly with an angle in 0, 90, 180, 270 degrees.

For CAMELYON16, we remove the random cropping and train with the full-size images. The color jittering, random flipping and rotation are kept.

The models are trained by minimizing the loss l_t for 160 epochs for GlaS and 20 epochs for CAMELYON16 using SGD with Nesterov momentum of 0.9 and a weight decay of 0.0001. Instead of using a step decay for the learning rate, we use a cosine annealing policy (see Figure-A II-1) which has been found to increase classification performance (He, Zhang, Zhang, Zhang, Xie & Li, 2019) with a starting learning rate η_0 of 0.01.

For inference, we simply feed the entire images to the model.

4.4.3 Size annotation

For both datasets (*i.e.* GlaS and CAMELYON16), we have access to the ground truth segmentation. This allows us to compute the true size of the positive region for each image. However, in a realistic scenario, estimating the size of the positive region introduce noise. Therefore, we investigate the impact of adding noise to the true size vectors $\boldsymbol{a}$. For both datasets, the provided masks are binary. For GlaS, which contains three classes (*i.e.* background, benign glands and malignant glands), this means that the size of the positive region corresponds to the size of the glands corresponding to the image-level label. Therefore, one component of the size vector for GlaS is always zero as an image is always either benign of malignant (the background is always present). We model the noise in the annotation with a normal distribution $\mathcal{N}(0, \sigma_s^2)$. For each training sample, we sample a noise $z \sim \mathcal{N}(0, \sigma_s^2)$ and add it to the size of the positive region $\hat{\boldsymbol{a}}_y = \boldsymbol{a}_y + z$, clip it to $[0, 1]$ and set the size of the background region to $\hat{\boldsymbol{a}}_0 = 1 - \hat{\boldsymbol{a}}_y$. The clipping ensure that $\sum \hat{\boldsymbol{a}} = 1$ for any noise z.

Since GlaS has a very small number of samples (80 for training) which do not all have the same size, we compute the size for each sample after performing the random cropping and add noise to that size. This means that we expect the model performance when adding noise to stay close to the clean performance. For CAMELYON16, we set a more realistic scenario where we compute the size for each image once and perturb it by a fixed amount throughout the whole training, preventing that the model learns the true size as the mean of all the sizes of that image.

We investigate the following values for $\sigma_s \in \{0.001, 0.01, 0.03, 0.1\}$ on GlaS and limit to $\sigma_s \in \{0.01, 0.1, 0.3\}$ for CAMELYON16 as one training takes 3 hours for one, two or four Nvidia V100 for sizes of 512×512, 768×768 and 1024×1024 respectively.

4.4.4 Hyper-parameters

As mentioned in Section 4.3, we investigate the use of three different losses:

- KL divergence,
- L2 distance,
- Adaptive loss (Barron, 2019).

Since these losses have different slopes, we perform a grid-search on the values of λ_s and λ_s with values $\lambda_s \in \{0, 0.1, 0.3, 1, 3, 10, 30, 100\}$ and $\lambda_e \in \{0, 0.01, 0.1, 1, 10\}$. Because of the small size of GlaS, a full training takes less than three minutes on a Nvidia V100 GPU compared to several hours for the CAMELYON16 dataset. Therefore, we perform the hyper-parameters tuning solely on the GlaS dataset, which represent $3,000$ trainings, and use the optimal values found for the CAMELYON16 dataset.

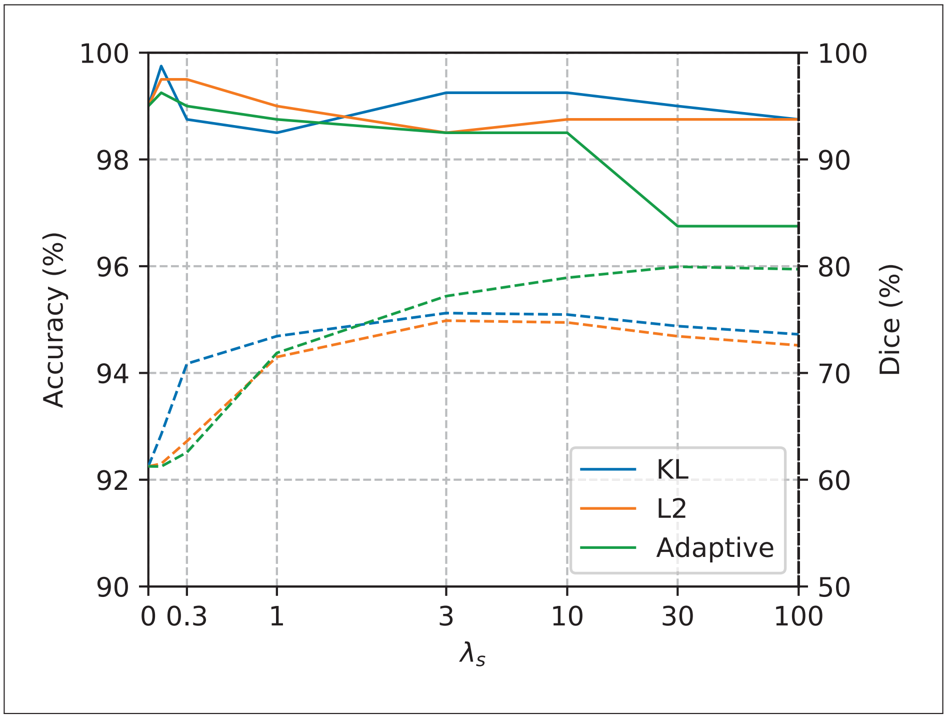


Figure 4.2 Accuracy and Dice for different values of λ_s ($\lambda_e = 0$ and $\sigma_s = 0$)

4.4.5 Results and Discussion

For the three loss types, we found that there is a trade-off between accuracy in classification and Dice index for segmentation. For larger values of λ_s, the Dice index increases up to a maximum at the cost of classification accuracy. Figure 4.2 shows the evolution of the accuracy and Dice for the three different losses and increasing values of λ_s while $\lambda_e = 0$ and $\sigma_s = 0$. We observe that the Dice benefits from a higher λ_s for all losses, which is not surprising. For too large values of $\lambda_s > 10$ however, the Dice starts dropping for KL and L2 losses. For the adaptive loss, the Dice stabilizes at 80% at the cost of a significant drop in accuracy. From this graph, we can safely conclude that a value of $\lambda_s = 10$ is a good starting point as it offers a satisfying trade-off between accuracy and Dice.

With $\lambda_s = 10$, we find the best value for λ_e is 0.1. Too high a value of λ_e leads to poor performance in both classification and segmentation. For KL and L2 losses, a value of λ_e leads to better Dices at the cost of much lower classification accuracy. Full tables for $\sigma_s = 0$ can be found in Chapter III with values of $\lambda_s \in \{0.3, 1, 3, 10\}$ and $\lambda_e \in \{0.01, 0.1, 1\}$. Several visual examples of the obtained segmentation using our method can be found in Chapter IV.

Table 4.1 Accuracy and Dice index over the test folds of GlaS dataset with $\sigma_s = 0$

Method	Supervision	Accuracy (%)	Dice (%)
CAM - Average	Image label	99.75 ± 0.56	68.43 ± 0.73
Deep MIL (adapted)	Image label	99.25 ± 1.12	72.13 ± 1.78
U-Net	Pixel label	-	90.54 ± 0.88
Jia *et al.*	Image label + Size	99.25 ± 0.68	72.58 ± 1.74
Ours - KL		99.00 ± 0.56	75.72 ± 1.01
Ours - L2		98.75 ± 0.88	75.04 ± 0.82
Ours - Adaptive		98.25 ± 1.43	79.07 ± 2.11

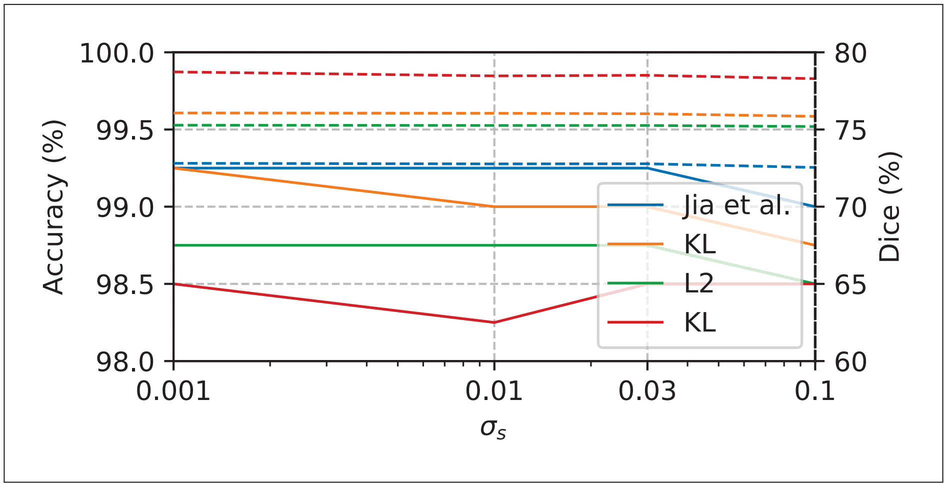


Figure 4.3 Accuracy and Dice for varying values of σ_s on GlaS

Table 4.1 reports the results for the different methods. We notice that the method proposed by Jia *et al.* (2017) does not improve significantly the results of Deep MIL which only uses labels as supervision. KL and L2 losses offer a certain level of improvement, but do not come close to the adaptive loss. This result is not surprising since the adaptive loss is a generalization of the L2 and other losses so it should at least perform as well as the L2 loss. The adaptive loss offers a significant improvement over all other methods which suggests that being robust to outliers is important for this application. Figure 4.3 shows that the impact of noise added to the true size is negligible. This is expected as mentioned in Subsection 4.4.3 since we are generating noise for each patch randomly cropped.

Table 4.2 Impact of noise added to the size for the CAMELYON16 dataset for our proposed method with the adaptive loss

Patch size	**512×512**		**768×768**		**1,024×1,024**	
σ_s	**Accuracy (%)**	**Dice (%)**	**Accuracy (%)**	**Dice (%)**	**Accuracy (%)**	**Dice (%)**
0	96.95 ± 1.74	73.47 ± 1.74	96.17 ± 1.29	75.74 ± 2.81	96.25 ± 0.66	79.74 ± 1.41
0.01	96.20 ± 2.73	70.90 ± 3.69	96.98 ± 1.09	75.85 ± 2.50	96.57 ± 1.34	79.93 ± 1.38
0.1	97.38 ± 1.09	71.48 ± 1.65	96.53 ± 0.72	73.85 ± 1.53	97.03 ± 0.67	78.66 ± 0.78
0.3	98.01 ± 0.49	62.88 ± 1.45	98.23 ± 0.33	64.52 ± 1.03	98.40 ± 0.57	67.44 ± 1.67
U-Net	–	77.68 ± 1.47	–	79.90 ± 1.30	–	80.72 ± 0.83

For CAMELYON16, we only trained using the adaptive loss since it presented much more promising results. We set $\lambda_s = 3$ and $\lambda_e = 0.1$. Table 4.2 reports the results for the three different sizes considered for CAMELYON16 with various levels of noise. While the classification accuracy drops a few percent compared to Table 2.8, the Dice significantly improves for larger images. The Dice even reaches levels comparable to the one achieved by a U-Net trained in a fully-supervised fashion with images of size 1024×1024. Adding noise to the true size (modeling the error an annotator would make) only impacts significantly the results when $\sigma_s > 0.1$. Interestingly, for sizes 768×768 and 1024×1024, adding a small amount of noise does improve the Dice, suggesting a regularizing effect from the noise. For $\sigma_s = 0.3$, the performance strongly degrades in term of Dice but increases in term of classification accuracy. This further confirms a trade-off between the two objectives the we aim to tackle.

Overall, these results on GlaS and CAMELYON16 suggest that our method could be used in a real-world scenario where an annotator would only have to estimate the true size. The advantage of this type of annotation is clear but requires some evaluation on the errors made by annotators in a real-world scenario.

CONCLUSION AND RECOMMENDATIONS

In this thesis, we studied different aspects of the problem of WSL for histological images.

First, we analyzed several existing methods proposed for natural images using only image-level labels. We evaluated their performance on three existing datasets: BreakHis, BACH and GlaS and created a modified version of CAMELYON16 to fit our evaluation requirements. The studied approaches showed not to be very effective on histological images in term of segmentation performance as they tend to have high false positive rates, predicting almost the whole image as a region of interest. This analysis allowed us to conclude that histological images indeed have different characteristics that needs to be taken into account to train models.

In a second contribution, we improved an existing method originally designed for binary classification to adapt it to the multi-class and multi-label scenarios. The method we proposed indeed improved the performance compared to similar methods on natural images on the competitive Pascal VOC 2012 dataset. However, this improvement did not transfer well to the histological images as it degraded classification performance while not improving segmentation performance significantly. This further confirmed the difference in characteristics of the two problems, and the need to use more supervision for accurate segmentation.

In a third contribution, we studied the impact of adding more supervision. This supplementary supervision came into the form of a size annotation that we include using an equality constraint as a loss term. We evaluated our proposed method on GlaS and CAMELYON16 and saw a large improvement in segmentation performance without significantly degrading classification performance. We also studied the impact of adding noise to the size information on the performance to conclude on the level of accuracy required in the size estimation.

Future Work

This work on histology suggests the following directions for future work:

- **Using the class structure as a prior**: in typical computer vision problems such as classification and segmentation in a set of given classes, there is no structure between the different classes. The consequence is that predicting a dog or a plane instead of a car is penalized the same way. However, in histological images, predicting a benign cancer state for an invasive cancer should be more penalized than predicting an insitu cancer. This strcture between classes is studied under the field of ordinal regression and could be beneficial for this application.

- **Including domain expert knowledge**: In the same way that priors were included in WSL for segmentation of natural images (edges define objects), it would be interesting to define and include in the model or the training algorithm priors about histological images. Such priors can typically reduce the need for stronger annotations.

- **Going further than binary segmentation**: Currently, all datasets that provide segmentation masks consider only two classes: benign and malignant. Some classification datasets consider more classes and thus, are usually more difficult. Therefore, collecting a segmented dataset with more than two classes could help with the research in this application.

APPENDIX I

SUPPLEMENTARY MATERIAL FOR THE PAPER TITLED DEEP WEAKLY-SUPERVISED LEARNING METHODS FOR CLASSIFICATION AND LOCALIZATION IN HISTOLOGY IMAGES: A SURVEY

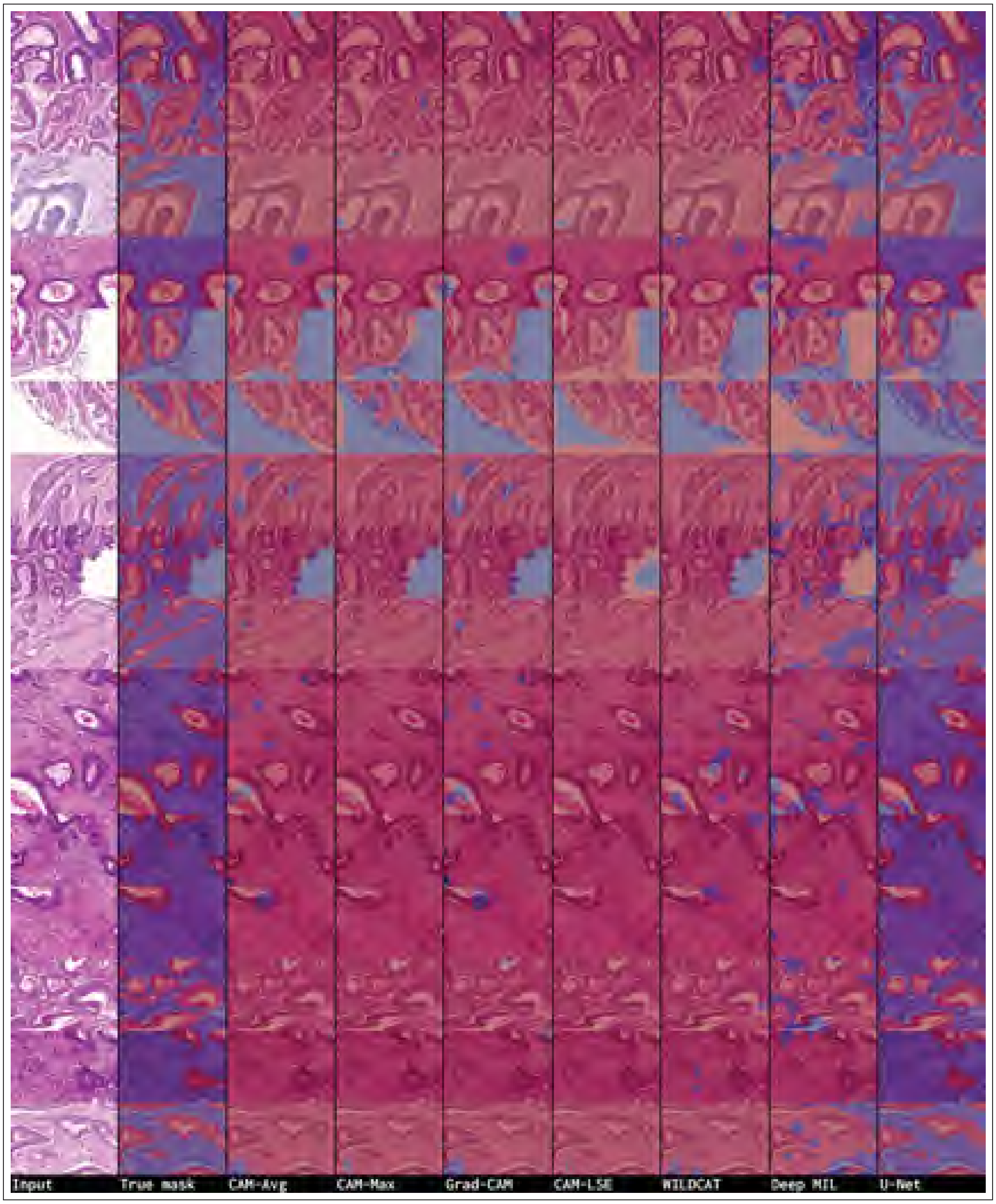


Figure-A I-1 Examples of visual comparison of the predicted binary mask of each WSOL method over GlaS test set (first split) for the malignant class. (Best visualized in color.)

