Semi-supervised Learning with Imbalanced Class in Medical Image

A thesis submitted in total fulfilment of the requirements for the degree of

Master of Science

By

Thanh Tri Huynh

Bachelor of Information Technology (Honours)

Department of Computer Science and Information Technology
School of Engineering and Mathematical Sciences
College of Science, Health and Engineering

La Trobe University
Victoria, Australia

August 2021

Contents

List of Figures

List of Tables

Abstract

Medical image classification and segmentation are often challenging for two reasons: a lack of labelled examples due to expensive and time-consuming annotation protocols; imbalanced class labels due to the relative scarcity of disease-positive individuals in the wider population for the classification task and a larger amount of background pixels compared to foreground pixels for the segmentation task. Existing semi-supervised learning (SSL) methods can take advantage of unlabelled data to improve performance, but they generally do not address the problem of class imbalance. Hence in this thesis we propose Adaptive Blended Consistency Loss (ABCL), a drop-in replacement for consistency loss in perturbation-based SSL methods. ABCL counteracts data skew by adaptively mixing the target class distribution of the consistency loss between the prediction made from the original versus augmented data samples. Our experiments involving two medical image classification and two segmentation datasets reveal ABCL consistently outperforms existing state-of-art semi-supervised learning algorithms including one which is designed to address the class imbalance problem.

Statement of Authorship

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgment in the main text of the thesis. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

Thanh Tri Huynh
28 August 2021

Acknowledgements

This research work was supported by a La Trobe Graduate Research Scholarship stipend and a La Trobe University Full-Fee Research Scholarship.

First and foremost, I would like to thank all those who have supported me to accomplish my thesis. I would especially like to express my deep and sincere gratitude to my research supervisor Assoc. Prof Zhen He for his tireless support and guidance throughout my thesis. With his great supervision and extensive knowledge, Zhen helped me to tackle troubles to accomplish my thesis. Furthermore, I also would like to sincerely thank my research co-supervisor Dr. Aiden Nibali for providing new ideas and concepts to explore with his expertise in this research domain. Finally, it would not be possible to accomplish my thesis without both of you.

Publications

This thesis contains work by the author that has been preparing for publication as follows:

- Tri Huynh, Aiden Nibali and Zhen He, "Semi-supervised learning for medical image classification using imbalanced training data", Submit to Computer Methods and Programs in Biomedicine Journal.

Chapter 1

Introduction

In recent years, computer aided diagnosis whose common underlying technology is machine learning has emerged as one of essential and promising tools in medical treatment. By automatically analysing various medical imaging modalities, machine learning has assisted doctors to diagnose diseases more efficiently. The two main tasks for computer aided diagnosis are segmentation and classification. At the early stages of a disease, tumors or lesions may be very small on patient scans and therefore easy for a radiologist or doctor to miss, hence a computer assisted image segmentation method can be used to highlight any tumors or lesions found. Medical classification techniques can automatically analyse and output a disease diagnosis from a patient's scans.

Recently, many supervised machine learning methods have achieved promising results in medical image classification [2, 3, 4, 5, 6, 7, 8] and in medical image segmentation [9, 10, 11, 12]. However supervised learning requires a substantial amount of fully labelled medical images, while in reality there may be many unlabelled images that can be used to boost accuracy. Therefore semi-supervised learning is an ideal method to make use of all scanned medical images including labelled and unlabelled samples. There is another common characteristic of medical image datasets. Medical image datasets often have very skewed data distributions such as a large number of negative disease cases versus a small number of positive disease cases. In terms of medical segmentation datasets, the proportion of pixels belonging to the disease of interest (e.g. tumor) is often substantially larger than the background (e.g. normal tissue). Therefore, It is critically important to address this class imbalance problem for medical images. The reason is if we don't develop methods to tackle this problem then the minor class (often the disease class) may be missed leading to potential fatal consequences. Although there are several studies focused on semi-supervised learning [13, 14, 15, 16] for medical images classification and [17, 18, 19, 20, 21] for medical image segmentation, none of these studies focused on addressing the class imbalance problem. Hence, we address this gap in the literature by studying explicitly the class imbalance problem for semi-supervised learning (SSL) for medical image classifica-

tion and semantic segmentation. For classification we conduct our experiments using the HAM10000 skin cancer dataset [22] and the retinal fundus REFUGE Challenge dataset [23]. For semantic segmentation we use the Nerve Ultrasound dataset [24] and Breast Cancer Ultrasound dataset [25]. Our approach is designed to work with any image datasets which have imbalanced class distributions.

There are many different types of semi-supervised learning algorithms. However, recently all state of the art semi-supervised learning algorithms [1, 26, 27, 28, 29] are perturbation based. Most perturbation based algorithms augment unlabelled data and then apply a consistency loss to make the predicted class distribution from the original and augmented unlabelled samples similar. We focus our study on modifying the consistency loss on one of the best performing perturbation based SSLs, called Unsupervised Data Augmentation (UDA) [1]. However, it is important to note the method we developed to address class imbalance can be used with any perturbation based SSL method that uses consistency loss.

The standard consistency loss (CL) used in UDA has two shortcomings. 1) It degrades the classification performance of the minor classes and 2) it always sets the target as the original example instead of a blend of the augmented and original examples. Hence, we propose the Adaptive Blended Consistency Loss (ABCL) method which tackles both shortcomings of standard CL by generating a target class distribution which is a blend of the original and augmented class distributions. The blended target class distribution skews towards either the original or augmented example depending on which predicted the minor class. In this thesis we apply our ABCL method on both medical image classification and semantic segmentation. Semantic segmentation requires each pixel (2D) or voxel (3D) of an image to be classified. We apply our ABCL method on the grain pixel or voxel rather than the whole image.

We performed extensive experiments comparing our ABCL method against rival methods on both medical image classification and segmentation datasets. For the HAM10000 skin cancer classification dataset ABCL achieved an unweighted average recall (UAR) of 0.67 versus 0.59 for the baseline implementation of UDA. Furthermore, ABCL also significantly outperformed the state-of-the-art Suppressed Consistency Loss (SCL) [82] method for the same dataset. For the retinal fundus glaucoma classification dataset, ABCL significantly outperformed SCL, increasing UAR from 0.57 to 0.67. For the segmentation problem, ABCL outperforms the baseline measured by the Dice Coefficient score on both the Nerve Ultrasound and Breast Cancer Ultrasound datasets. These results show that ABCL is able to consistently improve performance of consistency loss based SSL methods for class imbalanced classification and segmentation tasks. This thesis makes the following key contributions:

- We identify the importance of handling class imbalance for semi-supervised classification and segmentation of medical images. In contrast, no existing work explicitly addresses this problem for medical images.

- We propose the Adaptive Blended Consistency Loss (ABCL) as a replacement for the consistency loss of perturbation based SSL algorithms such as UDA and Mean Teacher [26] in order to tackle the class imbalance problem in SSL.
- We conduct extensive experiments on two classification and two segmentation medical image datasets to demonstrate the advantages of ABCL over standard consistency loss and other existing methods designed for addressing the class imbalance problem in both supervised learning and SSL.

1.1 Thesis structure

The structure of the thesis is as follows:

- **Chapter 2 - Background:** This chapter will provide the background knowledge required to understand the existing work and the contributions of this thesis.
- **Chapter 3 - Related Works:** This chapter will review related works in the area of semi-supervised classification and segmentation for general and medical imaging tasks, works relating to using Convolutional Neural Networks for medical image classification.
- **Chapter 4 - Problem Definition:** This chapter will define the problem studied in this thesis.
- **Chapter 5 - Methodology:** This chapter will describe proposed approaches to handle the problem.
- **Chapter 6 - Experiment Setup:** This chapter will list all materials needed in experiments and describe how experiments are implemented.
- **Chapter 7 - Experiment Results:** This chapter will show the results obtained and also evaluate and analyse those results.
- **Chapter 8 - Conclusion:** This chapter will summarise the achievement in this thesis and propose ideas for future works.

Chapter 2

Background

2.1 Supervised versus unsupervised learning

In the era of the digital world, machine learning has been playing a crucial role across many different industries and research fields such as medical research, biology, robotics, manufacturing, etc. Machine learning has been used to solve numerous problems and achieved near human accuracy. Moreover, beyond the limitation of computation and amount of data, machine learning becomes a more powerful technique than ever. Machine learning is a subset of artificial intelligence which trains models using data rather than explicit instructions in order to map inputs to outputs or make decisions [30]. Machine learning is mainly divided into three branches: supervised learning, unsupervised learning and reinforcement learning. To confine the scope of this literature survey, we will only discuss supervised and unsupervised learning. In supervised learning, a learnt model takes input data and predicts the output. The model learns the mapping from the input to the output by comparing the predicted output against the desired output. Importantly, the key requirement of supervised learning is that for every training sample, an input and desired output (known as the label) pair is provided. So, to make supervised models perform well, it is necessary to gather a large amount of labelled data. Moreover, two main categories of supervised learning are classification and regression. For classification, the output of the model should be a categorical class label. For regression, the goal is to output a numerical value. Another important branch of machine learning is unsupervised learning, which instead of knowing the explicit output label, it only receives the input data. The two important techniques of unsupervised learning are clustering and dimensionality reduction. The common goal of clustering is to find the pattern of a set of data points and cluster them into a group. Dimensionality reduction is the process of reducing the number of features or variables, also known as the dimension, in a dataset without impacting on the meaning of the data.

2.2 Semi-supervised learning

The brief review of the two popular paradigms of both supervised and unsupervised learning exposes different shortcomings, for supervised learning it is expensive to obtain a large amounts of labelled data and for unsupervised learning it is hard to extract the meaningful information from unlabelled data. Thus, one promising paradigm that overcomes these obstacles is semi-supervised learning (SSL) [31] which makes use of a small amount of labelled and a large amount of unlabelled data to optimize the training of models. The goal of semi-supervised learning is to learn patterns from unlabelled data and augment the prediction of the model by these patterns. More intuitively, by leveraging the unlabelled data, the model produces a better decision boundary for different classes which reflects the data's underlying structure [32]. Over the history of machine learning, researchers have invented various approaches to semi-supervised learning which mostly refer to either transductive learning or inductive learning. Specifically, given the data set of labelled (x_L,y_L) and unlabelled x_U data, the objective of the inductive learner is to make the prediction for any data points, whereas the transductive learner will only produce the label for unlabelled data points. In the broad perspective, the inductive semi-supervised learning generalise a model that can output the label for unseen instances, on the other hand, transductive semi-supervised learning is only able to make the label for unlabelled data, known instances, during training phase [33]. Figure 2.1

2.3 Convolutional Neural Network for image classification

Computer vision is one of the most important domains for the application of machine learning and includes various tasks such as object detection [34] or image classification [35], etc. Recently, many researchers have achieved state of the art results for these tasks on public datasets such as ImageNet [36], CIFAR 10 or CIFAR 100 [37]. To achieve such successful results, almost all of them have used one breakthrough type of neural networks, called Convolutional Neural Networks (CNNs) [38, 39]. This type of model has the ability to help the machine understand the representation of image better and also automatically learn features from the image.