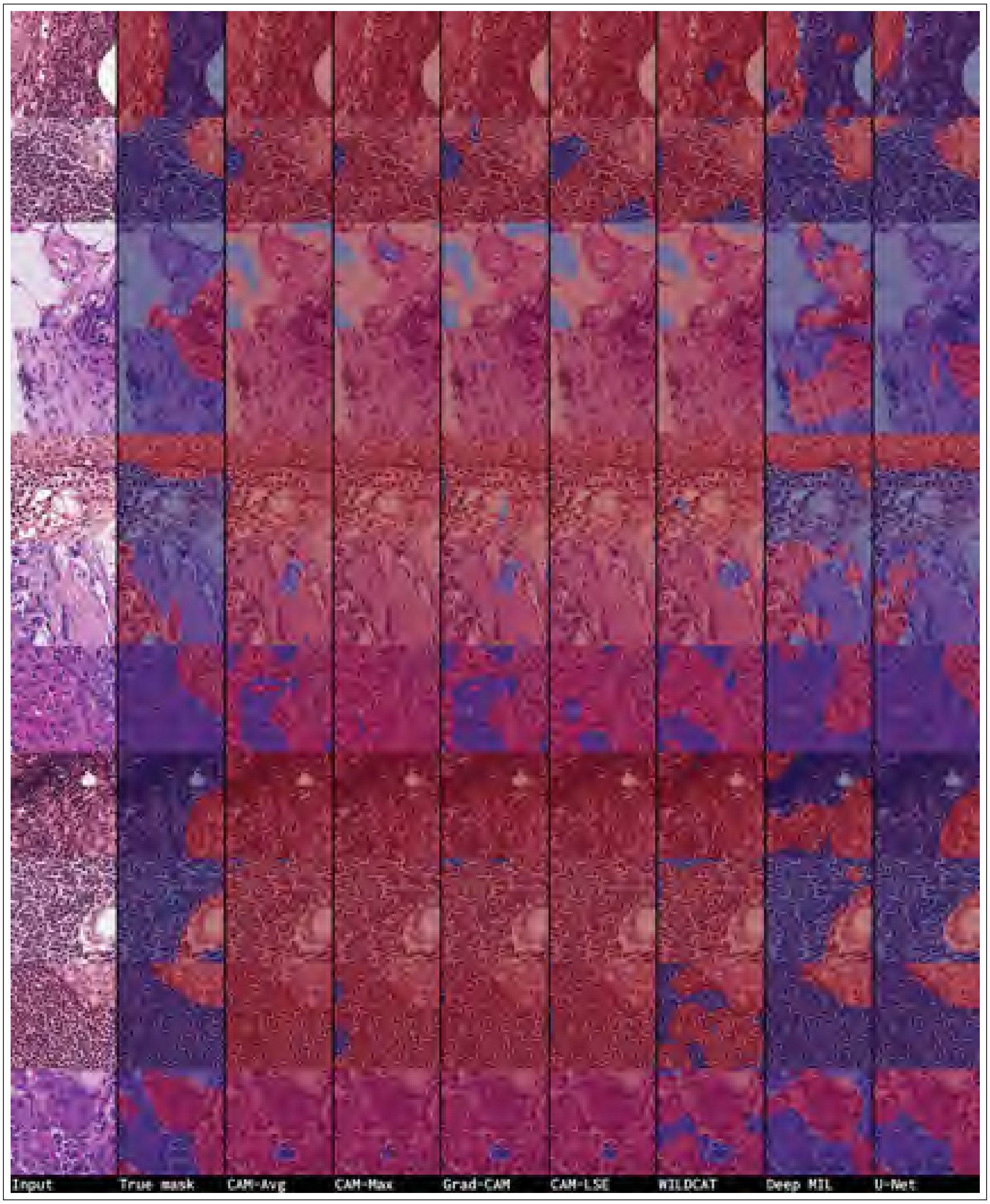


Figure-A I-2 Examples of visual comparison of the predicted binary mask of each WSOL method over CAMELYON16 test set (first split) for the metastatic class (patch size (512, 512)). (Best visualized in color.)

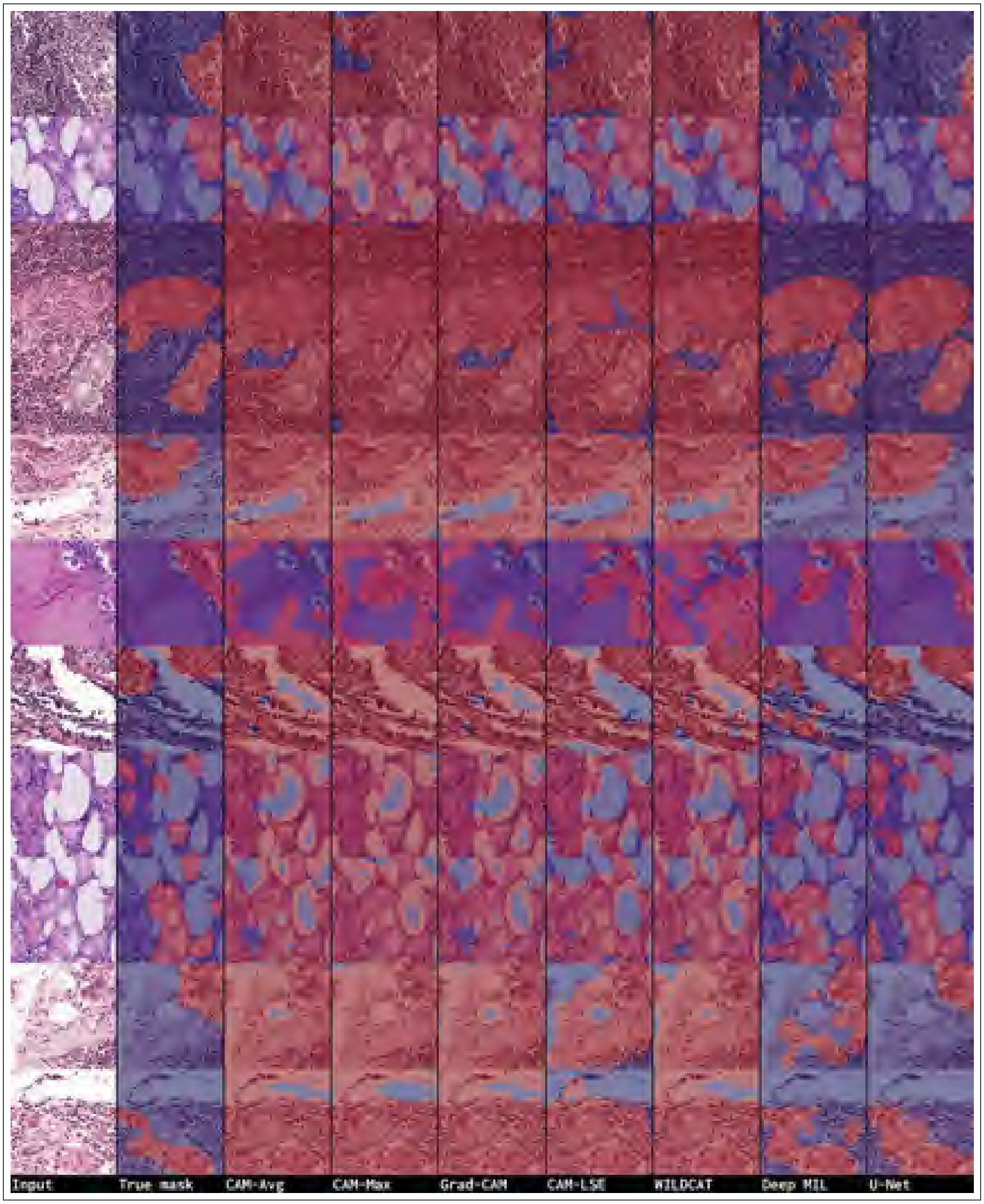


Figure-A I-3 Examples of visual comparison of the predicted binary mask of each WSOL method over CAMELYON16 test set (first split) for the metastatic class (patch size (768, 768)). (Best visualized in color.)

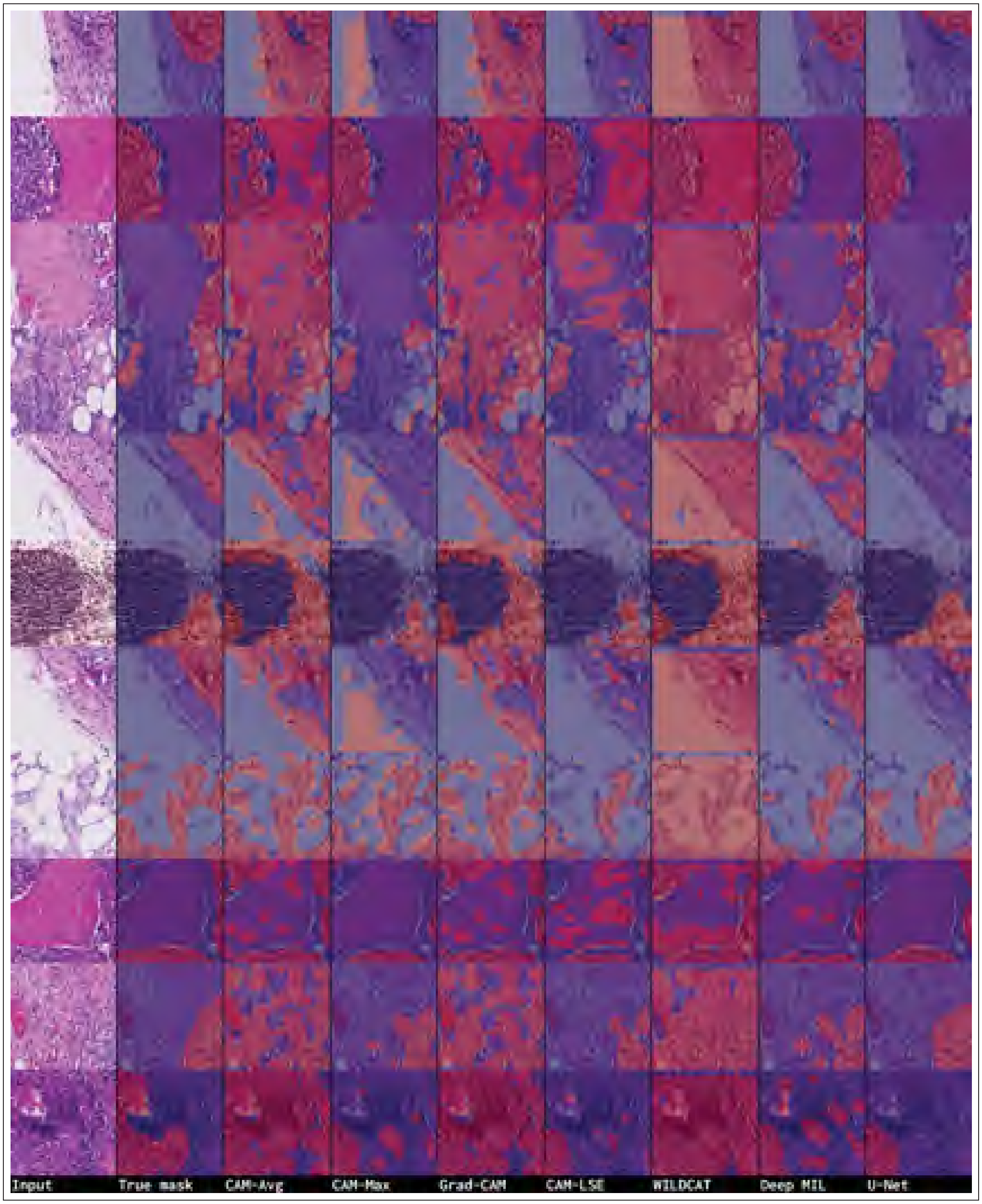


Figure-A I-4 Examples of visual comparison of the predicted binary mask of each WSOL method over CAMELYON16 test set (first split) for the metastatic class (patch size (1,024, 1,024)). (Best visualized in color.)

APPENDIX II

COSINE LEARNING RATE POLICY

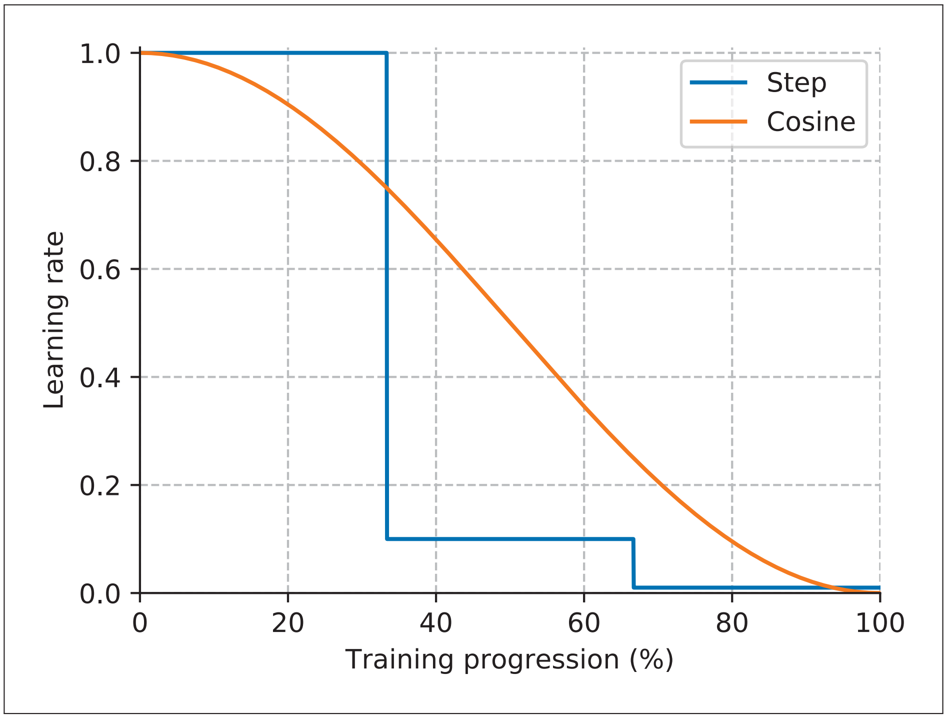


Figure-A II-1 Cosine learning rate policy with a starting learning rate of $\eta = 1$

APPENDIX III

GRID-SEARCH ON THE HYPER-PARAMETERS FOR THE SIZE SUPERVISION

Table-A III-1 Accuracy and Dice for different values of λ_s and λ_e

Loss Type	λ_s	λ_e	**Accuracy (%)**	**Dice (%)**
KL	0.3	0.01	99.00 ± 1.05	70.15 ± 2.00
		0.1	99.50 ± 0.68	69.25 ± 1.42
		1	96.25 ± 1.77	62.26 ± 1.28
	1	0.01	98.75 ± 0.88	73.54 ± 1.53
		0.1	99.00 ± 0.56	73.56 ± 1.30
		1	97.75 ± 1.63	65.33 ± 0.83
	3	0.01	99.25 ± 0.68	75.65 ± 0.66
		0.1	99.25 ± 0.68	75.80 ± 0.57
		1	97.00 ± 0.68	75.67 ± 1.38
	10	0.01	99.25 ± 0.68	75.49 ± 1.03
		0.1	99.00 ± 0.56	75.72 ± 1.01
		1	98.00 ± 0.68	76.96 ± 0.83
L2	0.3	0.01	99.50 ± 0.68	63.33 ± 0.85
		0.1	98.75 ± 1.53	61.71 ± 1.32
		1	97.75 ± 1.05	61.68 ± 1.06
	1	0.01	99.00 ± 0.56	71.20 ± 2.38
		0.1	99.00 ± 0.56	68.60 ± 3.28
		1	97.00 ± 1.12	61.80 ± 0.83
	3	0.01	98.50 ± 0.56	74.90 ± 0.80
		0.1	98.50 ± 0.56	75.00 ± 0.97
		1	97.25 ± 1.05	68.69 ± 1.92
	10	0.01	98.75 ± 0.88	74.78 ± 0.83
		0.1	98.75 ± 0.88	75.04 ± 0.82
		1	97.25 ± 1.05	76.09 ± 1.33
Adaptive	0.3	0.01	98.75 ± 1.53	62.44 ± 1.97
		0.1	98.00 ± 1.90	61.28 ± 1.36
		1	97.25 ± 1.63	61.57 ± 1.33
	1	0.01	98.50 ± 1.05	71.49 ± 2.58
		0.1	99.00 ± 1.05	69.51 ± 2.02
		1	97.25 ± 1.37	61.48 ± 0.83
	3	0.01	99.25 ± 0.68	76.87 ± 1.11
		0.1	98.25 ± 0.68	76.94 ± 0.79
		1	97.75 ± 1.05	70.22 ± 3.08
	10	0.01	98.00 ± 0.68	79.67 ± 1.20
		0.1	98.25 ± 1.43	79.07 ± 2.11
		1	97.50 ± 1.98	78.37 ± 1.81

APPENDIX IV

EXAMPLE OF SEGMENTATIONS FOR THE SIZE SUPERVISION

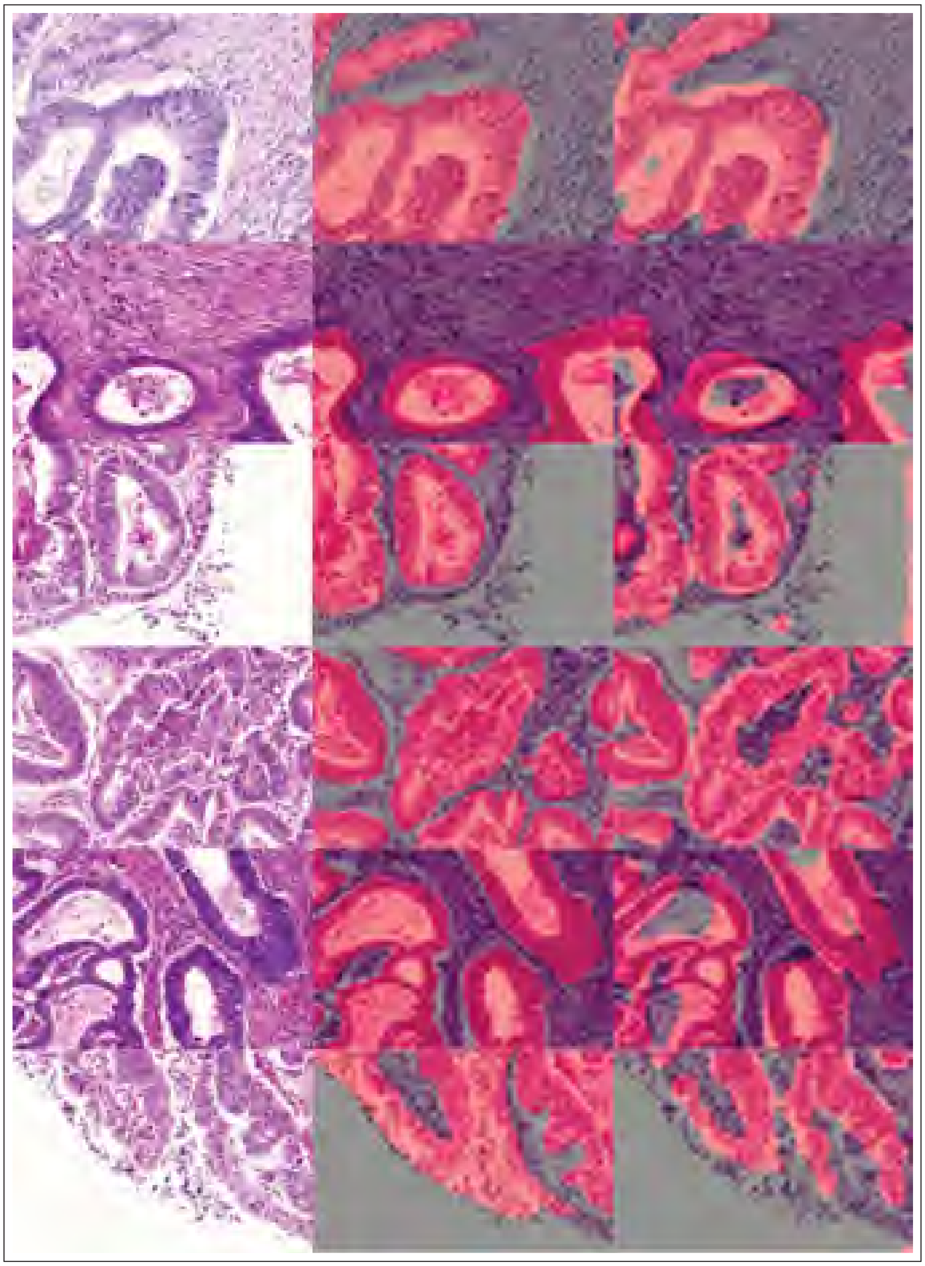


Figure-A IV-1 Examples of images (left) with the ground truth binary mask (center) and the predicted binary masks (right) for our proposed method with size supervision on the GlaS test set

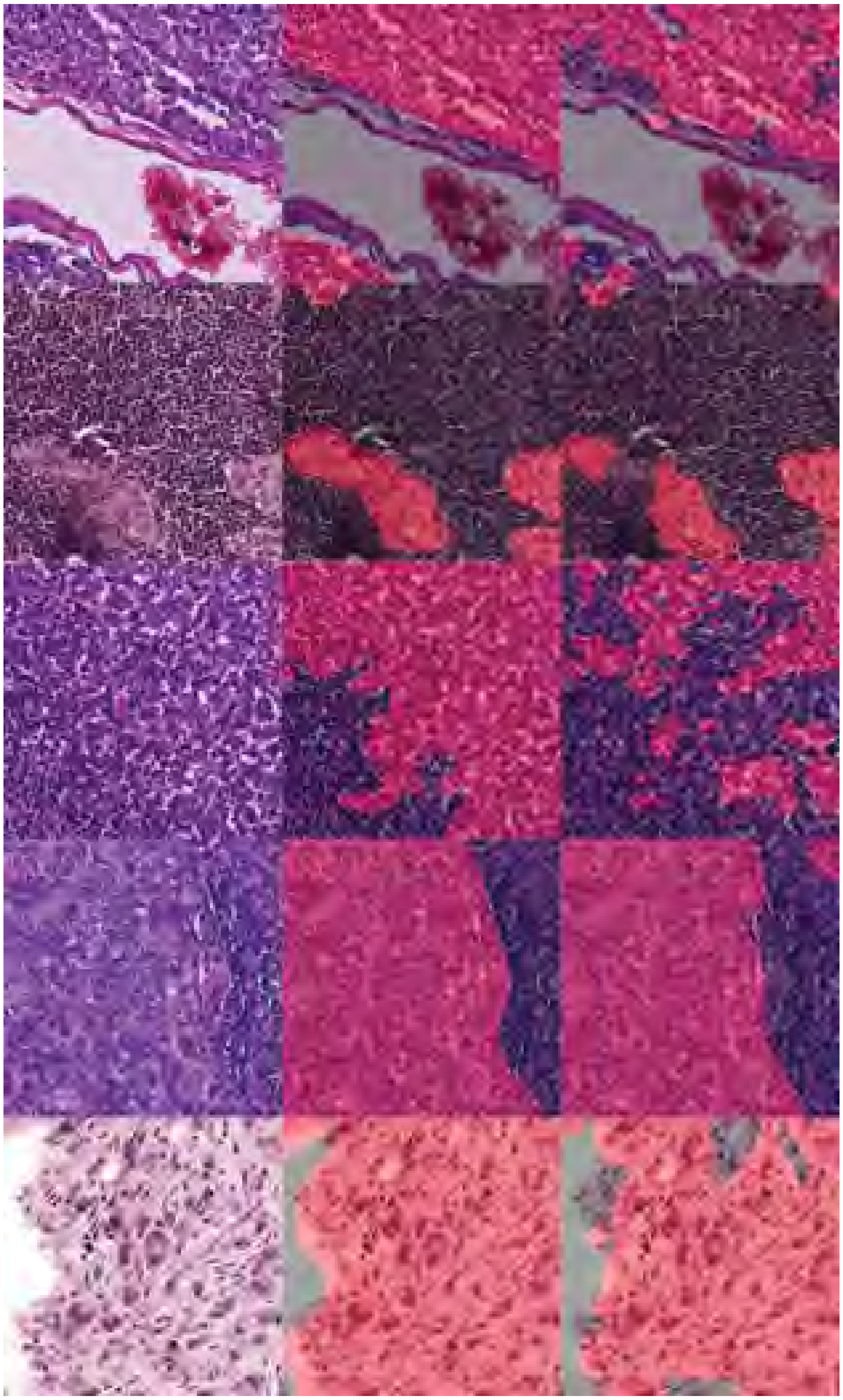


Figure-A IV-2 Examples of images (left) with the ground truth binary mask (center) and the predicted binary masks (right) for our proposed method with size supervision on the CAMELYON16 test set from the 1024 × 1024 size

APPENDIX V

DECOUPLING DIRECTION AND NORM FOR EFFICIENT GRADIENT-BASED L_2 ADVERSARIAL ATTACKS AND DEFENSES

Jérôme Rony* [1], Luiz G. Hafemann* [1], Luiz S. Oliveira[2], Ismail Ben Ayed[1], Robert Sabourin[1], Eric Granger[1]

* Equal contribution
[1] Department of Automated Manufacturing Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

[2] Department of Informatics, Federal University of Parana (UFPR),
Curitiba, PR, Brazil

Article accepted to the « IEEE Computer Socitey Conference on Computer Vision and Pattern Recognition (CVPR) » 2019.

Abstract

Research on adversarial examples in computer vision tasks has shown that small, often imperceptible changes to an image can induce misclassification, which has security implications for a wide range of image processing systems. Considering L_2 norm distortions, the Carlini and Wagner attack is presently the most effective white-box attack in the literature. However, this method is slow since it performs a line-search for one of the optimization terms, and often requires thousands of iterations. In this paper, an efficient approach is proposed to generate gradient-based attacks that induce misclassifications with low L_2 norm, by decoupling the direction and the norm of the adversarial perturbation that is added to the image. Experiments conducted on the MNIST, CIFAR-10 and ImageNet datasets indicate that our attack achieves comparable results to the state-of-the-art (in terms of L_2 norm) with considerably fewer iterations (as few as 100 iterations), which opens the possibility of using these attacks for adversarial training. Models trained with our attack achieve state-of-the-art robustness against white-box gradient-based L_2 attacks on the MNIST and CIFAR-10 datasets, outperforming the Madry defense when the attacks are limited to a maximum norm.

2. Introduction

Deep neural networks have achieved state-of-the-art performances on a wide variety of computer vision applications, such as image classification, object detection, tracking, and activity recognition (Gu, Wang, Kuen, Ma, Shahroudy, Shuai, Liu, Wang, Wang, Cai & Chen (2018)). In spite of their success in addressing these challenging tasks, they are vulnerable to active *adversaries*. Most notably, they are susceptible to *adversarial examples*[1], in which adding small perturbations to an image, often imperceptible to a human observer, causes a misclassification (Biggio & Roli (2018); Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow & Fergus (2014)).

Recent research on adversarial examples developed *attacks* that allow for evaluating the robustness of models, as well as *defenses* against these attacks. Attacks have been proposed to achieve different objectives, such as minimizing the amount of noise that induces misclassification (Carlini & Wagner (2017); Szegedy *et al.* (2014)), or being fast enough to be incorporated into the training procedure (Goodfellow, Shlens & Szegedy (2015); Tramèr, Kurakin, Papernot, Boneh & McDaniel (2018)). In particular, considering the case of obtaining adversarial examples with lowest perturbation (measured by its L_2 norm), the state-of-the-art attack has been proposed by Carlini and Wagner (C&W) (Carlini & Wagner (2017)). While this attack generates adversarial examples with low L_2 noise, it also requires a high number of iterations, which makes it impractical for training a robust model to defend against such attacks. In contrast, one-step attacks are fast to generate, but using them for training does not increase model robustness on white-box scenarios, with full knowledge of the model under attack (Tramèr *et al.* (2018)). Developing an attack that finds adversarial examples with low noise in few iterations would enable adversarial training with such examples, which could potentially increase model robustness against white-box attacks.

[1] This also affects other machine learning classifiers, but we restrict our analysis to CNNs, that are most commonly used in computer vision tasks.

Developing attacks that minimize the norm of the adversarial perturbations requires optimizing two objectives: 1) obtaining a low L_2 norm, while 2) inducing a misclassification. With the current state-of-the-art method, C&W (Carlini & Wagner (2017)), this is addressed by using a two-term loss function, with the weight balancing the two competing objectives found via an expensive line search, requiring a large number of iterations. This makes the evaluation of a system's robustness very slow and it is unpractical for adversarial training.

In this paper, we propose an efficient gradient-based attack called *Decoupled Direction and Norm*[2] (DDN) that induces misclassification with a low L_2 norm. This attack optimizes the cross-entropy loss, and instead of penalizing the norm in each iteration, projects the perturbation onto a L_2-sphere centered at the original image. The change in norm is then based on whether the sample is adversarial or not. Using this approach to decouple the direction and norm of the adversarial noise leads to an attack that needs significantly fewer iterations, achieving a level of performance comparable to state-of-the-art, while being amenable to be used for adversarial training.

A comprehensive set of experiments was conducted using the MNIST, CIFAR-10 and ImageNet datasets. Our attack obtains comparable results to the state-of-the-art while requiring much fewer iterations (~100 times less than C&W). For untargeted attacks on the ImageNet dataset, our attack achieves better performance than the C&W attack, taking less than 10 minutes to attack 1 000 images, versus over 35 hours to run the C&W attack.

Results for adversarial training on the MNIST and CIFAR-10 datasets indicate that DDN can achieve state-of-the-art robustness compared to the Madry defense (Madry, Makelov, Schmidt, Tsipras & Vladu (2018)). These models require that attacks use a higher average L_2 norm to induce misclassifications. They also obtain a higher accuracy when the L_2 norm of the attacks is bounded. On MNIST, if the attack norm is restricted to 1.5, the model trained with the Madry defense achieves 67.3% accuracy, while our model achieves 87.2% accuracy. On CIFAR-10, for

[2] Code available at https://github.com/jeromerony/fast_adversarial.

attacks restricted to a norm of 0.5, the Madry model achieves 56.1% accuracy, compared to 67.6% in our model.

3. Related Work

In this section, we formalize the problem of adversarial examples, the threat model, and review the main attack and defense methods proposed in the literature.

3.1 Problem Formulation

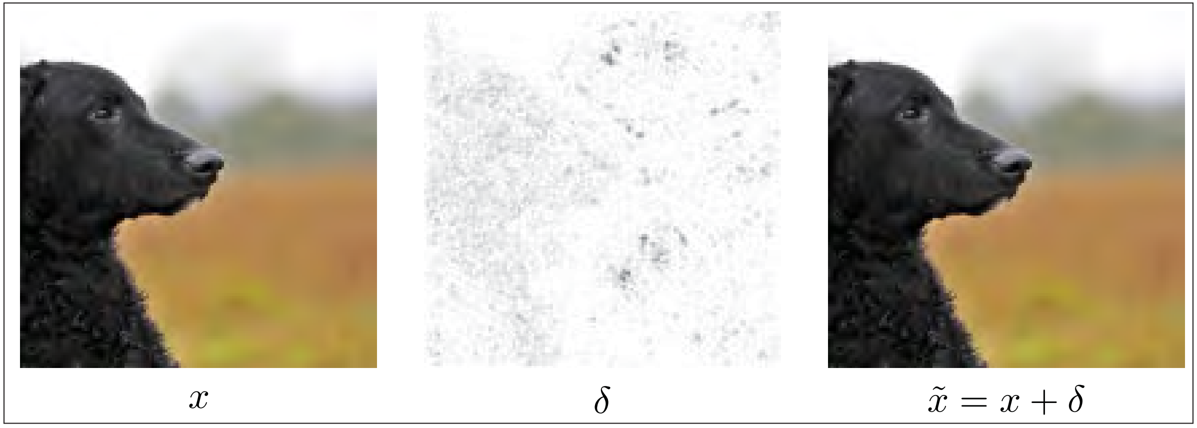


Figure-A V-1 Example of an adversarial image on the ImageNet dataset. The sample x is recognized as a Curly-coated retriever. Adding a perturbation δ we obtain an adversarial image that is classified as a microwave (with $\|\delta\|_2 = 0.7$).