Specifically, the architecture of CNNs is designed as a series of blocks and followed by a classification module [41]. Each block contains three layers: filter bank layer, non-linearity layer and pooling layer. In addition, the main characteristic of CNNs is the feature map which is the input and output of layers in each block. For example in Figure 2.2, the very first layer receives the RGB image of a dog and outputs features of the image such as eyes, ears, etc. The filter bank layer is a set of filters, which are

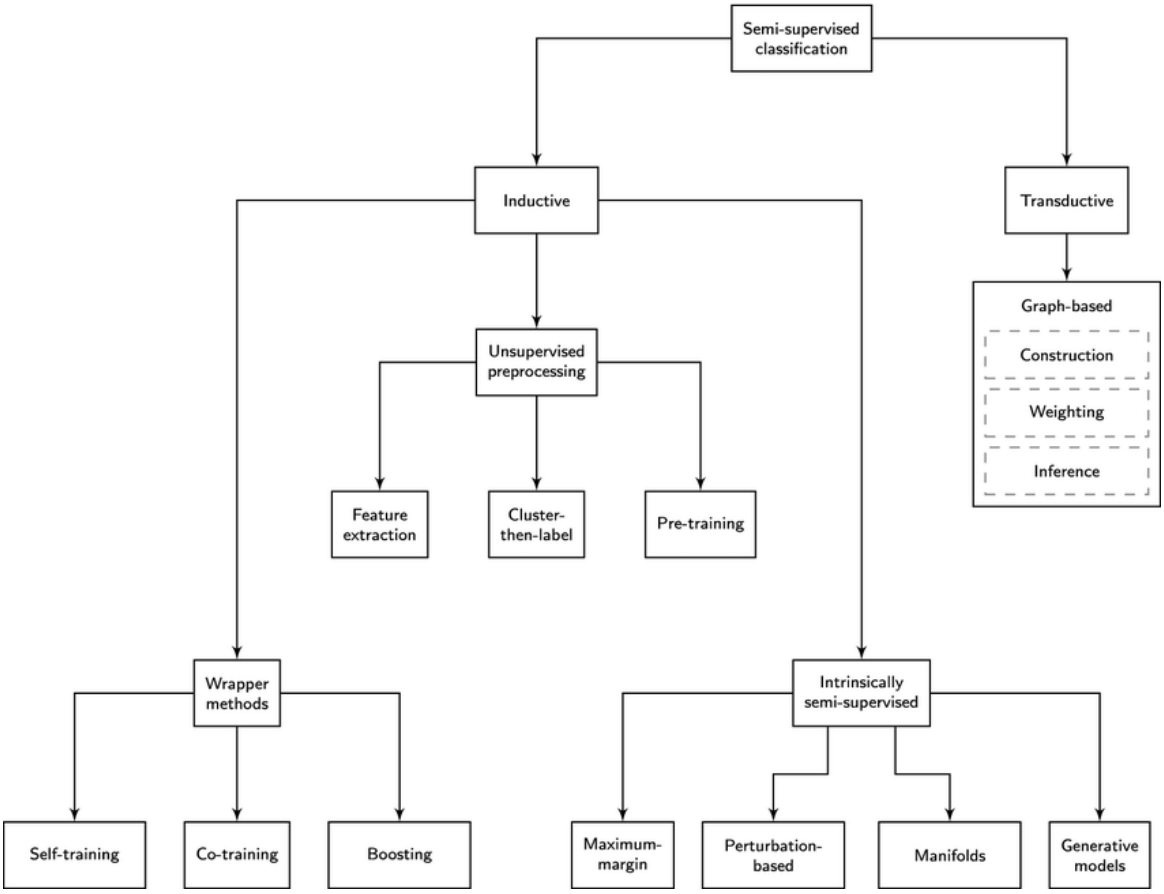


Figure 2.1: Visualization of the semi-supervised taxonomy [33].

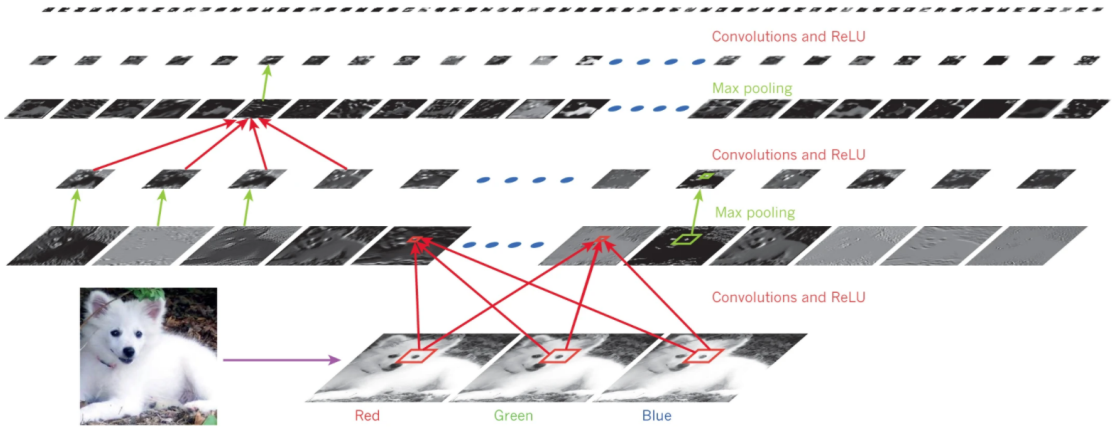


Figure 2.2: Inside Convolutional Neural Networks [40].

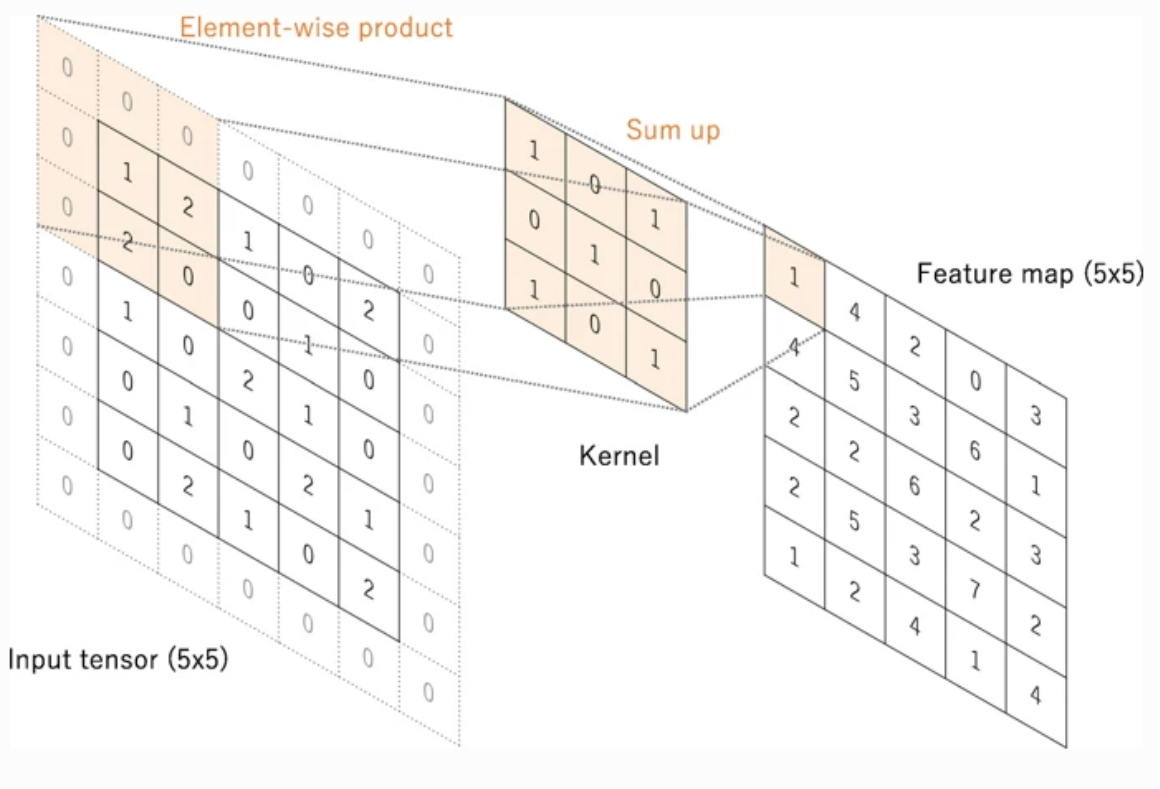


Figure 2.3: The example of filter over the image [42].

matrices of weights having dimensions (width,height) or (width,height,depth) corresponding to input of black and white image or color image respectively. Each filter scans through the image and makes the sum of element-wise operation with each scanned portion of the image. Intuitively, the filter is a window, which slides horizontally from top left to bottom right of the image and returns the feature map output, as shown in Figure 2.3. In addition, the filter often is called a feature detector and its weights sometimes is predefined to reflect what features it wants to detect such as vertical or horizontal edge.

Moreover, two important settings of filter banks are padding and stride. Padding is the process of adding zero pixels around the border of an image. Sometimes, the filter does not fit well with the image, that means the filter would be missing some parts of the image. Thus, doing padding into the image makes sure the filter captures all the information from the image. Stride is the number of pixels indicating how far the filter should move from current position on the image. The output of the filter bank layer is the input of the non-linearity layer which consists of an activation function. Most researchers use the Rectified Non-Linear Unit (ReLU) [43] which returns the direct input if it is not negative, otherwise returns zero. The purpose of non-linear activation function is to prevent the model from becoming a linear model and help the model to deal with the complex task. The last layer of CNNs block is the pooling layer which also uses the approach of scanning images. It applies the specific function with the filter size (width,height) as the parameter such as max pooling, average pooling or sum pooling onto every region of feature maps. For example, applying max pooling

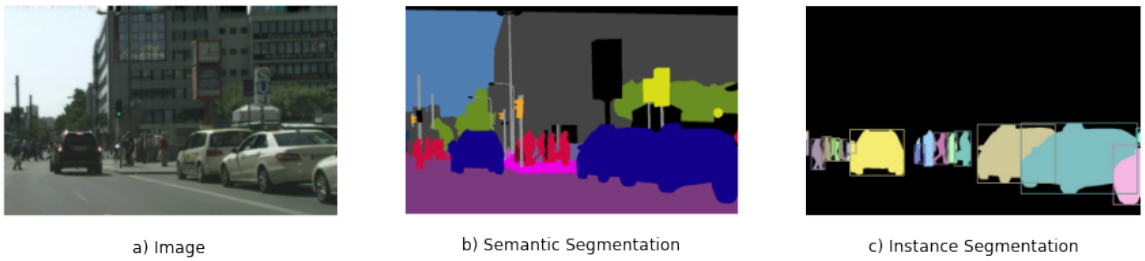


Figure 2.4: An example of semantic and instance segmentation [44].

with the filter size 2x2, the function will output the highest value of a specific region. That means, pooling layer can reduce the dimensionality of feature maps and also keep the most important information out of the region. In the higher level view, the two key benefits of CNNs are parameter sharing and sparsity of connections. With parameter sharing, a filter bank is useful in one part of the image and is probably useful in another part, thus it can reduce the parameter needed. With sparsity of connections, in each layer, each output value depends only on a small number of inputs.

Lastly, all feature maps captured from a series of blocks are flatten out into input units and fed into the classification module, called fully connected layer. This layer is like a normal neural network which outputs the score for each class. CNNs are normally back-propagated as neural networks and noticeably the weight of filters also be trained.

2.4 Deep learning for semantic segmentation

Image segmentation can be divided into two main tasks of semantic segmentation and instance segmentation. Semantic segmentation techniques label each pixel of an image with the class it belongs to. For example, label a pixel as belonging to a tumor or the background. Instance segmentation extends semantic segmentation by further separating pixels based on if they belong to different instances of the same class. Figure 2.4 shows an example of 2 types of segmentation.

From the earliest days, image segmentation techniques have centralized on traditional algorithms such as thresholding, region-growing, k-means clustering, etc. However, due to the success of deep learning models (especially CNNs), image segmentation methods have significantly improved in accuracy via CNN based methods such as DeepLab [45], U-Net [46]. To confine the scope of this review, we will focus our attention on semantic segmentation methods that use deep learning models.

As discussed in Section 2.3, CNNs are a dominant approach in computer vision, especially in the image classification task with the ability to automatically learn features and output the class prediction for the entire image (global prediction). Therefore, researchers have adapted the deep CNN approach to solve the semantic segmentation

task by making it output the class prediction for each pixel of the image (local prediction), known as the prediction mask. Early methods used CNNs to classify the center pixel for a given patch of input [47]. The image patch is slid across the image in order to output a prediction for each pixel of the image thus achieving the required output for semantic segmentation. However this approach is not efficient since the same CNN kernel is applied to the overlapping regions of nearby patches multiple times.

In order to improve the efficiency of applying CNNs to semantic segmentation, an intuitive idea [48] is to use fully convolutional neural networks to process the entire image in a single pass through the CNN. To do so, the CNN architecture is modified to keep the output size the same as the input image by using downsampling and upsampling layers within a fully convolutional neural network. This idea not only brings computational efficiency by downsampling layers but also produces the required output size using the upsampling layers. Moreover, the approach can obtain the global spatial information of the image. Due to their effectiveness and efficiency, fully convolutional neural networks are used in all current state of the art semantic segmentation methods.

2.4.1 Popular existing models

Fully Convolutional Networks

Long et al. [48] proposed a fully convolutional model by modifying existing state of the art CNN classification models such as VCG16, GoogleNet. The author replaced fully connected layers with fully convolutional layers including up-sampling layers. It allows the model to process an arbitrary input image size and produce a prediction mask of the same size as the input. Figure 2.5 illustrates the model. Notably, up-sampling layers include learnable deconvolution filters that up-sampling is performed using bilinear interpolation. Additionally, to refine semantics and spatial accuracy of the output, the author used skip of connections which combines the coarse, high level information of feature maps in middle layers with the lower level information from the earlier layers, Figure 2.6. The methods achieved 67.2% mean IoU on PASCAL VOC 2012 [49] dataset, establishing a new state of the art result at the time.

Deconvolution Network (DeConvNet)

The Deconvolution Network (DeConvNet) is another well-known CNN based model for pixel-wise segmentation which is inspired by encoder-decoder networks. Noh et al. [50] have proposed a novel segmentation network consisting of the convolution network (acts like an encoder network that encodes an input image into a vector of

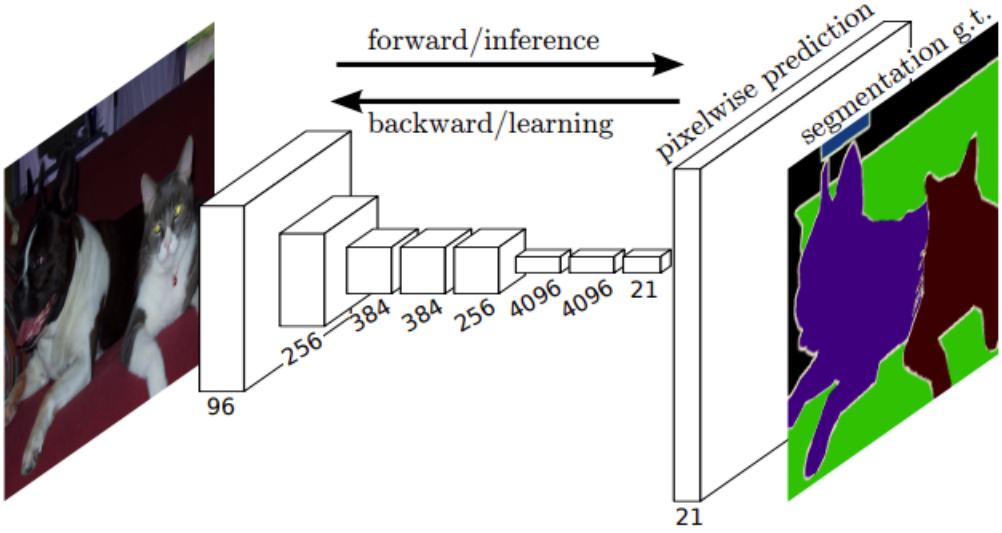


Figure 2.5: Fully convolutional networks for semantic segmentation. [48]

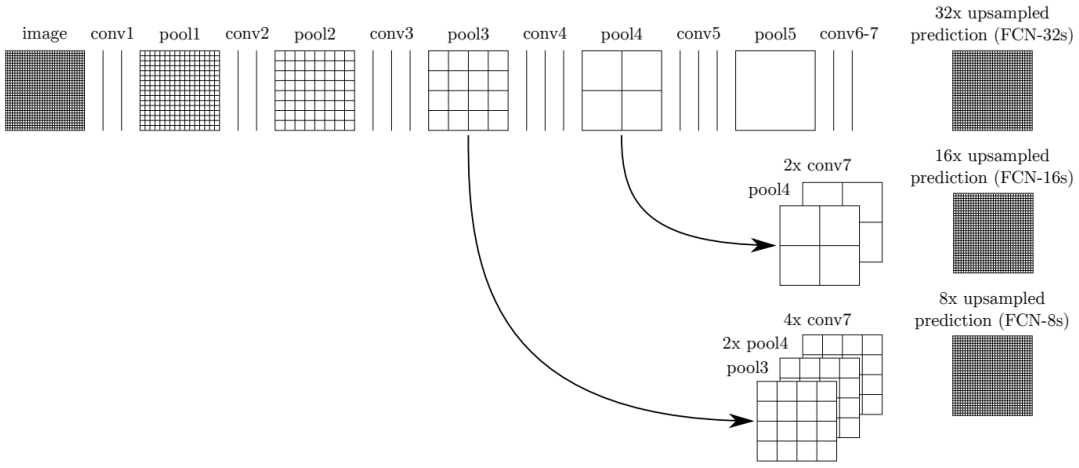


Figure 2.6: Skip connections: allows the model to use high level information from the middle layers to inform the final fine grained predictions. [48]

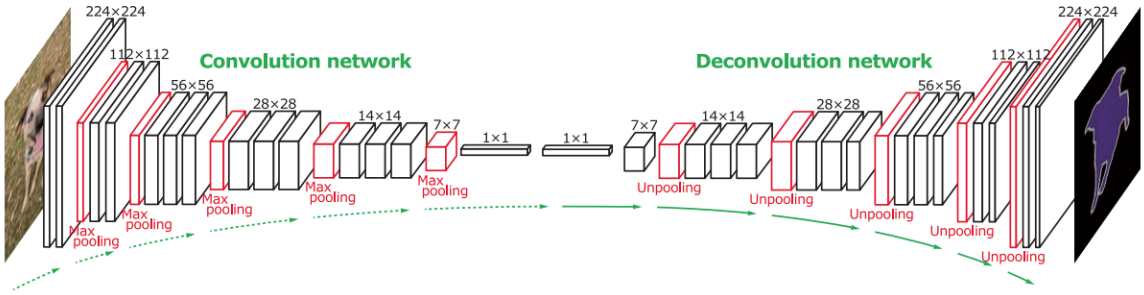


Figure 2.7: Overall architecture of the deconvolution network, based on the VGG 16-layer CNN model [50]

features) and the deconvolution network (acts like a decoder network) that transforms the encoding vector of features to a dense pixel-wise class prediction map of the same size as the original image. The model architecture for DeConvNet is shown in Figure 2.7. Notably, the convolution network (encoder) is not fully convolutional because it includes 2 fully connected layers which are augmented at the end to impose class-specific projection. The highlight part of this study is the deconvolution network which contains a sequence of trainable deconvolution, unpooling and rectified linear units (RELU) layers. The motivation for proposing a complete deconvolution network is the absence of real deconvotion in previous models (FCNs [48]) which is one of the main limitations. The network achieved 72.5% on PASCAL VOC 2012 dataset and outperformed all rival networks at the time.

SegNet

Badrinarayanan et al. [51] proposed an interesting fully deep convolutional Encoder-Decoder network for image segmentation, named SegNet. The network architecture is quite similar to DeconvNet which consist of the convolution network (encoder) and deconvolution network (decoder), however the author removes 2 fully connected layers of the VGG16 encoder network to make SegNet become fully convolutional. The novelty of SegNet is how the decoder network upsamples the feature maps from the encoder network. Specifically, the decoder network uses pooling indices obtained in the max-pooling step of the corresponding encoder to perform unpooling like DeconvNet, but without learnable deconvolution filters. Therefore, the upsampling stage requires no learning. After each upsampling, the upsampled maps which are sparse are convolved with trainable filters to produce dense feature maps. The network is illustrated in Figure 2.8.