Let x be an sample from the input space $\mathcal{X}$, with label y_{true} from a set of possible labels $\mathcal{Y}$. Let $D(x_1, x_2)$ be a distance measure that compares two input samples (ideally capturing their perceptual similarity). $P(\text{y}|x, \theta)$ is a model (classifier) parameterized by θ. An example $\tilde{x} \in \mathcal{X}$ is called *adversarial* (for non-targeted attacks) against the classifier if $\arg\max_j P(y_j|\tilde{x}, \theta) \neq y_{\text{true}}$ and $D(x, \tilde{x}) \leq \epsilon$, for a given maximum perturbation ϵ. A *targeted attack* with a given desired class y_{target} further requires that $\arg\max_j P(y_j|\tilde{x}, \theta) = y_{\text{target}}$. We denote as $J(x, y, \theta)$, the cross-entropy between the prediction of the model for an input x and a label y. Figure-A V-1 illustrates a targeted attack on the ImageNet dataset, against an Inception v3 model (Szegedy, Vanhoucke, Ioffe, Shlens & Wojna (2016b)).

In this paper, attacks are considered to be generated by a gradient-based optimization procedure, restricting our analysis to differentiable classifiers. These attacks can be formulated either to obtain a minimum distortion $D(x, \tilde{x})$, or to obtain the worst possible loss in a region $D(x, \tilde{x}) \leq \epsilon$. As an example, consider that the distance function is a norm (*e.g*. L_0, L_2 or L_∞), and the inputs are images (where each pixel's value is constrained between 0 and M). In a white-box scenario, the optimization procedure to obtain an non-targeted attack with minimum distortion δ can be formulated as:

$$\begin{aligned} \min_{\delta} \|\delta\| \quad \text{subject to} \quad & \arg\max_{j} P(y_j|x+\delta,\theta) \neq y_{\text{true}} \\ \text{and} \quad & 0 \leq x+\delta \leq M \end{aligned} \tag{A V-1}$$

With a similar formulation for *targeted attacks*, by changing the constraint to be equal to the target class.

If the objective is to obtain the worst possible loss for a given maximum noise of norm ϵ, the problem can be formulated as:

$$\begin{aligned} \min_{\delta} P(y_{\text{true}}|x+\delta,\theta) \;\; \text{subject to} \quad & \|\delta\| \leq \epsilon \\ \text{and} \quad & 0 \leq x+\delta \leq M \end{aligned} \tag{A V-2}$$

With a similar formulation for *targeted attacks*, by maximizing $P(y_{\text{target}}|x+\delta,\theta)$.

We focus on gradient-based attacks that optimize the L_2 norm of the distortion. While this distance does not perfectly capture perceptual similarity, it is widely used in computer vision to measure similarity between images (*e.g*. comparing image compression algorithms, where Peak Signal-to-Noise Ratio is used, which is directly related to the L_2 measure). A differentiable distance measure that captures perceptual similarity is still an open research problem.

3.2 Threat Model

In this paper, a *white-box* scenario is considered, also known as a Perfect Knowledge scenario (Biggio & Roli (2018)). In this scenario, we consider that an attacker has perfect knowledge of the system, including the neural network architecture and the learned weights θ. This threat model serves to evaluate system security under the *worst case* scenario. Other scenarios can be conceived to evaluate attacks under different assumptions on the attacker's knowledge, for instance, no access to the trained model, no access to the same training set, among others. These scenarios are referred as *black-box* or Limited-Knowledge (Biggio & Roli (2018)).

3.3 Attacks

Several attacks were proposed in the literature, either focusing on obtaining adversarial examples with a small δ (Equation A V-1) (Carlini & Wagner (2017); Moosavi-Dezfooli, Fawzi & Frossard (2016); Szegedy *et al.* (2014)), or on obtaining adversarial examples in one (or few) steps for adversarial training (Goodfellow *et al.* (2015); Kurakin, Goodfellow & Bengio (2017)).

L-BFGS. Szegedy *et al.* (2014) proposed an attack for minimally distorted examples (Equation A V-1), by considering the following approximation:

$$\begin{gathered}\min_{\delta} C\,\|\delta\|_2 + \log P(y_{\text{true}}|x+\delta,\theta) \\ \text{subject to}\;\; 0 \leq x+\delta \leq M\end{gathered} \tag{A V-3}$$

where the constraint $x + \delta \in [0, M]^n$ was addressed by using a box-constrained optimizer (L-BFGS: Limited memory Broyden–Fletcher–Goldfarb–Shanno), and a line-search to find an appropriate value of C.

FGSM. Goodfellow *et al.* (2015) proposed the Fast Gradient Sign Method, a one-step method that could generate adversarial examples. The original formulation was developed considering the L_∞ norm, but it has also been used to generate attacks that focus on the L_2 norm as follows:

$$\tilde{x} = x + \epsilon \frac{\nabla_x J(x,y,\theta)}{||\nabla_x J(x,y,\theta)||} \tag{A V-4}$$

where the constraint $\tilde{x} \in [0, M]^n$ was addressed by simply clipping the resulting adversarial example.

DeepFool. This method considers a linear approximation of the model, and iteratively refines an adversary example by choosing the point that would cross the decision boundary under this approximation. This method was developed for untargeted attacks, and for any L_p norm (Moosavi-Dezfooli *et al.* (2016)).

C&W. Similarly to the L-BFGS method, the C&W L_2 attack (Carlini & Wagner (2017)) minimizes two criteria at the same time – the perturbation that makes the sample adversarial (*e.g.* misclassified by the model), and the L_2 norm of the perturbation. Instead of using a box-constrained optimization method, they propose changing variables using the tanh function, and instead of optimizing the cross-entropy of the adversarial example, they use a difference between logits. For a targeted attack aiming to obtain class t, with Z denoting the model output before the softmax activation (logits), it optimizes:

$$\begin{aligned} &\min_{\delta} \left[\|\tilde{x} - x\|_2^2 + C f(\tilde{x}) \right] \\ \text{where} \quad & f(\tilde{x}) = \max(\max_{i \neq t}\{Z(\tilde{x})_i\} - Z(\tilde{x})_t, -\kappa) \\ \text{and} \quad & \tilde{x} = \frac{1}{2}(\tanh(\operatorname{arctanh}(x) + \delta) + 1) \end{aligned} \tag{A V-5}$$

where $Z(\tilde{x})_i$ denotes the logit corresponding to the i-th class. By increasing the confidence parameter κ, the adversarial sample will be misclassified with higher confidence. To use this attack in the untargeted setting, the definition of f is modified to $f(\tilde{x}) = \max(Z(\tilde{x})_y - \max_{i \neq y}\{Z(\tilde{x})_i\}, -\kappa)$ where y is the original label.

3.4 Defenses

Developing defenses against adversarial examples is an active area of research. To some extent, there is an *arms race* on developing defenses and attacks that break them. Goodfellow *et al*. proposed a method called *adversarial training* (Goodfellow *et al.* (2015)), in which the training data is augmented with FGSM samples. This was later shown not to be robust against iterative white-box attacks, nor black-box single-step attacks (Tramèr *et al.* (2018)). Papernot, McDaniel, Wu, Jha & Swami (2016) proposed a *distillation* procedure to train robust networks, which was shown to be easily broken by iterative white-box attacks (Carlini & Wagner (2017)). Other defenses involve *obfuscated gradients* (Athalye, Carlini & Wagner (2018)), where models either incorporate non-differentiable steps (such that the gradient cannot be computed) (Buckman, Roy, Raffel & Goodfellow (2018); Guo, Rana, Cissé & van der Maaten (2018)), or randomized elements (to induce incorrect estimations of the gradient) (Dhillon, Azizzadenesheli, Lipton, Bernstein, Kossaifi, Khanna & Anandkumar (2018); Xie, Wang, Zhang, Ren & Yuille (2018)). These defenses were later shown to be ineffective when attacked with Backward Pass Differentiable Approximation (BPDA) (Athalye *et al.* (2018)), where the actual model is used for forward propagation, and the gradient in the backward-pass is approximated. The Madry defense (Madry *et al.* (2018)), which considers a worst-case optimization, is the only defense that has been shown to be somewhat robust (on the MNIST and CIFAR-10 datasets). Below we provide more detail on the general approach of adversarial training, and the Madry defense.

Adversarial Training. This defense considers augmenting the training objective with adversarial examples (Goodfellow *et al.* (2015)), with the intention of improving robustness. Given a model with loss function $J(x, y, \theta)$, training is augmented as follows:

$$\tilde{J}(x, y, \theta) = \alpha J(x, y, \theta) + (1 - \alpha) J(\tilde{x}, y, \theta) \tag{A V-6}$$

where $\tilde{x}$ is an adversarial sample. In (Goodfellow *et al.* (2015)), the FGSM is used to generate the adversarial example in a single step. Tramèr *et al.* (2018) extended this method, showing that generating one-step attacks using the model under training introduced an issue. The model can converge to a degenerate solution where its gradients produce “easy” adversarial samples, causing the adversarial loss to have a limited influence on the training objective. They proposed a method in which an ensemble of models is also used to generate the adversarial examples $\tilde{x}$. This method displays some robustness against black-box attacks using surrogate models, but does not increase robustness in white-box scenarios.

Madry Defense. Madry *et al.* (2018) proposed a saddle point optimization problem, in which we optimize for the worst case:

$$\begin{aligned} &\min_{\theta} p(\theta) \\ \text{where} \quad & p(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\mathcal{S}} J(x+\delta, y, \theta)\Big] \end{aligned} \tag{A V-7}$$

where $\mathcal{D}$ is the training set, and $\mathcal{S}$ indicates the feasible region for the attacker (*e.g.* $\mathcal{S} = \{\delta : \|\delta\| < \epsilon\}$). They show that Equation A V-7 can be optimized by stochastic gradient descent – during each training iteration, it first finds the adversarial example that maximizes the loss around the current training sample x (*i.e.* maximizing the loss over δ, which is equivalent to minimizing the probability of the correct class as in Equation A V-2), and then, it minimizes the loss over θ. Experiments by Athalye *et al.* (2018) show that it was the only defense not broken under white-box attacks.

4. Decoupled Direction and Norm Attack

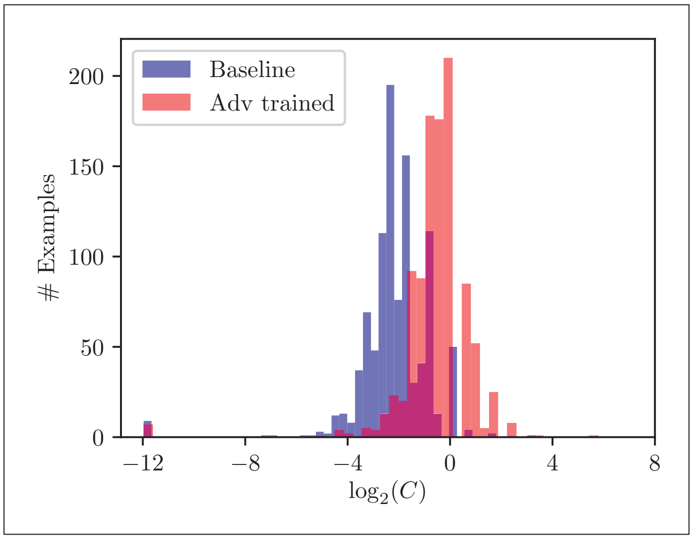


Figure-A V-2 Histogram of the best C found by the C&W algorithm with 9 search steps on the MNIST dataset

From the problem definition, we see that finding the worst adversary in a fixed region is an easier task. In Equation A V-2, both constraints can be expressed in terms of δ, and the resulting equation can be optimized using projected gradient descent. Finding the closest adversarial example is harder: Equation A V-1 has a constraint on the prediction of the model, which cannot be addressed by a simple projection. A common approach, which is used by Szegedy *et al.* (2014) and Carlini & Wagner (2017) is to approximate the constrained problem in Equation A V-1 by an unconstrained one, replacing the constraint with a *penalty*. This amounts to jointly optimizing both terms, the norm of δ and a classification term (see Eq. A V-3 and A V-5), with a sufficiently high parameter C. In the general context of constrained optimization, such a penalty-based approach is a well known general principle (Jensen & Bard (2003)). While tackling an unconstrained problem is convenient, penalty methods have well-known difficulties in practice. The main difficulty is that one has to choose parameter C in an *ad hoc* way. For instance, if C is too small in Equation A V-5, the example will not be adversarial; if it is too large, this term will dominate, and result in an adversarial example with more noise. This can be particularly problematic when optimizing with a low number of steps (*e.g.* to enable its use in adversarial training). Figure-A V-2 plots a histogram of the values of C that were obtained by running the C&W attack on the MNIST dataset. We can see that the optimum C varies significantly among different examples, ranging from 2^{-11} to 2^{5}. We also see that the distribution of the best constant C changes whether we attack a model with or without adversarial training (adversarially trained models often require higher C). Furthermore, penalty methods typically result in slow convergence (Jensen & Bard (2003)).

Given the difficulty of finding the appropriate constant C for this optimization, we propose an algorithm that does not impose a penalty on the L_2 norm during the optimization. Instead, the norm is constrained by projecting the adversarial perturbation δ on an ϵ-sphere around the original image x. Then, the L_2 norm is modified through a binary decision. If the sample x_k is not adversarial at step k, the norm is increased for step $k + 1$, otherwise it is decreased.

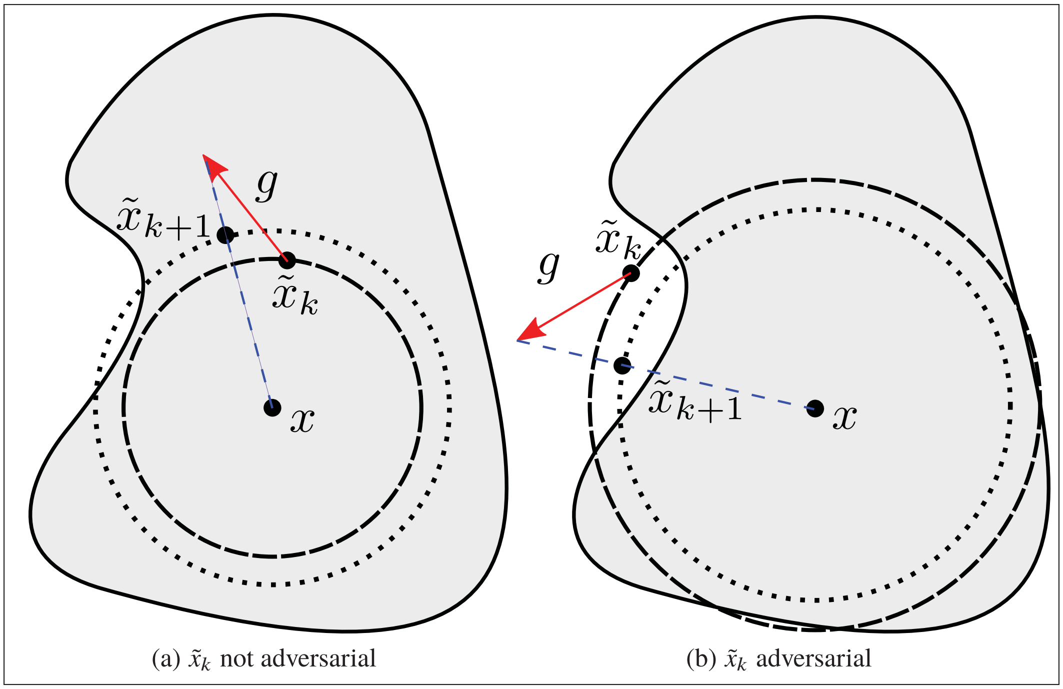
We also note that optimizing the cross-entropy may present two other difficulties. First, the function is not bounded, which can make it dominate in the optimization of Equation A V-3.

Algorithm-A V-1 Decoupled Direction and Norm Attack

```
**Input:** $x$: *original image to be attacked*
**Input:** $y$: *true label (untargeted) or target label (targeted)*
**Input:** $K$: *number of iterations*
**Input:** $\alpha$: *step size*
**Input:** $\gamma$: *factor to modify the norm in each iteration*
**Output:** $\tilde{x}$: *adversarial image*
Initialize $\delta_0 \leftarrow \mathbf{0}$, $\tilde{x}_0 \leftarrow x$, $\epsilon_0 \leftarrow 1$
If targeted attack: $m \leftarrow -1$ else $m \leftarrow +1$
for $k \leftarrow 1$ to $K$ do
    $g \leftarrow m\nabla_{\tilde{x}_{k-1}} J(\tilde{x}_{k-1}, y, \theta)$
    $g \leftarrow \alpha \frac{g}{\|g\|_2}$ // Step of size $\alpha$ in the direction of $g$
    $\delta_k \leftarrow \delta_{k-1} + g$
    if $\tilde{x}_{k-1}$ is adversarial then
        $\epsilon_k \leftarrow (1 - \gamma)\epsilon_{k-1}$ // Decrease norm
    end
    else
        $\epsilon_k \leftarrow (1 + \gamma)\epsilon_{k-1}$ // Increase norm
    end
    $\tilde{x}_k \leftarrow x + \epsilon_k \frac{\delta_k}{\|\delta_k\|_2}$ // Project $\delta_k$ onto an $\epsilon_k$-sphere around $x$
    $\tilde{x}_k \leftarrow \text{clip}(\tilde{x}_k, 0, 1)$ // Ensure $\tilde{x}_k \in \mathcal{X}$
end
Return $\tilde{x}_k$ that has lowest norm $\|\tilde{x}_k - x\|_2$ and is adversarial
```

Second, when attacking trained models, often the predicted probability of the correct class for the original image is very close to 1, which causes the cross entropy to start very low and increase by several orders of magnitude during the search for an adversarial example. This affects the norm of the gradient, making it hard to find an appropriate learning rate. C&W address these issues by optimizing the difference between logits instead of the cross-entropy. In this work, the issue of it being unbounded does not affect the attack procedure, since the decision to update the norm is done on the model's prediction (not on the cross-entropy). In order to handle the issue of large changes in gradient norm, we normalize the gradient to have unit norm before taking a step in its direction.

The full procedure is described in algorithm V-1 and illustrated in Figure-A V-3. We start from the original image x, and iteratively refine the noise δ_k. In iteration k, if the current sample



(a) $\tilde{x}_k$ not adversarial (b) $\tilde{x}_k$ adversarial

Figure-A V-3 Illustration of an untargeted attack. The shaded area denotes the region of the input space classified as y_{true}. In (a), $\tilde{x}_k$ is still not adversarial, and we increase the norm ϵ_{k+1} for the next iteration, otherwise it is reduced in (b). In both cases, we take a step g starting from the current point $\tilde{x}$, and project back to an ϵ_{k+1}-sphere centered at x.

$\tilde{x}_k = x + \delta_k$ is still not adversarial, we consider a larger norm $\epsilon_{k+1} = (1 + \gamma)\epsilon_k$. Otherwise, if the sample is adversarial, we consider a smaller $\epsilon_{k+1} = (1 - \gamma)\epsilon_k$. In both cases, we take a step g (step 5 of algorithm V-1) from the point $\tilde{x}_k$ (red arrow in Figure-A V-3), and project it back onto an ϵ_{k+1}-sphere centered at x (the direction given by the dashed blue line in Figure-A V-3), obtaining $\tilde{x}_{k+1}$. Lastly, $\tilde{x}_{k+1}$ is projected onto the feasible region of the input space $\mathcal{X}$. In the case of images normalized to $[0, 1]$, we simply clip the value of each pixel to be inside this range (step 13 of algorithm V-1). Besides this step, we can also consider quantizing the image in each iteration, to ensure the attack is a valid image.

It's worth noting that, when reaching a point where the decision boundary is tangent to the ϵ_k-sphere, g will have the same direction as δ_{k+1}. This means that δ_{k+1} will be projected on the direction of δ_k. Therefore, the norm will oscillate between the two sides of the decision boundary in this direction. Multiplying ϵ by $1 + \gamma$ and $1 - \gamma$ will result in a global decrease (on two steps) of the norm by $1 - \gamma^2$, leading to a finer search of the best norm.

5. Attack Evaluation

Table-A V-1 Performance of our DDN attack compared to C&W and DeepFool attacks on MNIST, CIFAR-10 and ImageNet in the untargeted scenario

	Attack	**Budget**	**Success**	**Mean L_2**	**Median L_2**	**#Grads**	**Run-time (s)**
MNIST	C&W	4×25	100.0	1.7382	1.7400	100	1.7
		1×100	99.4	1.5917	1.6405	100	1.7
		9×10 000	100.0	**1.3961**	1.4121	54 007	856.8
	DeepFool	100	75.4	1.9685	2.2909	98	-
	DDN	100	100.0	1.4563	1.4506	100	1.5
		300	100.0	1.4357	1.4386	300	4.5
		1 000	100.0	1.4240	1.4342	1 000	14.9
CIFAR-10	C&W	4×25	100.0	0.1924	0.1541	60	3.0
		1×100	99.8	0.1728	0.1620	91	4.6
		9×10 000	100.0	0.1543	0.1453	36 009	1 793.2
	DeepFool	100	99.7	0.1796	0.1497	25	-
	DDN	100	100.0	0.1503	0.1333	100	4.7
		300	100.0	0.1487	0.1322	300	14.2
		1 000	100.0	**0.1480**	0.1317	1 000	47.6
ImageNet	C&W	4×25	100.0	1.5812	1.3382	63	379.3
		1×100	100.0	0.9858	0.9587	48	287.1
		9×10 000	100.0	0.4692	0.3980	21 309	127 755.6
	DeepFool	100	98.5	0.3800	0.2655	41	-
	DDN	100	99.6	0.3831	0.3227	100	593.6
		300	100.0	0.3749	0.3210	300	1 779.4
		1 000	100.0	**0.3617**	0.3188	1 000	5 933.6

Experiments were conducted on the MNIST, CIFAR-10 and ImageNet datasets, comparing the proposed attack to the state-of-the-art L_2 attacks proposed in the literature: DeepFool (Moosavi-Dezfooli *et al.* (2016)) and C&W L_2 attack (Carlini & Wagner (2017)). We use the same model architectures with identical hyperparameters for training as in (Carlini & Wagner (2017)) for MNIST and CIFAR-10 (see the supplementary material for details). Our base classifiers obtain 99.44% and 85.51% accuracy on the test sets of MNIST and CIFAR-10, respectively. For the ImageNet experiments, we use a pre-trained Inception V3 (Szegedy *et al.*

Table-A V-2 Comparison of the DDN attack to the C&W L_2 attack on MNIST

	Average case		Least Likely	
Attack	**Success**	**Mean L_2**	**Success**	**Mean L_2**
C&W 4×25	96.11	2.8254	69.9	5.0090
C&W 1×100	86.89	2.0940	31.7	2.6062
C&W 9×10 000	100.00	1.9481	100.0	2.5370
DDN 100	100.00	1.9763	100.0	2.6008
DDN 300	100.00	1.9577	100.0	2.5503
DDN 1 000	100.00	1.9511	100.0	2.5348

Table-A V-3 Comparison of the DDN attack to the C&W L_2 attack on CIFAR-10

	Average case		Least Likely	
Attack	**Success**	**Mean L_2**	**Success**	**Mean L_2**
C&W 4×25	99.78	0.3247	98.7	0.5060
C&W 1×100	99.32	0.3104	95.8	0.4159
C&W 9×10 000	100.00	0.2798	100.0	0.3905
DDN 100	100.00	0.2925	100.0	0.4170
DDN 300	100.00	0.2887	100.0	0.4090
DDN 1 000	100.00	0.2867	100.0	0.4050

(2016b)), that achieves 22.51% top-1 error on the validation set. Inception V3 takes images of size 299×299 as input, which are cropped from images of size 342×342.

Table-A V-4 Comparison of the DDN attack to the C&W L_2 attack on ImageNet. For C&W 9×10 000, we report the results from Carlini & Wagner (2017).

	Average case		Least Likely	
Attack	**Success**	**Mean L_2**	**Success**	**Mean L_2**
C&W 4×25	99.13	4.2826	80.6	8.7336
C&W 1×100	96.74	1.7718	66.2	2.2997
C&W 9×10 000	100.00	0.96	100.0	2.22
DDN 100	99.98	1.0260	99.5	1.7074
DDN 300	100.00	0.9021	100.0	1.3634
DDN 1 000	100.00	0.8444	100.0	1.2240

For experiments with DeepFool (Moosavi-Dezfooli *et al.* (2016)), we used the implementation from Foolbox (Rauber, Brendel & Bethge (2017)), with a budget of 100 iterations. For the experiments with C&W, we ported the attack (originally implemented on TensorFlow) on PyTorch to evaluate the models in the frameworks in which they were trained. We use the same hyperparameters from (Carlini & Wagner (2017)): 9 search steps on C with an initial constant of 0.01, with 10 000 iterations for each search step (with early stopping) - we refer to this scenario as C&W 9×10 000 in the tables. As we are interested in obtaining attacks that require few iterations, we also report experiments in a scenario where the number of iterations is limited to 100. We consider a scenario of running 100 steps with a fixed C (1×100), and a scenario of running 4 search steps on C, of 25 iterations each (4×25). Since the hyperparameters proposed in (Carlini & Wagner (2017)) were tuned for a larger number of iterations and search steps, we performed a grid search for each dataset, using learning rates in the range [0.01, 0.05, 0.1, 0.5, 1], and C in the range [0.001, 0.01, 0.1, 1, 10, 100, 1 000]. We report the results for C&W with the hyperparameters that achieve best Median L_2. Selected parameters are listed in the supplementary material.

For the experiments using DDN, we ran attacks with budgets of 100, 300 and 1 000 iterations, in all cases, using $\epsilon_0 = 1$ and $\gamma = 0.05$. The initial step size $\alpha = 1$, was reduced with cosine annealing to 0.01 in the last iteration. The choice of γ is based on the encoding of images. For any correctly classified image, the smallest possible perturbation consists in changing one pixel by $1/255$ (for images encoded in 8 bit values), corresponding to a norm of $1/255$. Since we perform quantization, the values are rounded, meaning that the algorithm must be able to achieve a norm lower than $1.5/255 = 3/510$. When using K steps, this imposes:

$$\epsilon_0(1-\gamma)^K < \frac{3}{510} \Rightarrow \gamma > 1 - \left(\frac{3}{510\,\epsilon_0}\right)^{\frac{1}{K}} \tag{A V-8}$$

Using $\epsilon_0 = 1$ and $K = 100$ yields $\gamma \simeq 0.05$. Therefore, if there exists an adversarial example with smallest perturbation, the algorithm may find it in a fixed number of steps.

For the results with DDN, we consider quantized images (to 256 levels). The quantization step is included in each iteration (see step 13 of algorithm V-1). All results reported in the paper consider images in the [0, 1] range.

Two sets of experiments were conducted: untargeted attacks and targeted attacks. As in (Carlini & Wagner (2017)), we generated attacks on the first 1 000 images of the test set for MNIST and CIFAR-10, while for ImageNet we randomly chose 1 000 images from the validation set that are correctly classified. For the untargeted attacks, we report the success rate of the attack (percentage of samples for which an attack was found), the mean L_2 norm of the adversarial noise (for successful attacks), and the median L_2 norm over all attacks while considering unsuccessful attacks as worst-case adversarial (distance to a uniform gray image, as introduced by (Brendel *et al.* (2018))). We also report the average number (for batch execution) of gradient computations and the total run-times (in seconds) on a NVIDIA GTX 1080 Ti with 11GB of memory. We did not report run-times for the DeepFool attack, since the implementation from foolbox generates adversarial examples one-by-one and is executed on CPU, leading to unrepresentative run-times. Attacks on MNIST and CIFAR-10 have been executed in a single batch of 1 000 samples, whereas attacks on ImageNet have been executed in 20 batches of 50 samples.

For the targeted attacks, following the protocol from (Carlini & Wagner (2017)), we generate attacks against all possible classes on MNIST and CIFAR-10 (9 attacks per image), and against 100 randomly chosen classes for ImageNet (10% of the number of classes). Therefore, in each targeted attack experiment, we run 9 000 attacks on MNIST and CIFAR-10, and 100 000 attacks on ImageNet. Results are reported for two scenarios: 1) average over all attacks; 2) average performance when choosing the least likely class (*i.e.* choosing the worst attack performance over all target classes, for each image). The reported L_2 norms are, as in the untargeted scenario, the means over successful attacks.

Table-A V-1 reports the results of DDN compared to the C&W L_2 and DeepFool attacks on the MNIST, CIFAR-10 and ImageNet datasets. For the MNIST and CIFAR-10 datasets, results with DDN are comparable to the state-of-the-art. DDN obtains slightly worse L_2 norms on the

MNIST dataset (when compared to the C&W 9×10 000), however, our attack is able to get within 5% of the norm found by C&W in only 100 iterations compared to the 54 007 iterations required for the C&W L_2 attack. When the C&W attack is restricted to use a maximum of 100 iterations, it always performed worse than DDN with 100 iterations. On the ImageNet dataset, our attack obtains better Mean L_2 norms than both other attacks. The DDN attack needs 300 iterations to reach 100% success rate. DeepFool obtains close results but fails to reach 100% success rate. It is also worth noting that DeepFool seems to performs worse against adversarially trained models (discussed in Section 7). Supplementary material reports curves of the perturbation size against accuracy of the models for the three attacks.

Tables V-2, V-3 and V-4 present the results on targeted attacks on the MNIST, CIFAR-10 and ImageNet datasets, respectively. For the MNIST and CIFAR-10 datasets, DDN yields similar performance compared to the C&W attack with 9×10 000 iterations, and always perform better than the C&W attack when it is restricted to 100 iterations (we re-iterate that the hyperparameters for the C&W attack were tuned for each dataset, while the hyperparameters for DDN are fixed for all experiments). On the ImageNet dataset, DDN run with 100 iterations obtains superior performance than C&W. For all datasets, with the scenario restricted to 100 iterations, the C&W algorithm has a noticeable drop in success rate for finding adversarial examples to the least likely class.

6. Adversarial Training with DDN

Since the DDN attack can produce adversarial examples in relatively few iterations, it can be used for adversarial training. For this, we consider the following loss function:

$$\tilde{J}(x, y, \theta) = J(\tilde{x}, y, \theta) \qquad \text{(A V-9)}$$

where $\tilde{x}$ is an adversarial example produced by the DDN algorithm, that is projected to an ϵ-ball around x, such that the classifier is trained with adversarial examples with a maximum norm of ϵ.

It is worth making a parallel of this approach with the Madry defense (Madry *et al.* (2018)) where, in each iteration, the loss of the worst-case adversarial (see Equation A V-2) in an ϵ-ball around the original sample x is used for optimization. In our proposed adversarial training procedure, we optimize the loss of the closest adversarial example (see Equation A V-1). The intuition of this defense is to push the decision boundary away from x in each iteration. We do note that this method does not have the theoretical guarantees of the Madry defense. However, since in practice the Madry defense uses approximations (when searching for the global maximum of the loss around x), we argue that both methods deserve empirical comparison.

7. Defense Evaluation

We trained models using the same architectures as (Carlini & Wagner (2017)) for MNIST, and a Wide ResNet (WRN) 28-10 (Zagoruyko & Komodakis (2016)) for CIFAR-10 (similar to (Madry *et al.* (2018)) where they use a WRN 34-10). As described in Section 6, we augment the training images with adversarial perturbations. For each training step, we run the DDN attack with a budget of 100 iterations, and limit the norm of the perturbation to a maximum $\epsilon = 2.4$ on the MNIST experiments, and $\epsilon = 1$ for the CIFAR-10 experiments. For MNIST, we train the model for 30 epochs with a learning rate of 0.01 and then for 20 epochs with a learning rate of 0.001. To reduce the training time with CIFAR-10, we first train the model on original images for 200 epochs using the hyperparameters from (Zagoruyko & Komodakis (2016)). Then, we continue training for 30 more epochs using Equation A V-9, keeping the same final learning rate of 0.0008. Our robust MNIST model has a test accuracy of 99.01% on the clean samples, while the Madry model has an accuracy of 98.53%. On CIFAR-10, our model reaches a test accuracy of 89.0% while the model by Madry *et al*. obtains 87.3%.