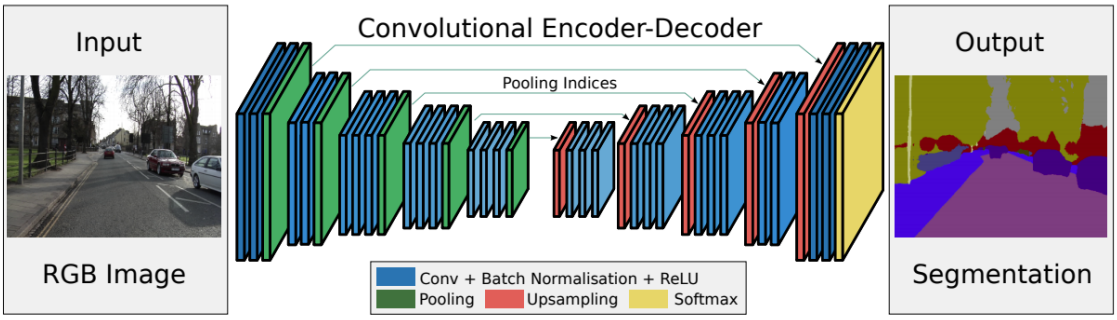


Figure 2.8: The SegNet network architecture. [51]

UNet

So far we have discussed popular existing models for general image segmentation. In contrast the UNet [46] was initially proposed to solve medical image segmentation and then be used in general tasks. Ronneberger et al. [46] built a U-shaped fully convolutional network for semantic segmentation comprising a contracting path (down-sampling) to capture the context and a symmetric expanding path (up-sampling) to localize objects. The contracting path architecture is similar to a typical convolution network which obtains feature maps and reduces its resolution. While the expanding path is deconvolution network-like, that upsamples feature maps. At every upsampling layer, by using the skip connections, upsampled feature maps and the correspondingly cropped feature maps from the contracting path are concatenated. Then these concatenated feature maps are convolved by two consecutive 3 x 3 convolutional filters to halve the number of channels, hence yields U-shape network, illustrated in Figure 2.9.

DeepLab

Chen et al. proposed a series of DeepLab models including DeepLabv1 [45], DeepLabv2 [52], DeepLabv3 [53] and DeepLabv3+ [54], which have become one of the most popular models for semantic segmentation.

DeepLabv1 [45] uses a fully convolutional neural network architecture (with VGG 16 as backbone) for semantic segmentation. It has the two important features which are dilated convolutions in the convolutional layers and fully connected Conditional Random Field (CRF) (probabilistic graphical model) at the output stage. With no extra computational cost, dilated convolutions enlarge the receptive field of the kernel by using gaps when applying the kernel to the input. By applying dilated convolutions, the author claims it can reduce the degree of signal downsampling, as well as condition on larger context. Applying the fully connected CRF on the segmentation score map refines the segmentation result and achieves better object localization. As illustrated in Figure 2.10, the downsampled feature map from the fully convolutional network

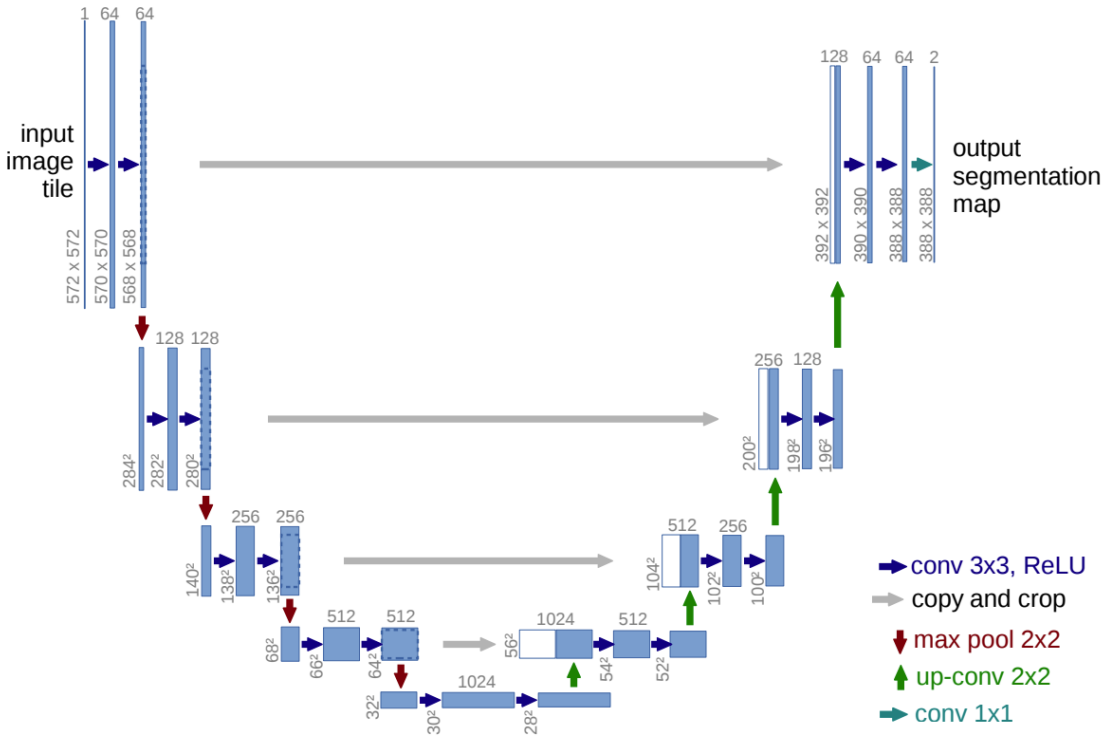


Figure 2.9: UNet network architecture. The left is the contracting path and the right is the expanding path. [46]

is first upsampled to the original image resolution using bilinear interpolation and then refined by the fully connected CRF to produce the final prediction mask. At publication time DeepLabv1 achieved the state of the art result of 71.6% IoU at the PASCAL VOC-2012 semantic image segmentation task.

One year later, Chen et al. [52] proposed DeepLabv2 by updating DeepLabv1 with 2 major improvements. First, DeepLabv2's fully convolutional layers used a ResNet CNN as backbone. Second, the author also proposed atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales, illustrated in Figure 2.11. Specifically, the incoming convolutional feature layers are filtered using multiple parallel dilated convolutional layers with different dilated rates, hence segmenting objects as well as image context at multiple scales. This work established the new state of the art result on PASCAL VOC2012, achieving 79.7% IoU.

Chen et al. again updated DeepLabv2 to propose a better variant of DeepLab which is DeepLabv3 [53] by revisiting the use of dilated convolution and removing the use of fully connected CRF. The author experimented with DeepLabv3 with 2 main ideas: **Going Deeper with Atrous Convolution**, illustrated in Figure 2.12 the design is to have more blocks with atrous convolution in cascade. The dilation introduced in the convolutional kernels makes it easier to capture long range information in the deeper blocks. **The modified Atrous Spatial Pyramid Pooling (ASPP)**, proposed in DeepLabv2 is illustrated in Figure 2.13. The new ASPP is added with learnable batch normalization, image-level features and 1x1 convolutions.

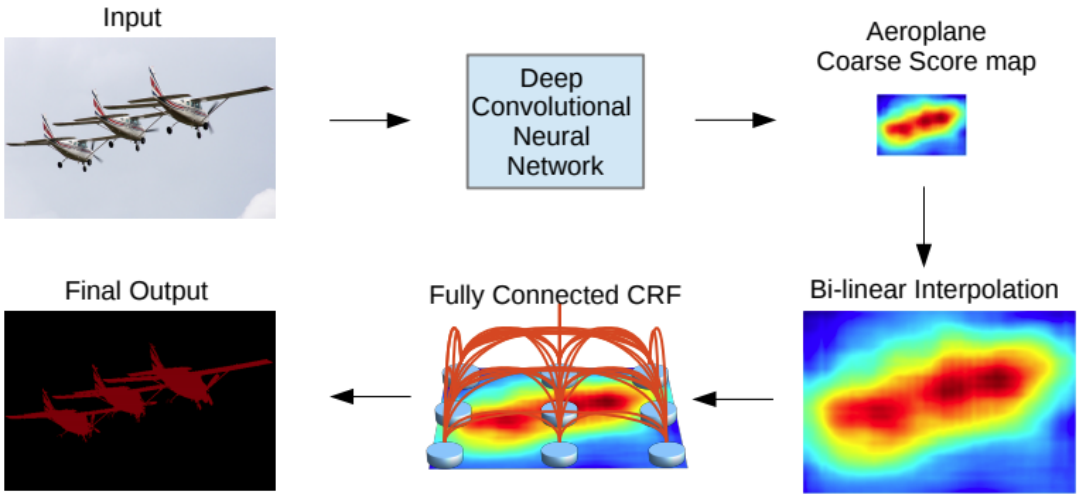


Figure 2.10: The overall model architecture of DeepLabv1. [45]

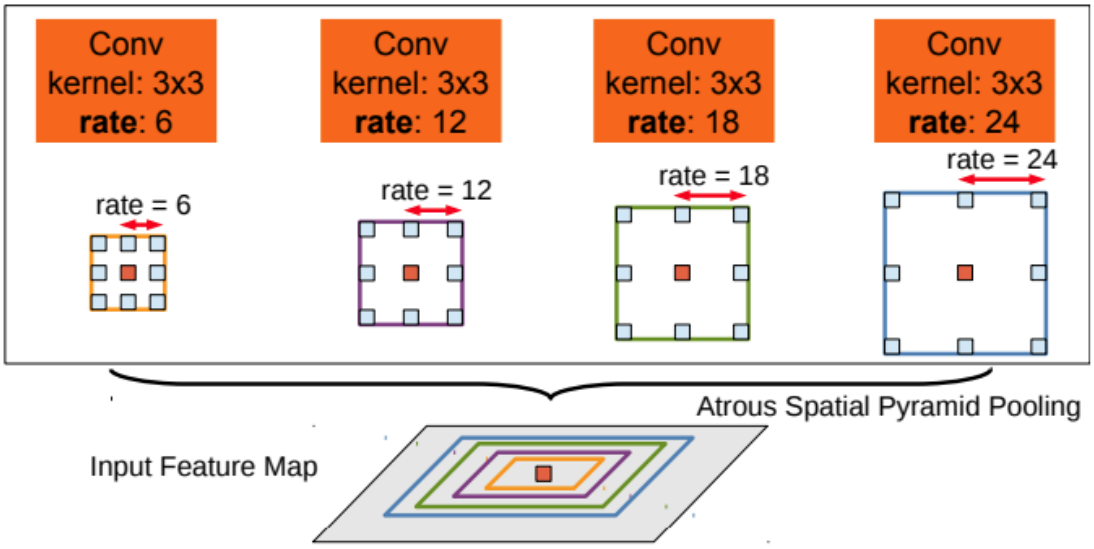
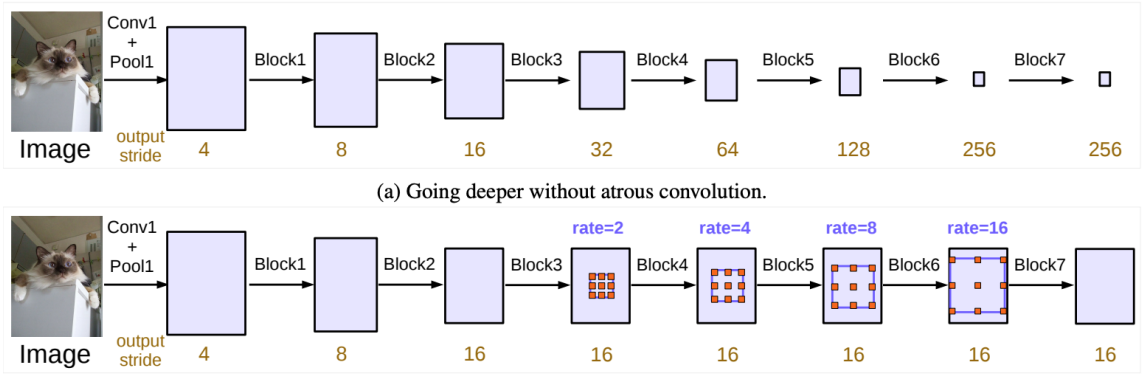


Figure 2.11: The illustration of ASPP [52]



(a) Going deeper without atrous convolution.