We evaluate the adversarial robustness of the models using three untargeted attacks: Carlini 9×10 000, DeepFool 100 and DDN 1 000. For each sample, we consider the smallest adversarial perturbation produced by the three attacks and report it in the "**All**" row. Tables V-5 and V-6 report the results of this evaluation with a comparison to the defense of Madry *et al.* (2018)[3] and

[3] Models taken from https://github.com/MadryLab

Table-A V-5 Evaluation of the robustness of our adversarial training on MNIST against the Madry defense

Defense	**Attack**	**Attack Success**	**Mean L_2**	**Median L_2**	**Model Accuracy at $\epsilon \leq 1.5$**
Baseline	C&W 9×10 000	100.0	1.3961	1.4121	42.1
	DeepFool 100	75.4	1.9685	2.2909	81.8
	DDN 1 000	100.0	1.4240	1.4342	45.2
	All	100.0	1.3778	1.3946	40.8
Madry *et al.*	C&W 9×10 000	100.0	2.0813	2.1071	73.0
	DeepFool 100	91.6	4.9585	5.2946	93.1
	DDN 1 000	99.6	1.8436	1.8994	69.9
	All	100.0	1.6917	1.8307	67.3
Ours	C&W 9×10 000	100.0	2.5181	2.6146	88.0
	DeepFool 100	94.3	3.9449	4.1754	92.7
	DDN 1 000	100.0	2.4874	2.5781	87.6
	All	100.0	**2.4497**	2.5538	**87.2**

Table-A V-6 Evaluation of the robustness of our adversarial training on CIFAR-10 against the Madry defense

Defense	**Attack**	**Attack Success**	**Mean L_2**	**Median L_2**	**Model Accuracy at $\epsilon \leq 0.5$**
Baseline WRN 28-10	C&W 9×10 000	100.0	0.1343	0.1273	0.2
	DeepFool 100	99.3	0.5085	0.4241	38.3
	DDN 1 000	100.0	0.1430	0.1370	0.1
	All	100.0	0.1282	0.1222	0.1
Madry *et al.* WRN 34-10	C&W 9×10 000	100.0	0.6912	0.6050	57.1
	DeepFool 100	95.6	1.4856	0.9576	64.7
	DDN 1 000	100.0	0.6732	0.5876	56.9
	All	100.0	0.6601	0.5804	56.1
Ours WRN 28-10	C&W 9×10 000	100.0	0.8860	0.8254	67.9
	DeepFool 100	99.7	1.5298	1.1163	69.9
	DDN 1 000	100.0	0.8688	0.8177	68.0
	All	100.0	**0.8597**	0.8151	**67.6**

the baseline (without adversarial training) for CIFAR-10. For MNIST, the baseline corresponds to the model used in Section 5. We observe that for attacks with unbounded norm, the attacks can successfully generate adversarial examples almost 100% of the time. However, an increased L_2 norm is required to generate attacks against the model trained with DDN.

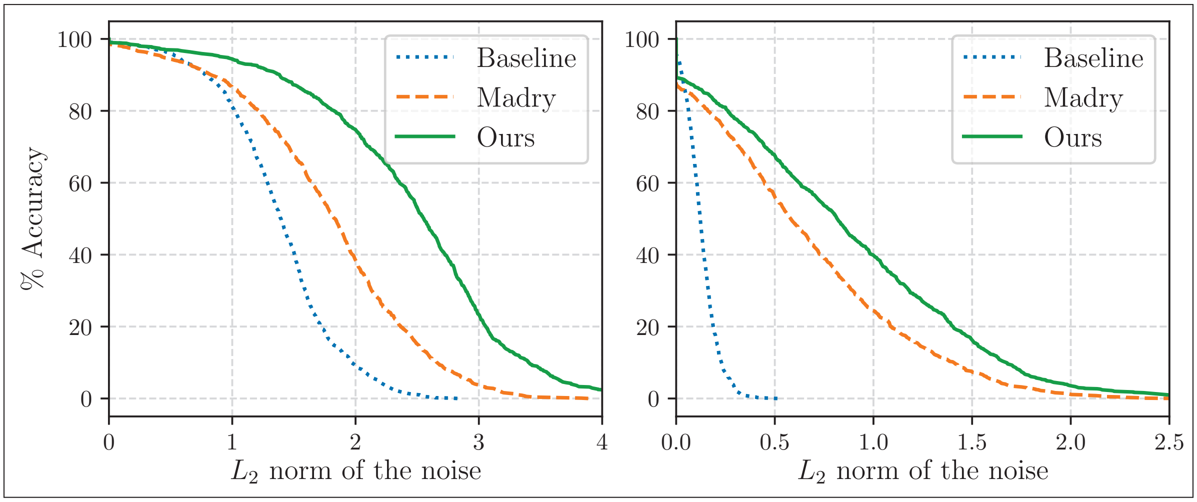


Figure-A V-4 Models robustness on MNIST (left) and CIFAR-10 (right): impact on accuracy as we increase the maximum perturbation ϵ

Figure-A V-4 shows the robustness of the MNIST and CIFAR-10 models respectively for different attacks with increasing maximum L_2 norm. These figures can be interpreted as the expected accuracy of the systems in a scenario where the adversary is constrained to make changes with norm $L_2 \leq \epsilon$. For instance on MNIST, if the attacker is limited to a maximum norm of $\epsilon = 1.5$, the baseline performance decreases to 40.8%; Madry to 67.3% and our defense to 87.2%. At $\epsilon = 2.0$, baseline performance decreases to 9.2%, Madry to 38.6% and our defense to 74.8%. On CIFAR-10, if the attacker is limited to a maximum norm of $\epsilon = 0.5$, the baseline performance decreases to 0.1%; Madry to 56.1% and our defense to 67.6%. At $\epsilon = 1.0$, baseline performance decreases to 0%, Madry to 24.4% and our defense to 39.9%. For both datasets, the model trained with DDN outperforms the model trained with the Madry defense for all values of ϵ.

Figure-A V-5 shows adversarial examples produced by the DDN 1 000 attack for different models on MNIST and CIFAR-10. On MNIST, adversarial examples for the baseline are not meaningful (the still visually belong to the original class), whereas some adversarial examples obtained for the adversarially trained model (DDN) actually change classes (bottom right: 0 changes to 6). For all models, there are still some adversarial examples that are very close to the original images (first column). On CIFAR-10, while the adversarially trained models require higher norms for the attacks, most adversarial examples still perceptually resemble the original images. In few cases (bottom-right example for CIFAR-10), it could cause a confusion: it can appear as changing to class 1 - a (cropped) automobile facing right.

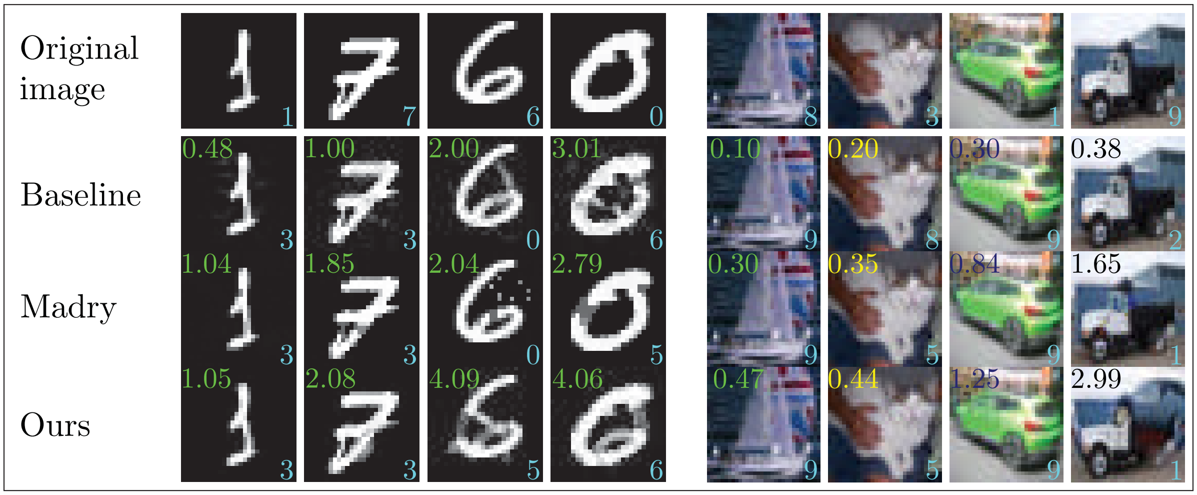


Figure-A V-5 Adversarial examples with varied levels of noise δ against three models: baseline, Madry defense and our defense. Text on top-left of each image indicate $||\delta||_2$; text on bottom-right indicates the predicted class[4].

8. Conclusion

We presented the *Decoupled Direction and Norm* attack, which obtains comparable results with the state-of-the-art for L_2 norm adversarial perturbations, but in much fewer iterations. Our attack allows for faster evaluation of the robustness of differentiable models, and enables a novel adversarial training, where, at each iteration, we train with examples close to the decision boundary. Our experiments with MNIST and CIFAR-10 show state-of-the-art robustness against L_2-based attacks in a white-box scenario. Future work includes the evaluation of the transferability of attacks in black-box scenarios.

The methods presented in this paper were used in NIPS 2018 Adversarial Vision Challenge Brendel *et al.* (2018), ranking first in untargeted attacks, and third in targeted attacks and

[4] For CIFAR-10: 1: automobile, 2: bird, 3: cat, 5: dog, 8: ship, 9: truck.

robust models (both attacks and defense in a black-box scenario). These results highlight the effectiveness of the defense mechanism, and suggest that attacks using adversarially-trained surrogate models can be effective in black-box scenarios, which is a promising future direction.

Supplementary material

9. Model architectures

Table-A V-7 lists the architectures of the CNNs used in the Attack Evaluation - we used the same architecture as in (Carlini & Wagner (2017)) for a fair comparison against the C&W and DeepFool attacks. Table-A V-8 lists the architecture used in the robust model (defense) trained on CIFAR-10. We used a Wide ResNet with 28 layers and widening factor of 10 (WRN-28-10). The residual blocks used are the “basic block" (He, Zhang, Ren & Sun (2015); Zagoruyko & Komodakis (2016)), with stride 1 for the first group and stride 2 for the second an third groups. This architecture is slightly different from the one used by Madry *et al.* (2018), where Madry *et al.* (2018) they use a modified version of Wide ResNet with 5 residual blocks instead of 4 in each group, and without convolutions in the residual connections (when the shape of the output changes, *e.g.* with stride=2).

Table-A V-7 CNN architectures used for the Attack Evaluation

Layer Type	**MNIST Model**	**CIFAR-10 Model**
Convolution + ReLU	$3\times3\times32$	$3\times3\times64$
Convolution + ReLU	$3\times3\times32$	$3\times3\times64$
Max Pooling	2×2	2×2
Convolution + ReLU	$3\times3\times64$	$3\times3\times128$
Convolution + ReLU	$3\times3\times64$	$3\times3\times128$
Max Pooling	2×2	2×2
Fully Connected + ReLU	200	256
Fully Connected + ReLU	200	256
Fully Connected + Softmax	10	10

10. Hyperparameters selected for the C&W attack

We considered a scenario of running the C&W attack with 100 steps and a fixed C (1×100), and a scenario of running 4 search steps on C, of 25 iterations each (4×25). Since the hyperparameters proposed in (Carlini & Wagner (2017)) were tuned for a larger number of iterations and search

Table-A V-8 CIFAR-10 architecture used for the Defense evaluation

Layer Type	**Size**
Convolution	$3 \times 3 \times 16$
Residual Block	$\begin{bmatrix} 3 \times 3, 160 \\ 3 \times 3, 160 \end{bmatrix} \times 4$
Residual Block	$\begin{bmatrix} 3 \times 3, 320 \\ 3 \times 3, 320 \end{bmatrix} \times 4$
Residual Block	$\begin{bmatrix} 3 \times 3, 640 \\ 3 \times 3, 640 \end{bmatrix} \times 4$
Batch Normalization + ReLU	-
Average Pooling	8×8
Fully Connected + Softmax	10

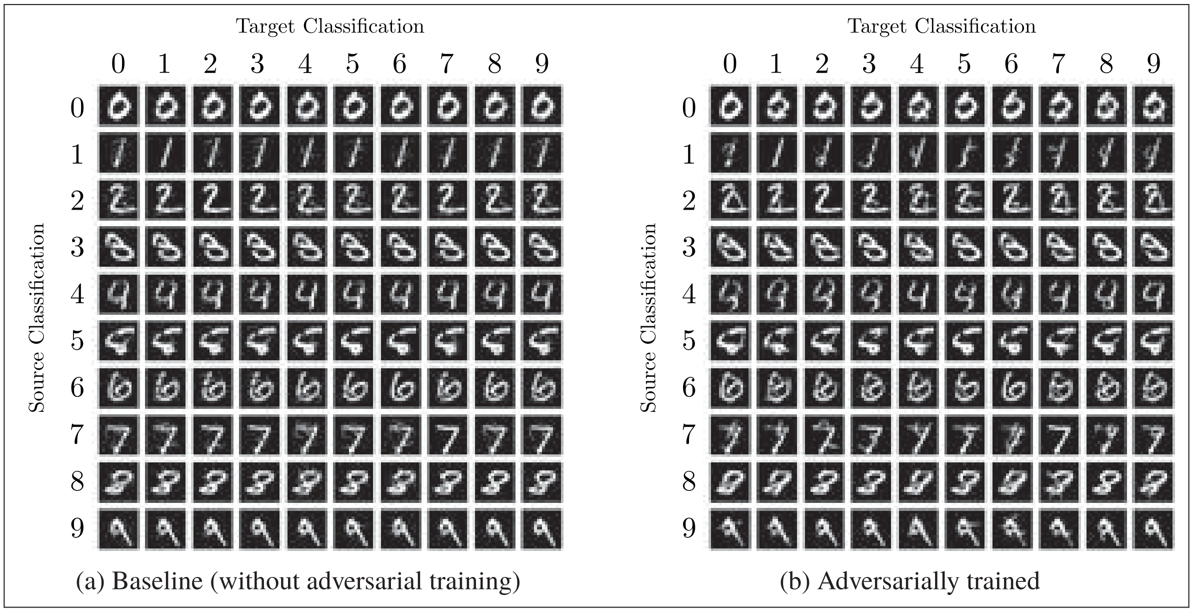
steps, we performed a grid search for each dataset, using learning rates in the range [0.01, 0.05, 0.1, 0.5, 1], and C in the range [0.001, 0.01, 0.1, 1, 10, 100, 1 000]. We selected the hyperparameters that resulted in targeted attacks with lowest Median L_2 for each dataset. Table-A V-9 lists the hyperparameters found through this search procedure.

Table-A V-9 Hyperparameters used for the C&W attack when restricted to 100 iterations.

Dataset	**# Iterations**	**Parameters**
MNIST	1×100	$\alpha = 0.1, C = 1$
MNIST	4×25	$\alpha = 0.5, C = 1$
CIFAR-10	1×100	$\alpha = 0.01, C = 0.1$
CIFAR-10	4×25	$\alpha = 0.01, C = 0.1$
ImageNet	1×100	$\alpha = 0.01, C = 1$
ImageNet	4×25	$\alpha = 0.01, C = 10$

11. Examples of adversarial images

Figure-A V-6 plots a grid of attacks (obtained with the C&W attack) against the first 10 examples in the MNIST dataset. The rows indicate the source classification (label), and the columns indicate the target class used to generate the attack (images on the diagonal are the original



(a) Baseline (without adversarial training)

(b) Adversarially trained

Figure-A V-6 Adversarial examples obtained using the C&W L_2 attack on two models: (a) Baseline, (b) model adversarially trained with our attack

samples). We can see that in the adversarially trained model, the attacks need to introduce much larger changes to the samples in order to make them adversarial, and some of the adversarial samples visually resemble another class.

Figure-A V-7 shows randomly-selected adversarial examples for the CIFAR-10 dataset, comparing the baseline model (WRN 28-10), the Madry defense and our proposed defense. For each image and model, we ran three attacks (DDN 1 000, C&W 9×10 000, DeepFool 100), and present the adversarial example with minimum L_2 perturbation among them. Figure-A V-8 shows cherry-picked adversarial examples on CIFAR-10, that visually resemble another class, when attacking the proposed defense. We see that on the average case (randomly-selected), adversarial examples against the defenses still require low amounts of noise (perceptually) to induce misclassification. On the other hand, we notice that on adversarially trained models, some examples do require a much larger change on the image, making it effectively resemble another class.

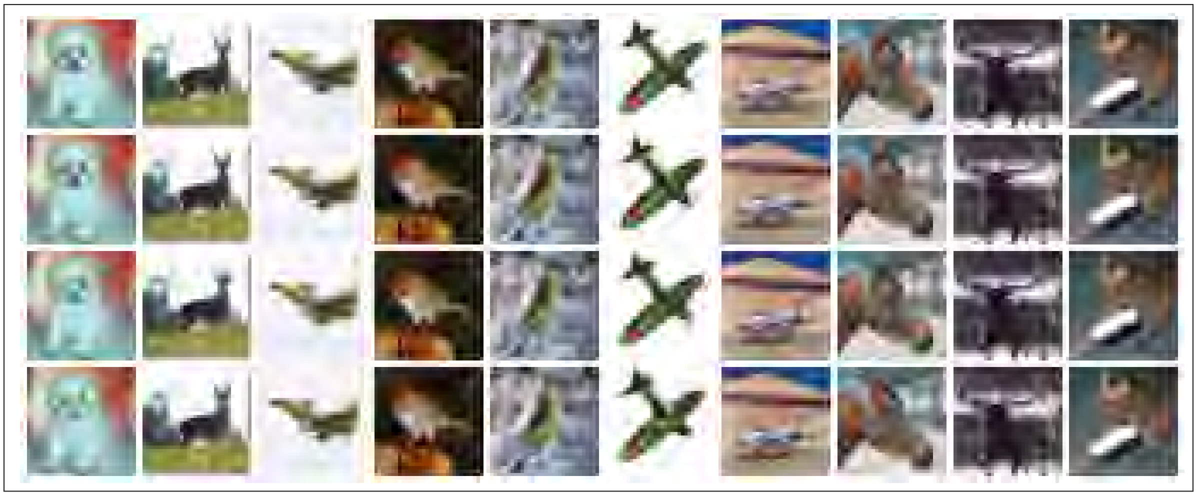


Figure-A V-7 Randomly chosen adversarial examples on CIFAR-10 for three models. **Top row**: original images; **second row**: attacks against the baseline; **third row**: attacks against the Madry defense.

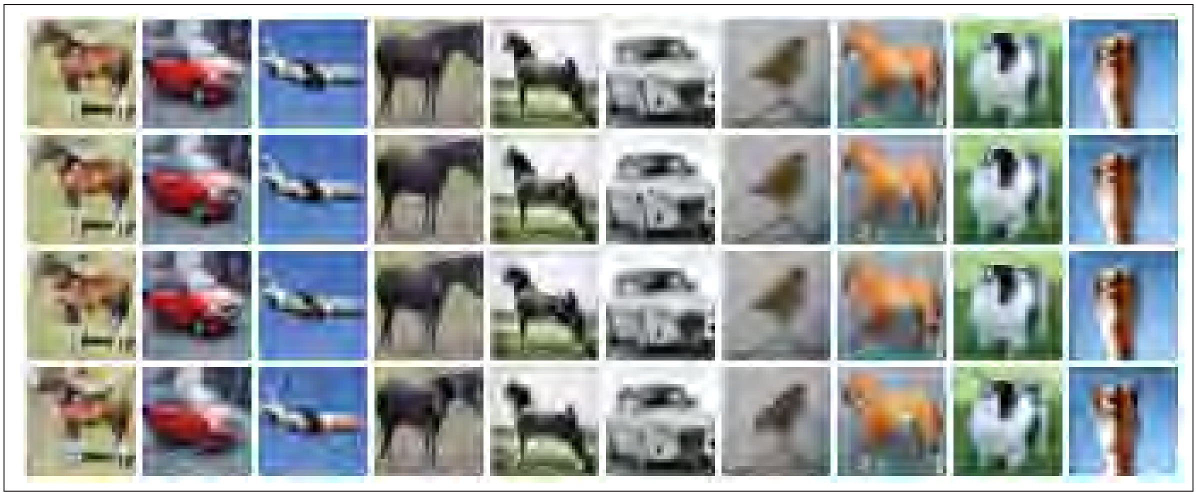
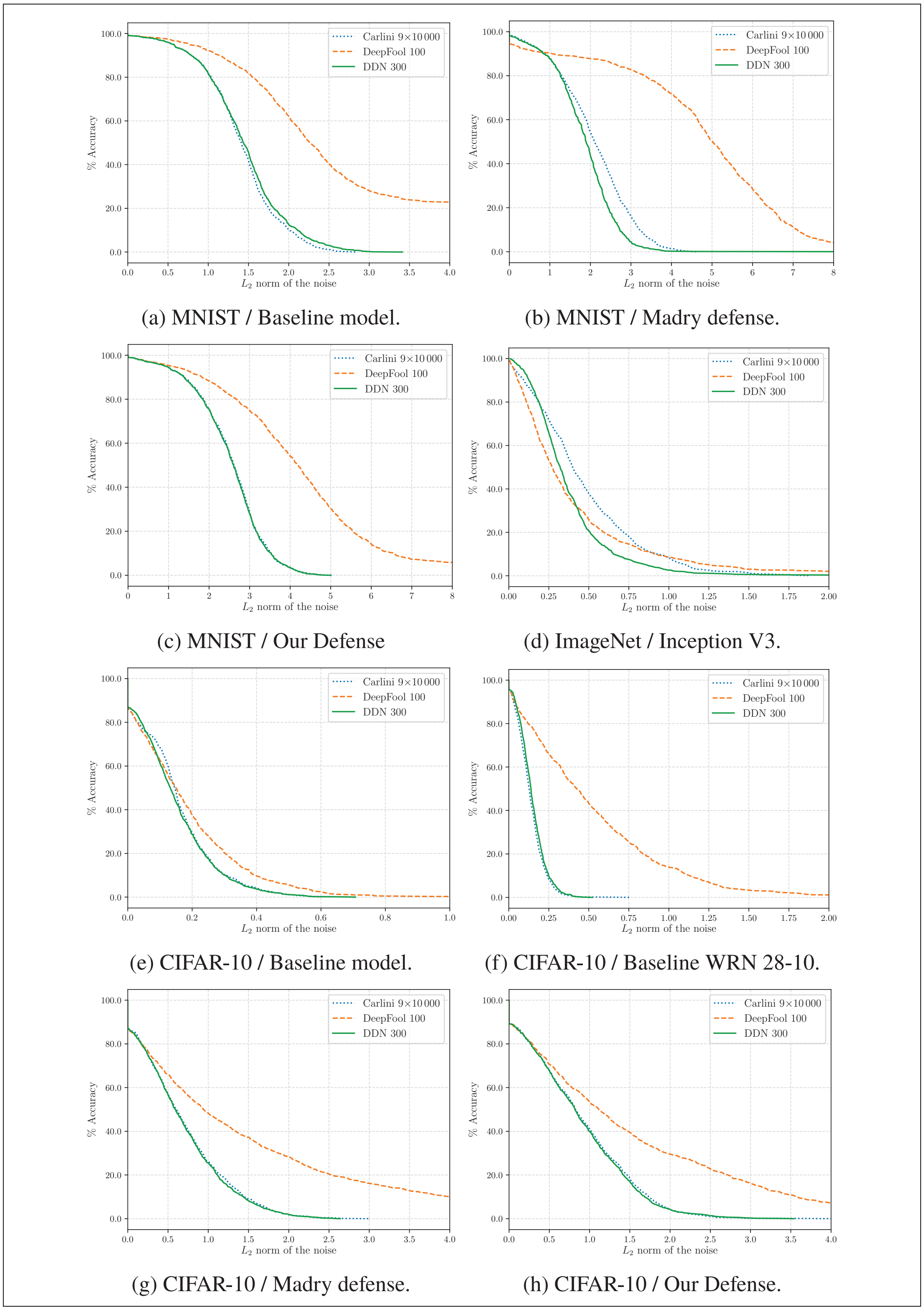


Figure-A V-8 Cherry-picked adversarial examples on CIFAR-10 for three models. **Top row**: original images; **second row**: attacks against the baseline; **third row**: attacks against the Madry defense; **bottom row**: attacks against the proposed defense. Predicted labels for the last row are, from left to right: dog, ship, deer, dog, dog, truck, horse, dog, cat, cat.

12. Attack performance curves

Figure-A V-9 reports curves of the perturbation size against accuracy of the models for three attacks: Carlini 9×10 000, DeepFool 100 and DDN 300.



(a) MNIST / Baseline model.

(b) MNIST / Madry defense.

(c) MNIST / Our Defense

(d) ImageNet / Inception V3.

(e) CIFAR-10 / Baseline model.

(f) CIFAR-10 / Baseline WRN 28-10.

(g) CIFAR-10 / Madry defense.

(h) CIFAR-10 / Our Defense.

Figure-A V-9 Attacks performances on different datasets and models

REFERENCES

Aresta, G., Araújo, T., Kwok, S. et al. (2018). BACH: Grand Challenge on Breast Cancer Histology Images. *coRR*, abs/1808.04277.

Athalye, A., Carlini, N. & Wagner, D. (2018). Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *Proceedings of the 35th International Conference on Machine Learning*, 80, 274–283.

Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Bamford, P. & Lovell, B. (2001). Method for accurate unsupervised cell nucleus segmentation. *Conf. of the Intern. Conf. of the IEEE Engineering in Medicine and Biology Society*.

Bankman, I. (2008). *Handbook of medical image processing and analysis*. Elsevier.

Barron, J. T. (2019). A general and adaptive robust loss function. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339.

Bartels, P., Thompson, D., Bibbo, M. et al. (1992). Bayesian belief networks in quantitative histopathology. *Analytical and quantitative cytology and histology*, 14.

Basavanhally, A., Agner, S., Alexe, G. et al. (2008). Manifold learning with graph-based features for identifying extent of lymphocytic infiltration from high grade, her2+ breast cancer histology.

Belharbi, S., Rony, J., Dolz, J., Ben Ayed, I., McCaffrey, L. & Granger, E. (2019). Weakly Supervised Object Localization using Min-Max Entropy: an Interpretable Framework. *coRR*, abs/1907.12934.

Bency, A. J., Kwon, H., Lee, H. et al. (2016). Weakly supervised localization using deep feature maps. *ECCV*.

Bianco, S., Cadene, R., Celona, L. & Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6, 64270–64277.

Biggio, B. & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84, 317–331. doi: 10.1016/j.patcog.2018.07.023.

Bilen, H. & Vedaldi, A. (2016). Weakly supervised deep detection networks. *CVPR*.

Bilen, H., Pedersoli, M. & Tuytelaars, T. (2014). Weakly supervised object detection with posterior regularization. *BMVC*.

Bilen, H., Pedersoli, M. & Tuytelaars, T. (2015). Weakly supervised object detection with convex clustering. *CVPR*.

Bilgin, C., Demir, C., Nagi, C. et al. (2007). Cell-graph mining for breast tissue modeling and classification. *Intern. Conf. of the IEEE Engineering in Medicine and Biology Society*.

Black, M. J. & Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1), 75–104.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Brendel, W., Rauber, J., Kurakin, A., Papernot, N., Veliqi, B., Salathé, M., Mohanty, S. P. & Bethge, M. (2018). Adversarial Vision Challenge. *arXiv:1808.01976*.

Brendel, W., Rauber, J., Kurakin, A., Papernot, N., Veliqi, B., Mohanty, S. P., Laurent, F., Salathé, M., Bethge, M., Yu, Y. et al. (2020). Adversarial vision challenge. In *The NeurIPS'18 Competition* (pp. 129–153). Springer.

Buckman, J., Roy, A., Raffel, C. & Goodfellow, I. (2018). Thermometer Encoding: One Hot Way To Resist Adversarial Examples. *International Conference on Learning Representations*.

Bándi, P., Geessink, O., Manson, Q. et al. (2019). From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge. *IEEE Transactions on Medical Imaging*, 38.

Caicedo, J. C., González, F. A. & Romero, E. (2011). Content-based histopathology image retrieval using a kernel-based semantic annotation framework. *Jour. of biomedical informatics*, 44.

Caie, P. D., Turnbull, A. K., Farrington, S. M. et al. (2014). Quantification of tumour budding, lymphatic vessel density and invasion through image analysis in colorectal cancer. *Journal of translational medicine*, 12.

Cao, C., Liu, X., Yang, Y. et al. (2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *ICCV*.

Carbonneau, M.-A., Cheplygina, V., Granger, E. & Gagnon, G. (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77, 329–353.

Carlini, N. & Wagner, D. (2017). Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy (SP)*, pp. 39–57.

Chandler, D. E. & Roberson, R. W. (2009). *Bioimaging: current concepts in light and electron microscopy*. Jones & Bartlett Publishers.

Charbonnier, P., Blanc-Feraud, L., Aubert, G. & Barlaud, M. (1994). Two deterministic half-quadratic regularization algorithms for computed imaging. *Proceedings of 1st International Conference on Image Processing*, 2, 168–172.

Chattopadhyay, A., Sarkar, A., Howlader, P. et al. (2018). Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. *WACV*.

Chen, H., Qi, X., Yu, L. et al. (2016). DCAN: deep contour-aware networks for accurate gland segmentation. *CVPR*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. (2018a). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. (2018b). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *ECCV*.

Cheplygina, V., de Bruijne, M. & Pluim, J. (2019). Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *MIA*, 54.

Cinbis, R. G., Verbeek, J. & Schmid, C. (2017). Weakly supervised object localization with multi-fold multiple instance learning. *IEEE trans. on pattern analysis and machine intelligence*, 39.

Ciompi, F., Geessink, O., Bejnordi, B. E. et al. (2017). The importance of stain normalization in colorectal tissue classification with convolutional networks. *Biomedical Imaging*.

Cireşan, D. C., Giusti, A., Gambardella, L. M. et al. (2013). Mitosis detection in breast cancer histology images with deep neural networks. *MICCAI*.

Cooper, L., Sertel, O., Kong, J. et al. (2009). Feature-based registration of histopathology images with different stains: An application for computerized follicular lymphoma prognosis.

Computer methods and programs in biomedicine, 96.

Courtiol, P., Tramel, E. W., Sanselme, M. et al. (2018). Classification and Disease Localization in Histopathology Using Only Global Labels: A Weakly-Supervised Approach. *coRR*, abs/1802.02212.

Couture, H. D., Marron, J., Perou, C. M. et al. (2018). Multiple Instance Learning for Heterogeneous Images: Training a CNN for Histopathology. *coRR*, abs/1806.05083.

Daisuke, K. & Shumpei, I. (2018). Machine Learning Methods for Histopathological Image Analysis. *Computational and Structural Biotechnology Journal*, 16.

Deng, J., Dong, W., Socher, R. et al. (2009). ImageNet: A large-scale hierarchical image database. *CVPR*.

Dennis Jr, J. E. & Welsch, R. E. (1978). Techniques for nonlinear least squares and robust regression. *Communications in Statistics-Simulation and Computation*, 7(4), 345–359.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A. & Anandkumar, A. (2018). Stochastic Activation Pruning for Robust Adversarial Defense. *International Conference on Learning Representations*.

Diba, A., Sharma, V., Pazandeh, A. M. et al. (2017). Weakly Supervised Cascaded Convolutional Networks. *CVPR*.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26.

Dolz, J., Desrosiers, C. & Ben Ayed, I. (2018). 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*.

Doshi-Velez, F. & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *coRR*, abs/1702.08608.

Doyle, S., Madabhushi, A., Feldman, M. et al. (2006a). A boosting cascade for automated detection of prostate cancer from digitized histology. *MICCAI*.

Doyle, S., Rodriguez, C., Madabhushi, A. et al. (2006b). Detecting prostatic adenocarcinoma from digitized histology using a multi-scale hierarchical classification approach. *Intern. Conf. of the IEEE Engineering in Medicine and Biology Society*.

Doyle, S., Hwang, M., Shah, K. et al. (2007). Automated grading of prostate cancer using architectural and textural image features. *Intern. Symposium on Biomedical Imaging: From Nano to Macro*.

Doyle, S., Agner, S., Madabhushi, A. et al. (2008). Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features. *Intern. Symposium on Biomedical Imaging: From Nano to Macro*.

Durand, T., Thome, N. & Cord, M. (2015). MANTRA: Minimum Maximum Latent Structural SVM for Image Classification and Ranking. *ICCV*.

Durand, T., Thome, N. & Cord, M. (2016). Weldon: Weakly supervised learning of deep convolutional neural networks. *CVPR*.

Durand, T., Mordan, T., Thome, N. et al. (2017). Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. *CVPR*.

Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P. et al. (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast CancerMachine Learning Detection of Breast Cancer Lymph Node MetastasesMachine

Learning Detection of Breast Cancer Lymph Node Metastases. *JAMA*, 318.

Everingham, M., Van Gool, L., Williams, C. K. I. et al. (2010a). The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010b). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.

Feng, X., Yang, J., Laine, A. F. et al. (2017). Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules. *MICCAI*.

Frenay, B. & Verleysen, M. (2014). Classification in the Presence of Label Noise: A Survey. *TNNLS*, 25.

Ganan, S. & McClure, D. (1985). Bayesian image analysis: An application to single photon emission tomography. *Amer. Statist. Assoc*, 12–18.

Gartner, L. P. & Hiatt, J. L. (2006). Color Textbook of Histology: with Student Consult Online Access.

Ge, W., Yang, S. & Yu, Y. (2018). Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning. *CVPR*.

Geng, X. (2016). Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7), 1734–1748.

Gertych, A., Ing, N., Ma, Z. et al. (2015). Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 46.

Girshick, R. (2015). Fast r-cnn. *ICCV*.

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Gondal, W. M., Köhler, J. M., Grzeszick, R. et al. (2017). Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. *ICIP*.

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

Goodfellow, I. J., Shlens, J. & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations*.