(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Figure 2.12: The overview of atrous convolution designed in cascade in DeepLabv3. [53]

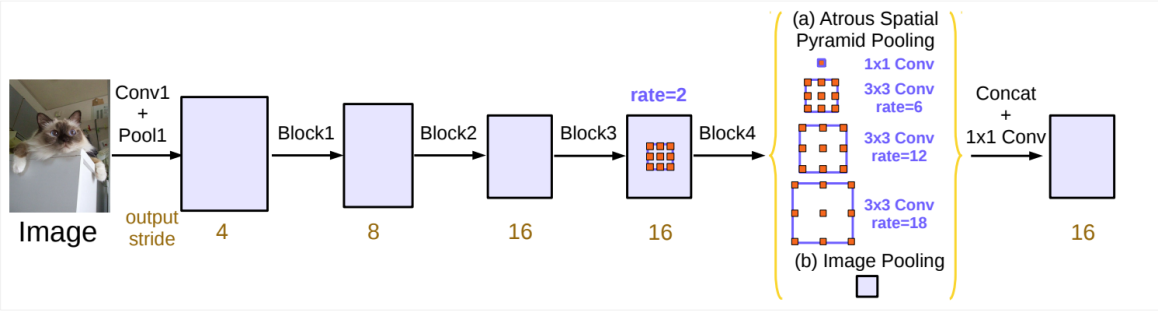


Figure 2.13: The overview of new ASPP in DeepLabv3. [53]

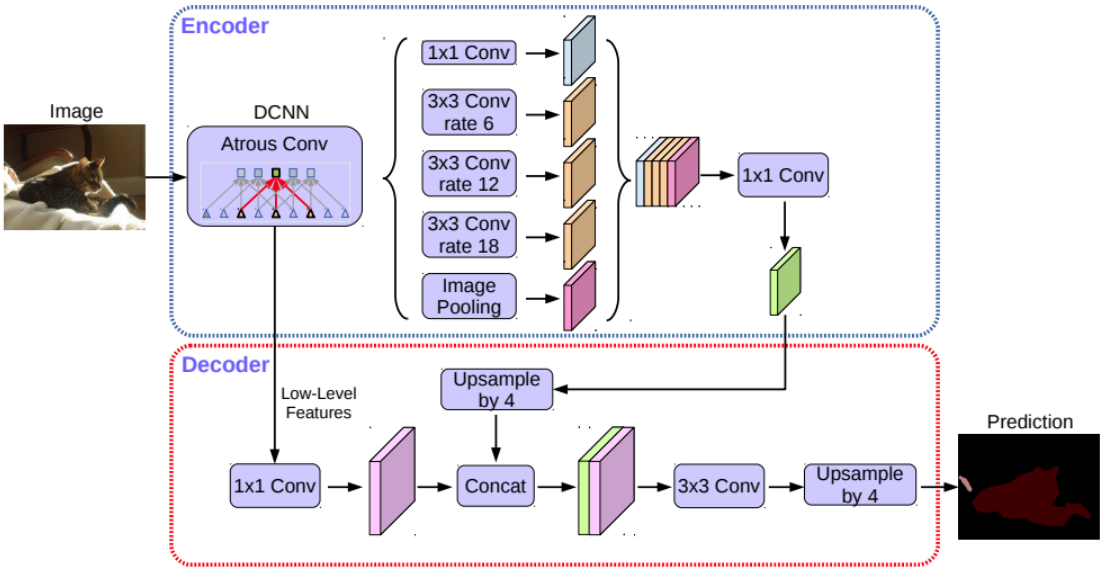


Figure 2.14: The overall architecture of encoder-decoder-like semantic segmentation network of DeepLabv3+ [54]

All feature maps produced from multi-parallel dilated convolution with different rates and image-level features are concatenated and processed by the 1x1 convolution, then passed through the final 1x1 convolution to produce final logits. DeepLabv3 achieved a significant improvement on the PASCAL VOC 2012 dataset with 85.7% IoU.

Chen et al. extended DeepLabv3 to release DeepLabv3+ [54] by adapting the system into encoder-decoder semantic segmentation architecture and using atrous convolution layers separable. Figure 2.14 shows the architecture of DeepLabv3+. The DeepLabv3 network is used as an encoder network with the Xception network as backbone to the fully convolutional layers and still employs the ASPP. Moreover, a new effective decoder network is added to refine the segmentation results especially for large objects. The encoding features are first upsampled using bilinear interpolation and then concatenated with the corresponding and same resolution low-level features from Xception FCNs backbone. Then these concatenated features are convolved and enlarged again to produce the final prediction map. DeepLabv3+ achieved a 3.3% improvement on the results of DeepLabv3, reaching 89% IoU on PASCAL VOC 2012 dataset.

2.5 Data Augmentation for image classification

Recently, data augmentation for images have played a very significant role in the successful training of deep CNN models which typically require a large amount of data. This is a technique of creating more images from information of existing images without changing the meaning of images, hence making the data set more diverse. Applying data augmentation helps the model frequently see more unseen data, consequently tackle the overfitting issue and make the model generalise better. Data augmentation for images is mainly divided into three groups: normal augmentation, neural network based method and augmentation search.

2.5.1 Normal Augmentation

In normal augmentation, the additional images are generated by using simple mathematical functions that transform various aspects of images such as texture, shape and color. Below we describe some of the most popular data augmentation methods:

Rotation: The image is rotated clockwise or counter-clockwise by a random angle between 0 to 360 degree. This means the object in the image will be rotated but its shape will not change. Figure 2.15

Flipping: We can flip the direction of the image either horizontally or vertically. That means the transformed image is horizontally or vertically symmetrical with respect to the original image. Figure 2.15

Shifting: Shifting augmentation is also known as translation augmentation that moves all pixel values of the image in the left,right,top and bottom direction. In the case that the object in the image moves too far, the image may lose too much information, thus making it impossible to make a correct prediction on its label. Figure 2.15

Crop: A region of the image is randomly cropped with a determined size. This technique generates new images of smaller size. Figure 2.15

Cutout [55]: Inspired by the dropout regularization technique [56], cutout is the technique that randomly removes a fixed sized region from images and replaces it with random pixel values. It helps the computer vision machine model to see more simulated situations where the object in the image is partially occluded. Moreover, this technique is also relatively similar to Random Erasing [57]. Figure 2.15

Colour Transformation [58]: For grayscale images, each pixel contains one intensity value in the range of 0-255 inclusive. For color images, each pixel contains 3 intensity values of color channels represented in a particular color space system such as RGB (Red-Green-Blue), CMY (Cyan-Magenta-Yellow) and HSI (Hue-Saturation-

Intensity). To do colour augmentation, four aspects (hue, saturation, brightness and contrast) are manipulated. The most popular methods of colour transformation are histogram equalisation, and adjusting the following: brightness, contrast, white balancing, sharpening and blurring [59]. Figure 2.16

Noise Injection [58]: The image is injected with an amount of noisy pixel value, usually from Gaussian noise in order to help the model to learn more robust features. Figure 2.15

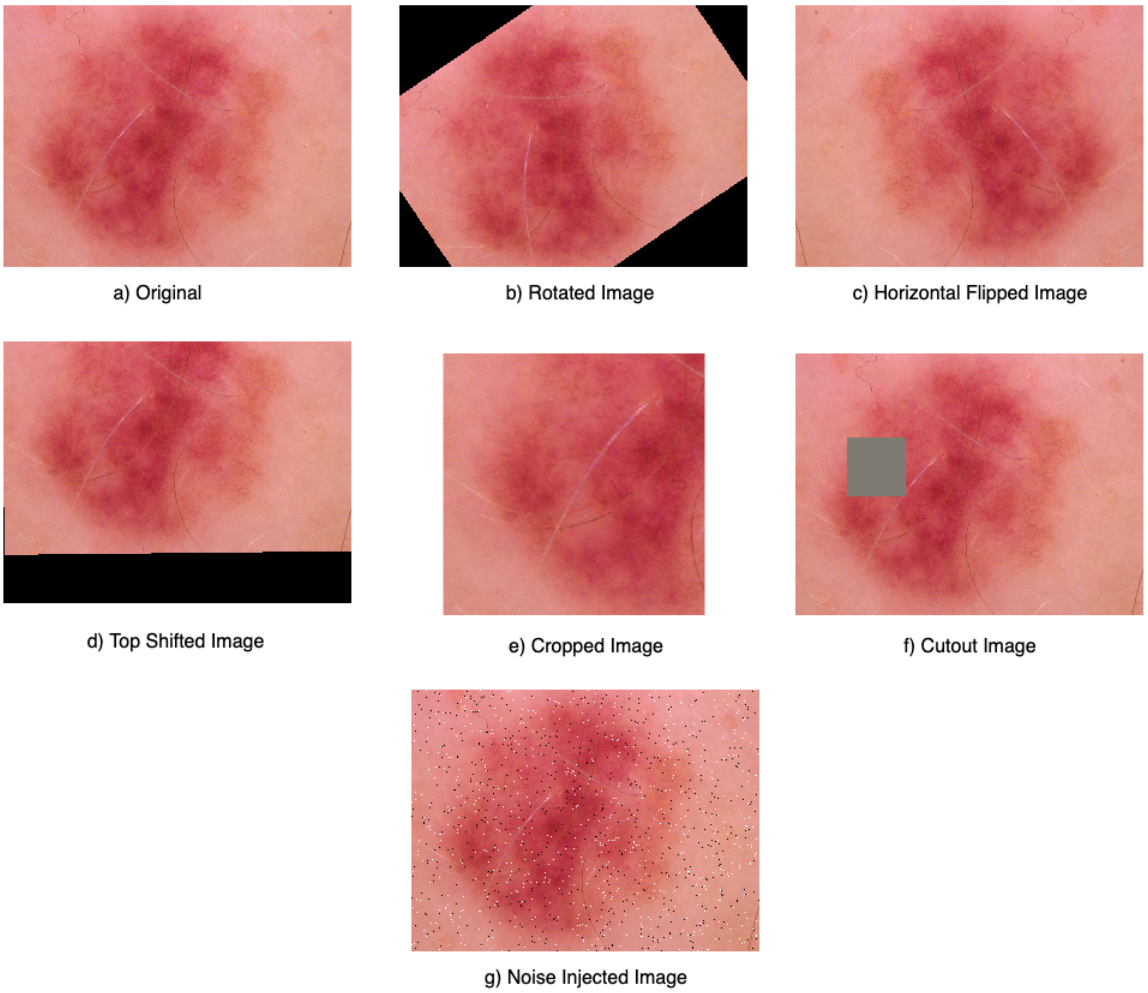


Figure 2.15: Examples of normal augmentation method on dermatoscopic skin lesion image.

2.5.2 Neural Network Based method

In this method, the new image is generated from a trained neural network and it will preserve the semantic of its label.

Generative Adversarial Networks(GANs) [60]: The idea of GANs is that having the generative model and the discriminative model, the discriminative model is trained with real data to be able to detect the sample data whether is real or fake. Then, the generative model is trained to pit against the trained discriminative model, that means the generative model releases the sample that is detected as real data by

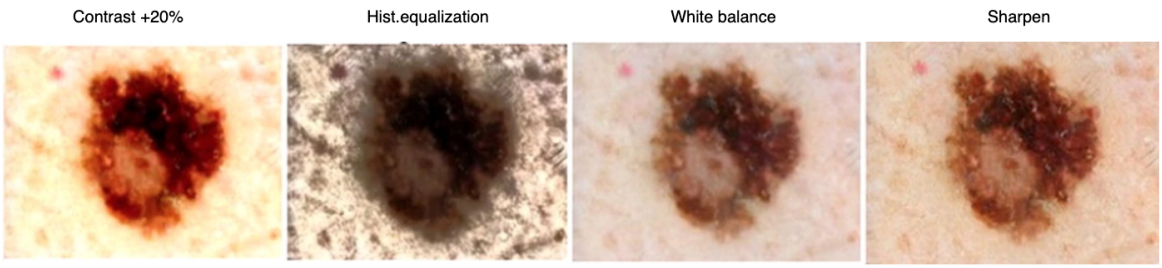


Figure 2.16: Examples of colour augmentation [58].

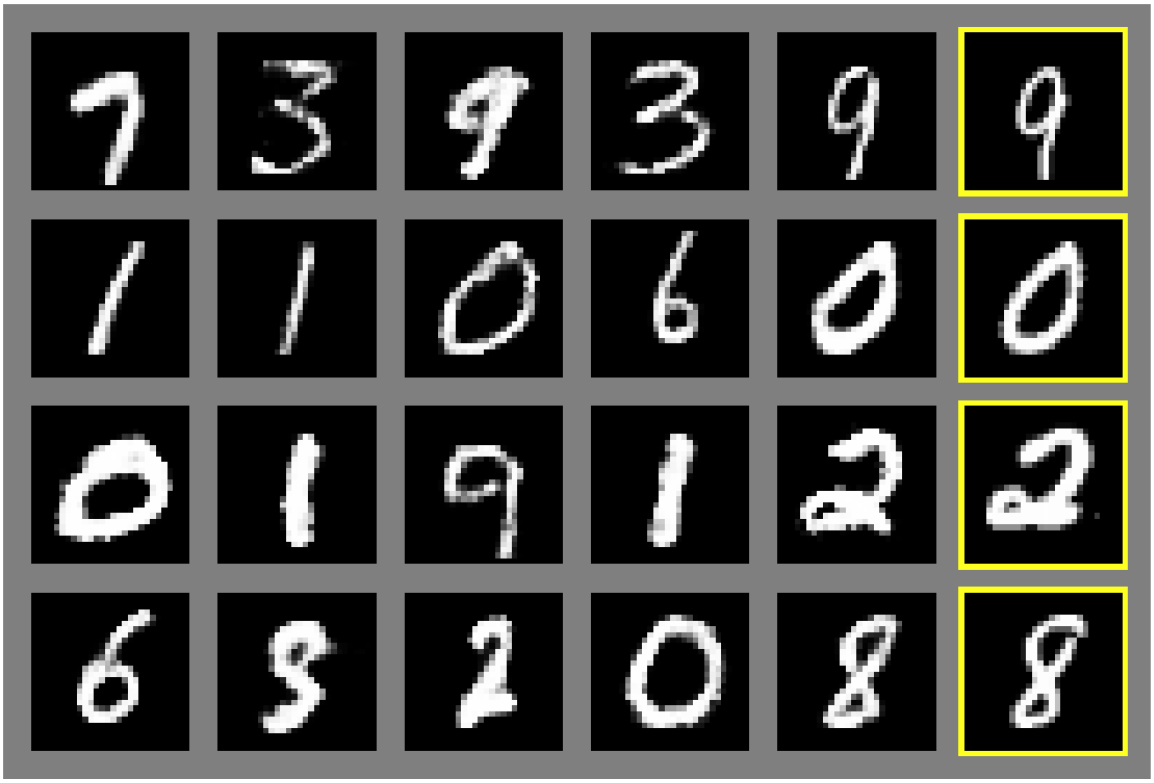


Figure 2.17: Samples are drawn from a generator. The yellow box is the latest version of each sample [60].

the discriminative model. Figure 2.17

Neural Style Transfer [61]: This technique is not only known as the creation of artistic images, but is also a very useful technique for image augmentation such that augmented images have the same content as original images but different styles. The idea is to feed two images into two CNNs, one of those contains the content for the new image and the other contains the style. The CNNs will separate and recombine the content and style in order to generate new images. Figure 2.18

2.5.3 Augmentation Search

Manually applying many different random normal augmentation techniques to the training process at the same time makes it hard to figure out which one improves the model's performance. Thus, augmentation search algorithms automatically find an

Figure 2.18: The artistic image B is generated by CNNs. The image A is the content image and the small image at bottom-left is style image [61].

optimal data augmentation policy which determines for a given data set the best set of normal augmentation methods to use and their corresponding best settings.

AutoAugmentation [62]: This method consists of two components: a search algorithm and a search space. The search algorithm uses a reinforcement learning algorithm which finds the best data augmentation policy from the search space. In the search space, a policy consists of many sub-policies. Moreover, each sub-policy has two normal image transformations corresponding with the probability of applying transformations and the magnitude of transformations.

RandAugmentation [63]: AutoAugmentation is computationally expensive due to its focus on finding augmentation policies using a separate search phase performed on proxy tasks which include searching different models and dataset sizes. In contrast, RandAugmentation directly evaluates the impact of the data augmentation on the target model and dataset. This results in a much smaller search space involving the set of available operations, magnitude of all operations and just two hyperparameters (N and M) to tune.

2.6 Imbalanced Class

2.6.1 Image classification

For supervised learning

Training data often do not have balanced distribution. For example fraud classification from transactional data, fraud transactions are much rarer than non fraud

transactions. Another example task in healthcare is cancer detection where there are usually many more healthy samples than cancer samples. Therefore, in the testing phase, the machine learning model mostly predicts the majority class in the training set rather than the minority class, leading to much lower accuracy for predicting the minority class. Many methods have been proposed to address class imbalance, which can be grouped into three categories: data-level, algorithm-level and hybrid approaches [64].

Data-level Approach: Before feeding the data into the model during training, the dataset is resampled to create a balanced distribution across classes. This randomly reduces the number of majority class samples and randomly increases the sampling of the minority class. However there are some downsides of these techniques, according to [64] under-sampling the majority class might substantially reduce the amount of information from the eliminated data. Over-sampling will increase training time and may cause the model to over-fit. To overcome these downsides, several advanced sampling methods have been proposed in the literature. For under-sampling, in [65, 66, 67], intelligent methods are used to efficiently select which majority class samples are eliminated. For oversampling in [68, 69, 70], intelligent methods are used to create new artificial minority class samples from existing minority class samples.

Algorithm-level Approach: Instead of resampling the data distribution, this method modifies the model's algorithm to give more emphasis on the minority class and can be divided into 3 approaches [64]: new loss function, cost-sensitive learning and threshold moving. Mean False Error [71], Focal Loss [72], Weighted Cross Entrpy Loss [73] are new loss functions that reduce the effect of majority class on the loss. In cost-sensitive learning [74, 75, 76] the model learns from the classification cost matrix, that means each class is assigned different costs. Lastly, in the threshold moving technique [77, 78, 79], the idea is to adjust the decision threshold to reduce bias on easy samples, in other words, make more correct predictions on hard samples.

Hybrid-level Approach: There are several hybrid methods [80, 81] that combine both of the above approaches that firstly resamples data distribution and then applies algorithmic approaches such as cost-sensitive learning.

For semi-supervised learning

All of the above methods which alleviate skewed data distribution only works for supervised learning where we know the labelled data distribution. It is really challenging to use a semi-supervised learning approach where the distribution of the unlabelled data is unknown.

Suppressed Consistency Loss: For SSL perturbation based methods, Minsung [82] has proposed an algorithm level method which is Suppressed Consistency Loss (SLC). In the imbalanced class learning environment, the decision boundary is more

likely to move into the low-density areas of the minor class with consistent regularization that causes the model to misclassify the minor class. Therefore, SLC will suppress the consistency loss of minor classes which in turn tends to push the decision boundary against the low-density areas.

2.6.2 Semantic segmentation

While classification suffers from class imbalance at the whole image grain, semantic segmentation suffers from the same problem at the pixel grain. Especially in binary medical image segmentation, the number of background class pixels is often much larger foreground class pixels. The methods to deal with class imbalance in semantic segmentation are also divided into 2 main groups: data level and algorithmic level approaches.

Data level: It is difficult to use oversampling or undersampling methods developed for image classification for semantic segmentation. According to [83], it would be meaningless to undersample an image by removing some major class pixels/voxels because it would remove a lot of information from the image. For patch-based segmentation approaches, there are several existing works [47, 84] that use the patch selection algorithm to balance the number of patches from major and minor classes.

Algorithmic level: Algorithmic approaches are more commonly used to address class imbalance for segmentation tasks. These methods include Weighted Cross Entropy Loss [73], Focal Loss [72], Dice Loss [85] which is built from Dice Coefficient metric and aims to maximize the similarity of two images, Tversky Loss [86] which modifies Dice Coefficient metric in a way that weigh False Negative more than False Positive and pre-compute weight map for the ground truth in UNet [46].