Greenberg, S. D. (1984). *Computer-assisted image analysis cytology*. Karger, S Publishers.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377. doi: 10.1016/j.patcog.2017.10.013.

Guillaud, M., Cox, D., Malpica, A. et al. (2004). Quantitative histopathological analysis of cervical intra-epithelial neoplasia sections: methodological issues. 26.

Guillaud, M., Adler-Storthz, K., Malpica, A. et al. (2005). Subvisual chromatin changes in cervical epithelium measured by texture image analysis and correlated with HPV. *Gynecologic oncology*, 99.

Guo, C., Rana, M., Cissé, M. & van der Maaten, L. (2018). Countering Adversarial Images using Input Transformations. *International Conference on Learning Representations*.

Gurcan, M., Pan, T., Shimada, H. et al. (2006). Image analysis for neuroblastoma classification: Hysteresis thresholding for cell segmentation.

Gurcan, M. N., Boucheron, L., Can, A. et al. (2009). Histopathological image analysis: A review. *IEEE reviews in biomedical engineering*, 2.

Hamilton, P., Anderson, N., Bartels, P. et al. (1994). Expert system support using Bayesian belief networks in the diagnosis of fine needle aspiration biopsy specimens of the breast. *Journal of clinical pathology*, 47.

Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S. & Malik, J. (2011). Semantic Contours from Inverse Detectors. *International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S. et al. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *ECCV*.

He, K., Zhang, X., Ren, S. et al. (2016). Deep residual learning for image recognition. *CVPR*.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*.

He, L. (2009). *Towards image understanding: deformable contour and graph-based image segmentation and recognition*. Lambert Academic Publishing.

He, L., Long, L. R., Antani, S. et al. (2010). Computer assisted diagnosis in histopathology. *Sequence and genome analysis: methods and applications*, 15.

He, L., Long, L. R., Antani, S. et al. (2012). Histology image analysis for carcinoma detection and grading. *Computer methods and programs in biomedicine*, 107.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J. & Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 558–567.

Hipp, J. D., Fernandez, A., Compton, C. C. et al. (2011). Why a pathology image should not be considered as a radiology image. *Journal of pathology informatics*, 2.

Hou, L., Samaras, D., Kurc, T. M. et al. (2016). Patch-based convolutional neural network for whole slide tissue image classification. *CVPR*.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1), 73–101.

Ilse, M., Tomczak, J. M. & Welling, M. (2018). Attention-based deep multiple instance learning. *coRR*, abs/1802.04712.

Izadyyazdanabadi, M., Belykh, E., Cavallo, C. et al. (2018). Weakly-Supervised Learning-Based Feature Localization in Confocal Laser Endomicroscopy Glioma Images. *coRR*, abs/1804.09428.

Janowczyk, A. & Madabhushi, A. (2016). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Jour. of pathology informatics*, 7.

Janowczyk, A., Basavanhally, A. & Madabhushi, A. (2017). Stain normalization using sparse autoencoders (StaNoSA): Application to digital pathology. *Computerized Medical Imaging and Graphics*, 57.

Jensen, P. A. & Bard, J. F. a. (2003). *Operations Research Models and Methods*. Wiley.

Jia, Z., Huang, X., Eric, I., Chang, C. & Xu, Y. (2017). Constrained deep weak supervision for histopathology image segmentation. *IEEE transactions on medical imaging*, 36(11), 2376–2388.

Jie, Z., Wei, Y., Jin, X. et al. (2017). Deep Self-Taught Learning for Weakly Supervised Object Localization. *CVPR*.

Jütting, U., Gais, P., Rodenacker, K. et al. (1999). Diagnosis and prognosis of neuroendocrine tumours of the lung by means of high resolution image analysis. *Analytical Cellular Pathology*, 18.

Kandemir, M. & Hamprecht, F. (2015). Computer-aided diagnosis from weak supervision: A benchmarking study. *Computerized medical imaging and graphics*, 42.

Kantorov, V., Oquab, M., Cho, M. et al. (2016). Contextlocnet: Context-aware deep network models for weakly supervised localization. *ECCV*.

Kayser, G., Riede, U., Werner, M. et al. (2002). Towards an automated morphological classification of histological images of common lung carcinomas. *Elec J Pathol Histol*, 8.

Kervadec, H., Dolz, J., Tang, M., Granger, E., Boykov, Y. & Ayed, I. B. (2019). Constrained-CNN losses for weakly supervised segmentation. *Medical image analysis*, 54, 88–99.

Kieffer, B., Babaie, M., Kalra, S. et al. (2017). Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. *Intern. Conf. on Image Processing Theory, Tools and Applications*.

Kiernan, J. A. (1990). *Histological and histochemical methods: theory and practice*. Pergamon press Oxford.

Kong, J., Sertel, O., Lozanski, G. et al. (2007a). Automated detection of follicular centers for follicular lymphoma grading. *Advancing Practice, Instruction, and Innovation Through Informatics Conf.*

Kong, J., Shimada, H., Boyer, K. et al. (2007b). Image analysis for automated assessment of grade of neuroblastic differentiation. *Intern. Symposium on Biomedical Imaging: From Nano to Macro*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

Kurakin, A., Goodfellow, I. & Bengio, S. (2017). Adversarial examples in the physical world. *International Conference on Learning Representations (workshop track)*.

Lamprecht, M. R., Sabatini, D. M. & Carpenter, A. E. (2007). CellProfiler™: free, versatile software for automated biological image analysis. *Biotechniques*, 42.

Lee, J., Kim, E., Lee, S., Lee, J. & Yoon, S. (2019). FickleNet: Weakly and Semi-Supervised Semantic Image Segmentation Using Stochastic Inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5267–5276.

Li, Y. & Ping, W. (2018). Cancer Metastasis Detection With Neural Conditional Random Field. *Medical Imaging with Deep Learning*.

Lin, M., Chen, Q. & Yan, S. (2013). Network in network. *coRR*, abs/1312.4400.

Lin, T.-Y., Maire, M., Belongie, S. et al. (2014). Microsoft COCO: Common Objects in Context. *ECCV*.

Lipton, Z. C. (2018). The Mythos of Model Interpretability. *Queue*, 16.

Litjens, G., Kooi, T., Bejnordi, B. E. et al. (2017). A survey on deep learning in medical image analysis. *MIA*.

Liu, W., Anguelov, D., Erhan, D. et al. (2016). Ssd: Single shot multibox detector. *ECCV*.

Madabhushi, A. (2009). Digital pathology image analysis: opportunities and challenges. *Imaging in medicine*, 1.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. *International Conference on Learning Representations*.

Marcus, G. (2001). *The algebraic mind: Integrating connectionism and cognitive science*. MIT Press.

Marcus, G. (2018). Deep Learning: A Critical Appraisal. *CoRR*, abs/1801.00631.

Meijer, G., Beliën, J., Van Diest, P. et al. (1997). Origins of ... image analysis in clinical pathology. *Journal of clinical pathology*, 50.

Méndez, A. J., Tahoces, P. G., Lado, M. J. et al. (1998). Computer-aided diagnosis: Automatic detection of malignant masses in digitized mammograms. *Medical Physics*, 25.

Mescher, A. L. (2013). *Junqueira's basic histology: text and atlas*. McGraw-Hill Medical.

Molnar, C. et al. (2018). Interpretable machine learning: A guide for making black box models explainable.

Moosavi-Dezfooli, S.-M., Fawzi, A. & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582.

Mungle, T., Tewary, S., Das, D. et al. (2017). MRF-ANN: a machine learning approach for automated ER scoring of breast cancer immunohistochemical images. *Journal of microscopy*, 267.

Murphy, D. B. & Davidson, M. W. (2001). *Fundamentals of light microscopy and electronic imaging*. Wiley Online Library.

Naik, S., Doyle, S., Madabhushi, A. et al. (2007). Automated Gland Segmentation and Gleason Grading of Prostate Histology by Integrating Low-, High-level and Domain Specific Information. *Workshop on Microscopic Image Analysis with Applications in Biology*.

Nelson, D. L., Lehninger, A. L. & Cox, M. M. (2008). *Lehninger principles of biochemistry*. Macmillan.

O'Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group.

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

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9.

Papernot, N., McDaniel, P., Wu, X., Jha, S. & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. *IEEE Symposium on Security and Privacy (SP)*, pp. 582–597.

Petushi, S., Garcia, F. U., Haber, M. M. et al. (2006). Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC medical imaging*, 6.

Pinheiro, P. H. O. & Collobert, R. (2015a). From image-level to pixel-level labeling with Convolutional Networks. *CVPR*.

Pinheiro, P. O. & Collobert, R. (2015b). From image-level to pixel-level labeling with convolutional networks. *CVPR*.

Quellec, G., Cazuguel, G., Cochener, B. & Lamard, M. (2017). Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*, 10.

Qureshi, H., Sertel, O., Rajpoot, N. et al. (2008). Adaptive discriminant wavelet packet transform and local binary patterns for meningioma subtype classification. *MICCAI*.

Rakelly, K., Shelhamer, E., Darrell, T. et al. (2018). Few-Shot Segmentation Propagation with Guided Networks. *CoRR*, abs/1806.07373.

Rauber, J., Brendel, W. & Bethge, M. (2017). Foolbox: A Python toolbox to benchmark the robustness of machine learning models. *arXiv:1707.04131*.

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

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *ACM SIGKDD 2016*.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI*.

Ross, M., Kaye, G. & Pawlina, W. (2003). Histology a Text and Atlas. Lippincott Williams&Wilkins, Philadelphia.

Roux, L., Racoceanu, D., Loménie, N. et al. (2013). Mitosis detection in breast cancer histological images An ICPR 2012 contest. *Jour. of pathology informatics*, 4.

Samek, W., Wiegand, T. & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *coRR*, abs/1708.08296.

Schroeder, W., Ng, L. & Cates, J. (2003). The ITK software guide.

Sedai, S., Mahapatra, D., Ge, Z. et al. (2018). Deep multiscale convolutional feature learning for weakly supervised localization of chest pathologies in X-ray images. *Intern. Workshop on Machine Learning in Medical Imaging*.

Selvaraju, R. R., Cogswell, M., Das, A. et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *ICCV*.

Settles, B. (2009). *Active learning literature survey*.

Shah, M., Wang, D., Rubadue, C. et al. (2017). Deep learning assessment of tumor proliferation in breast cancer histological images. *Intern. Conf. on Bioinformatics and Biomedicine*.

Sheikhzadeh, F., Guillaud, M. & Ward, R. K. (2016). Automatic labeling of molecular biomarkers of whole slide immunohistochemistry images using fully convolutional networks. *coRR*, abs/1612.09420.

Shen, Y., Ji, R., Zhang, S. et al. (2018). Generative Adversarial Learning Towards Fast Weakly Supervised Detection. *CVPR*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90.

Sirinukunwattana, K., Pluim, J. P., Chen, H. et al. (2017). Gland segmentation in colon histology images: The glas challenge contest. *MIA*, 35.

Spanhol, F. A., Oliveira, L. S., Petitjean, C. et al. (2016a). Breast cancer histopathological image classification using Convolutional Neural Networks. *IJCNN*.

Spanhol, F. A., Oliveira, L. S., Petitjean, C. et al. (2016b). A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Trans. on Biomedical Engineering*, 63.

Sternberg, S. (1997). *Histology for pathologists*. Lippincott-Raven Philadelphia.

Su, W., Yuan, Y. & Zhu, M. (2015). A Relationship Between the Average Precision and the Area Under the ROC Curve. *Int. Conf. on The Theory of Information Retrieval*.

Sudharshan, P., Petitjean, C., Spanhol, F. et al. (2019). Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications*, 117.

Sugiyama, M. & Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Sukhbaatar, S., Bruna, J., Paluri, M. et al. (2014). Training convolutional networks with noisy labels. *coRR*, abs/1406.2080.

Sun, C., Paluri, M., Collobert, R. et al. (2016). Pronet: Learning to propose object-specific boxes for cascaded neural networks. *CVPR*.

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W. & Wang, J. (2019). High-Resolution Representations for Labeling Pixels and Regions. *CoRR*, abs/1904.04514.

Szegedy, C., Zaremba, W., Sutskever, I. et al. (2013). Intriguing properties of neural networks. *coRR*, abs/1312.6199.

Szegedy, C., Vanhoucke, V., Ioffe, S. et al. (2016a). Rethinking the inception architecture for computer vision. *CVPR*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. & Fergus, R. (2014). Intriguing properties of neural networks. *International Conference on Learning Representations*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016b). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Tabesh, A., Teverovskiy, M., Pang, H.-Y. et al. (2007). Multifeature prostate cancer diagnosis and Gleason grading of histological images. *IEEE trans. on medical imaging*, 26.

Tang, J., Rangayyan, R. M., Xu, J. et al. (2009). Computer-aided detection and diagnosis of breast cancer with mammography: recent advances. *IEEE Trans. on Information Technology in Biomedicine*, 13.

Tang, P., Wang, X., Bai, X. et al. (2017). Multiple instance detection network with online instance classifier refinement. *CVPR*.

Tang, P., Wang, X., Wang, A. et al. (2018). Weakly Supervised Region Proposal Network and Object Detection. *ECCV*.

Teh, E. W., Rochan, M. & Wang, Y. (2016). Attention Networks for Weakly Supervised Object Localization. *BMVC*.

Török, P. & Kao, F. (2007). *Optical imaging and microscopy: techniques and advanced systems*. Springer.

Tramèr, F., Kurakin, A., Papernot, N., Boneh, D. & McDaniel, P. (2018). Ensemble Adversarial Training: Attacks and Defenses. *International Conference on Learning Representations*.

Uijlings, J. R., Van De Sande, K. E., Gevers, T. et al. (2013). Selective search for object recognition. *IJCV*, 104.

Van de Sande, K. E., Uijlings, J. R., Gevers, T. et al. (2011). Segmentation as selective search for object recognition. *ICCV*.

Veta, M., Pluim, J., Van Diest, P. J. et al. (2014). Breast cancer histopathology image analysis: A review. *IEEE Transactions on Biomedical Engineering*, 61.

Veta, M., Heng, Y. J., Stathonikos, N. et al. (2018). Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge. *coRR*, abs/1807.08284.

Wan, F., Wei, P., Jiao, J. et al. (2018). Min-Entropy Latent Model for Weakly Supervised Object Detection. *CVPR*.

Wang, C., Ren, W., Huang, K. et al. (2014). Weakly supervised object localization with latent category learning. *ECCV*.

Wang, D., Foran, D. J., Ren, J. et al. (2015). Exploring automatic prostate histopathology image gleason grading via local structure modeling. *Intern. Conf. the IEEE Engineering in Medicine and Biology Society*.

Wang, X., Yan, Y., Tang, P., Bai, X. & Liu, W. (2018). Revisiting multiple instance neural networks. *Pattern Recognition*, 74.

Wang, Y. & Yao, Q. (2019). Few-shot Learning: A Survey. *CoRR*, abs/1904.05046.

Weind, K. L., Maier, C. F., Rutt, B. K. et al. (1998). Invasive carcinomas and fibroadenomas of the breast: comparison of microvessel distributions–implications for imaging modalities. *Radiology*, 208.

Wootton, R., Springall, D., Polak, J. M. et al. (1995). *Image analysis in histology*. CUP Archive.

Xie, C., Wang, J., Zhang, Z., Ren, Z. & Yuille, A. (2018). Mitigating Adversarial Effects Through Randomization. *International Conference on Learning Representations*.

Xie, J., Liu, R., Joseph Luttrell, I. et al. (2019). Deep Learning Based Analysis of Histopathological Images of Breast Cancer. *Frontiers in genetics*.

Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500.

Xu, J., Luo, X., Wang, G. et al. (2016). A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing*, 191.

Yoo, T. S. (2004). *Insight into images: principles and practice for segmentation, registration, and image analysis*. AK Peters/CRC Press.

Zagoruyko, S. & Komodakis, N. (2016). Wide Residual Networks. *Proceedings of the British Machine Vision Conference*, pp. 87.1-87.12. doi: 10.5244/C.30.87.

Zhang, C., Bengio, S., Hardt, M. et al. (2017). Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530.

Zhang, J., Bargal, S. A., Lin, Z. et al. (2018a). Top-down neural attention by excitation backprop. *IJCV*, 126.

Zhang, Q.-C. & Zhu, S.-C. (2018). Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19, 27–39.

Zhang, X., Wei, Y., Feng, J. et al. (2018b). Adversarial complementary learning for weakly supervised object localization. *CVPR*.

Zhang, Z. (1995). *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting*.

Zhou, B., Khosla, A., Lapedriza, A. et al. (2016). Learning deep features for discriminative localization. *CVPR*.

Zhou, Z.-H. (2004). Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep*.

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

Zhu, Y., Zhou, Y., Ye, Q. et al. (2017). Soft proposal networks for weakly supervised object localization. *ICCV*.

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