To our knowledge there are no studies that mainly focus on dealing with class imbalance in the context of semi-supervised semantic segmentation. Therefore, we address this gap in the literature.

Chapter 3

Related Works

3.1 Semi-supervised learning

Our focus is on intrinsically semi-supervised methods, a subclass of inductive semi-supervised learning. In order to make use of unlabelled data, these methods often follow smoothness and low-density assumptions. The low-density assumption states that the decision boundary should pass through low-density areas in the data space. The smoothness assumption states that if two data points are close, it should have the same predicted label [31]. Based on the assumption, the perturbation based method increases the robustness of the model by utilizing noise on unlabelled data. That means the original input and the input perturbed with noise should both produce the same predicted label. There are several techniques based on this approach which recently achieved state of the art results, mostly on various classification tasks, such as Temporal Ensembling [27], Mean Teacher [26] and Virtual Adversarial Training [28].

$\prod$-MODEL [27]: Inspired by ensemble learning, the idea is to encourage the consistency of a model to two of its outputs z and $\tilde{z}$, which is produced from the same input by applying different dropout and data augmentation conditions, over each batch training. Figure 3.1 shows the learning framework. The model is learned from two losses, the supervised loss on z prediction of only labelled inputs by using the cross entropy loss function and mean squared loss between the z and $\tilde{z}$. On CIFAR-10 with only 4000 labels, the $\prod$-MODEL achieved an error rate of 12.36%.

Temporal Ensembling [27]: This technique is built on top of the $\prod$-MODEL with the difference being the model only produces one output z and the prediction $\tilde{z}$ is based on the model's output in the previous epoch. The learning framework has shown in the same figure of $\prod$-MODEL, Figure 3.1. After every training epoch, the model's output z is accumulated into an ensemble of outputs Z by updating $Z \leftarrow \alpha Z + (1- \alpha)z$ and calculating $\tilde{z}$ by dividing Z by factor $(1-\alpha^t)$. By only producing one output at the time, the Temporal Ensembling learning method is faster than $\prod$-

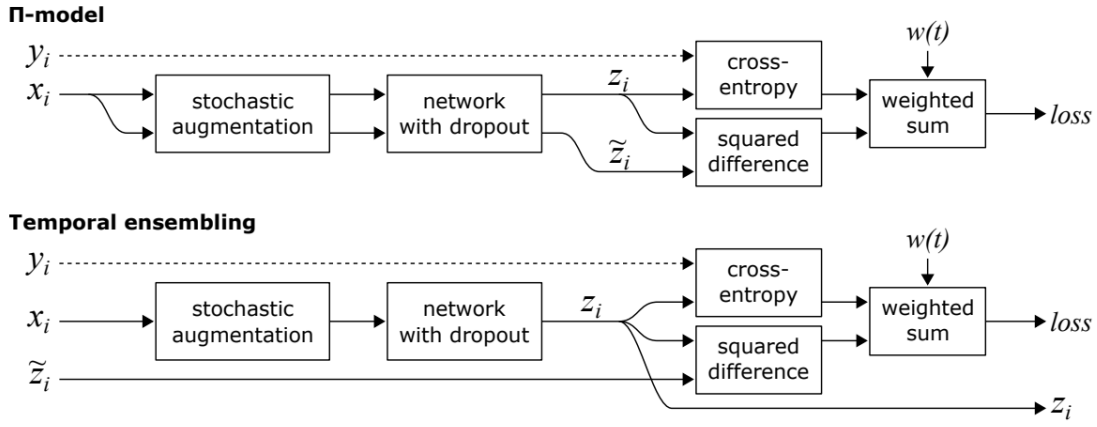


Figure 3.1: [27] Learning framework of $\prod$-MODEL and Temporal Ensembling method.

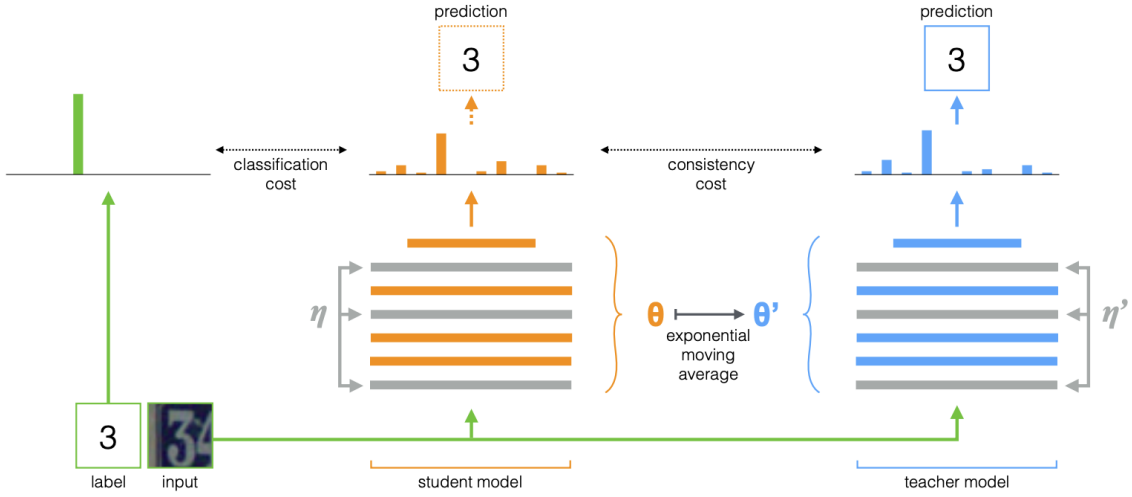


Figure 3.2: [26] Learning framework of Mean Teacher method.

MODEL. Moreover, the output $\tilde{z}$ calculated in the previous epoch can be expected to have less noise than the one in $\prod$-MODEL. The slightly better result is reported on 4000 labels of CIFAR-10 with 12.16% error rate.

Mean Teacher [26]: This technique consists of the dual student and teacher models which produce z and $\tilde{z}$ prediction from all inputs X by applying two types of stochastic noise. Figure 3.2 illustrates the learning framework. The weights θ of the student model is updated by the supervised loss on only labelled data of student's output z and the consistency loss which is the distance between student's output z and teacher output $\tilde{z}$ from the entire input. Whereas, the teacher's weight θ' is updated by the exponential moving average of the student's weight over each batch training, which computed as follows: $\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$. On 4000 labels of CIFAR-10 and trained on ResNet, the Mean Teacher achieved the state of the art result with 6.28% error rate.

Virtual Adversarial Training [28]: Most perturbation based methods apply stochastic augmentation to the input. However, in this approach, the authors directly applied small directional noise to the corresponding input which makes the model's prediction change the most, also known as adversarial example [87]. Basically, the method is inspired by Adversarial Training [87] which uses the following loss function:

$$L_{adv}(x_l, \theta) := D[q(y|x_l), p(y|x_l + r_{adv}, \theta)] \tag{3.1}$$

$$\text{where } r_{adv} := \underset{r;||r||\leq\epsilon}{\arg\max}\, D[q(y|x_l), p(y|x_l + r, \theta)] \tag{3.2}$$

The goal of the loss function is to minimize the divergence between the true distribution of output label $q(y|x_l)$ and the model's output distribution $p(y|x_l + r_{adv}, \theta)$ by applying adversarial noise r_{adv} to inputs x_l. However, the adversarial training only works well on supervised tasks. To transfer to the semi-supervised task, the loss function is reformulated as:

$$L(x_*, \theta) := D[p(y|x_*, \hat{\theta}), p(y|x_* + r_{adv}, \theta)] \tag{3.3}$$

$$\text{where } r_{adv} := \underset{r;||r||_2\leq\epsilon}{\arg\max}\, D[p(y|x_*, \hat{\theta}), p(y|x_* + r)] \tag{3.4}$$

Where x_* is either unlabelled x_u or labelled x_l. Indeed, we can not obtain $q(y|x_u)$, therefore the authors proposed the term called virtual labels which replaces $q(y|x_*)$ with its current output $p(y|x_*, \hat{\theta})$. This is the reason for the use of the terms "virtual" on all of the techniques reported by the paper including virtual adversarial training, virtual adversarial noise. On 4000 labels of CIFAR-10, the Virtual Adversarial Training achieved the state of the art result with 10.55% error rate.

Unsupervised Data Augmentation [1]: the authors found applying advanced data augmentation on unlabelled samples could effectively make the model more invariant to noisy input in the semi-supervised learning approach. Instead of applying only some normal data augmentations such as flip, rotation in Temporal Ensembling, Mean Teacher or virtual adversarial noise in Virtual Adversarial Training, they used an auto augmentation method called **RandAugmentation** [63] to find policies and magnitudes of chosen transformations which yield the best performance on labelled samples and then apply these to unlabelled samples. As a result, on CIFAR 10 with 4000 labels, this approach achieved a 5.29% error rate which is near the state of the art result for supervised learning approach.

MixMatch [29]: This approach does not fully belong to perturbation based method because it just utilizes the idea of perturbation based semi-supervised method as a part of the algorithm. MixMatch unifies current dominant approaches such as perturbation based semi-supervised learning methods discussed above, entropy min-

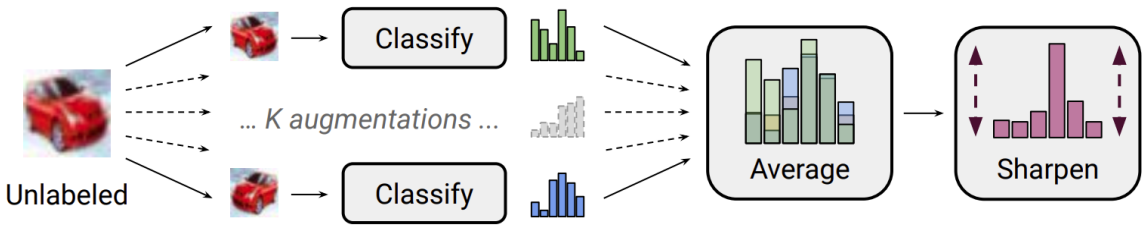


Figure 3.3: [29] Diagram of the label guessing process in MixMatch.

imization which encourages the model to make low entropy prediction on unlabelled samples and **MixUp** [88], a technique that combines samples for training. Moreover, these approaches are combined to generate a new set of labelled data X' and unlabelled data U' with predicted labels. In more detail, firstly, apply data augmentation on a batch of labelled samples X and unlabelled samples U, to create augmented batches $\hat{X}$ and $\hat{U}$. Then, construct the predicted labels for augmented $\hat{U}$ by applying a Sharpening function to reduce the entropy of the average model's predictions on augmented $\hat{U}$ Figure 3.3. Next, to prepare for the MixUp step, shuffle $\hat{X}$ and $\hat{U}$ with predicted labels to create set W. After that, apply the MixUp function on $\hat{U}$ and a part of W samples, $\hat{X}$ and the rest of W samples to produce a new set of labelled data X' and unlabelled data U'. Finally, the authors just used the standard semi-supervised learning loss framework which uses the cross entropy loss function for MixUp X' model's prediction of labelled data, L2 loss for MixUp U' model's predicted outputs on unlabelled data. With 4000 labels on CIFAR 10, this technique achieved the state of the art result of 4.95% error rate.

Another subcategory of intrinsically semi-supervised learning is generative models. The primary characteristic of this method is to generate new unlabelled data for classification.

Categorical Generative Adversarial Networks [89]: The idea is to build on top of **GANs** [60] which consists of the discriminator and generative functions and only work on unsupervised tasks. To deal with semi-supervised tasks, the authors made an extension to the discriminator so that classifies K class for all samples from the labelled dataset and discriminates real versus fake samples from the generator. Intuitively, the job of the generator is changed from "generate samples that belong to the real dataset" to "generate samples that belong to the K class dataset". Additionally, cross entropy loss term is also added to penalize the error of discriminator's prediction on real data and corresponding labels. As a result, on standard 4000 labels CIFAR, the network achieved a 19.58% error rate.

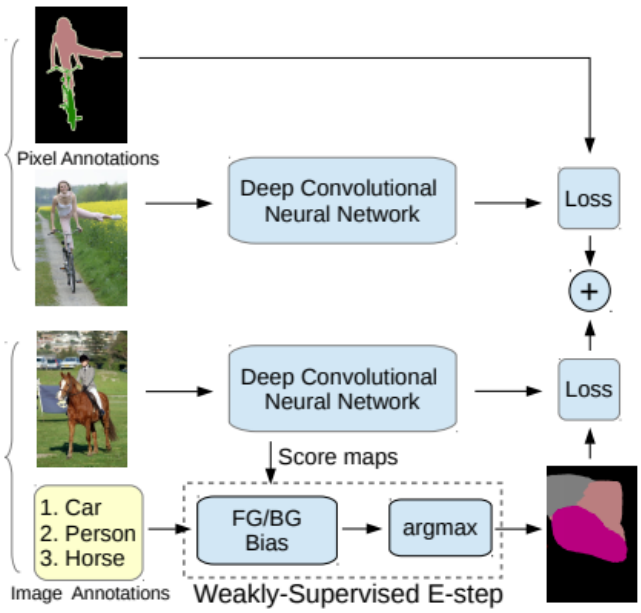


Figure 3.4: DeepLab model training methodology using a combination of pixel-level (strong labels) and image-level (weak labels) annotations. [90]

3.2 Semi-supervised learning in semantic segmentation

In this section, we review several popular semi-supervised semantic segmentation frameworks which combine small amounts of pixel-level labelled data with large amounts of data that are either weakly labelled or not labelled. Weak labels include bounding boxes or image-level labels.

3.2.1 Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation (2015)

Papandreou et al. [90] propose a semi-supervised framework for semantic segmentation by using DeepLab as a segmentation network and proposing Expectation-Maximization (EM) algorithm to estimate the segmentation mask from weakly labelled data. The total loss has two components, illustrated in Figure 3.4: one on strongly labelled data is from the prediction map produced by DeepLab network with its pixel-level annotations; another loss on weakly labelled data is from the prediction map produced by DeepLab network with its estimated pixel-level map which is generated from the EM algorithm. They conducted their experiments using the PASCAL VOC 2012 dataset, with 1400 strong labels and 9000 weak labels (image-level labels) which is generated by summarizing the pixel-level annotations. The proposed method achieved 66% IoU.

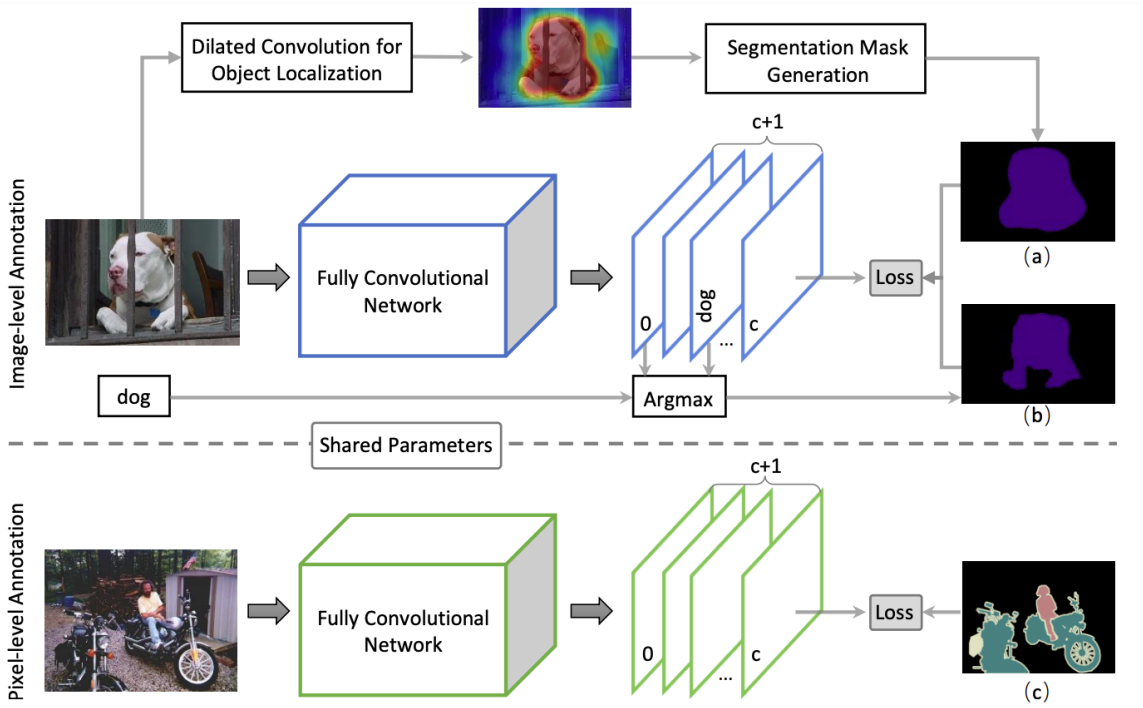


Figure 3.5: Diagram illustrating the semi-supervised semantic segmentation method proposed by Wei et al. [91]

3.2.2 Revisiting Dilated Convolution: A Simple Approach for Weakly and Semi -Supervised Semantic Segmentation (2018)

Wei et al. [91] proposed the semi-supervised semantic segmentation method that uses the dilated convolutions for object localization. The training framework includes two modules using shared Fully Convolutional Networks(FCNs) as the segmentation network backbone, shown in Figure 3.5:

The first module works on image-level labelled data. The proposed augmented classification network that has a VGG16 backbone with multi-dilated convolutional (MDC) blocks produces the dense localization map. The dense localization map is then further processed to generate the segmentation mask. This segmentation mask serves as supervision in the loss function with the segmentation mask produced by FCNs and the ground truth image-level label.

The second module is the normal supervised FCNs segmentation which employs pixel-level labelled data. The loss from two modules is combined to perform the semi-supervised learning. This method achieved 67.6% IoU on the PASCAL VOC 2012 dataset with 1400 strong labels and 9000 weak labels.

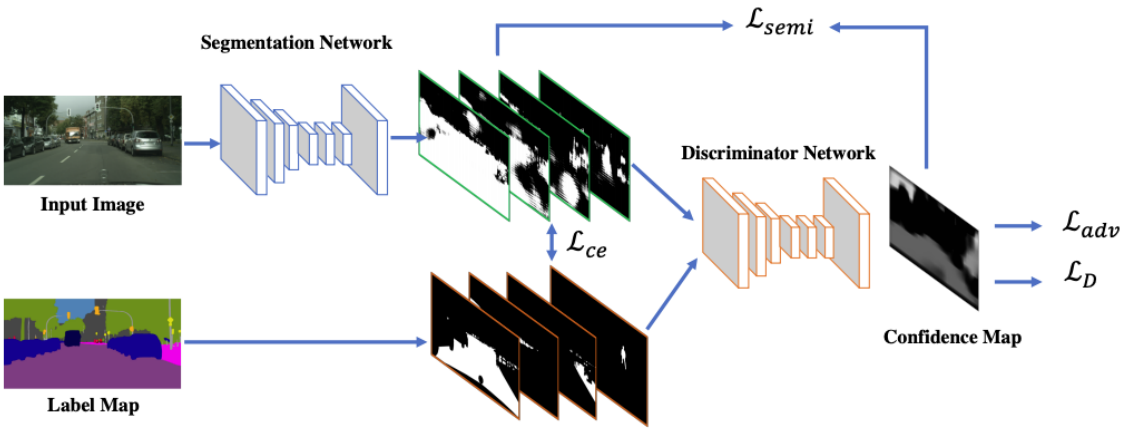


Figure 3.6: Diagram illustrating the Adversarial semi-supervised semantic segmentation framework [92]

3.2.3 Adversarial Learning for Semi-Supervised Semantic Segmentation (2018)

Exploiting the success of GANs [60], Hung et al. [92] proposed the adversarial learning for semi-supervised semantic segmentation. The framework contains two networks, segmentation network which is DeepLabv2 and discriminator network, shown in Figure 3.6. Instead of classifying an image as real or fake, the author designed the FCN discriminator to distinguish between the predicted probability maps from the ground truth segmentation maps. In the semi-supervised training process, the method uses strong labelled data and unlabelled data which is a new feature in this study. When using strong labelled data, the segmentation network is trained using two loss functions: Cross-entropy loss (L_{ce}) on the segmentation ground truth and adversarial loss (L_{adv}) to fool the discriminator. When using unlabelled data, the segmentation network is trained with Cross entropy semi-supervised loss (L_{semi}). Specifically, the predicted segmentation map of the unlabelled image produced by the segmentation network is passed through the discriminator network to obtain the confidence map. Then, in the context of self-training, this confidence map serves as a target for the previous segmentation prediction in the L_{semi} loss. Notably, the discriminator network is only trained with the strongly labelled data. The study achieved 69.5% IoU on PASCAL VOC 2012 dataset using 1400 strong labelled images and 9000 unlabelled images.

3.2.4 Semi-Supervised Semantic Segmentation with High and Low-level Consistency (2019)

Mittal et al. [93] proposed a semi-supervised segmentation framework that contains two branches which are illustrated on Figure 3.7. Their method is called the Semi-Supervised Semantic Segmentation GAN-based network (s4GAN) which improves the

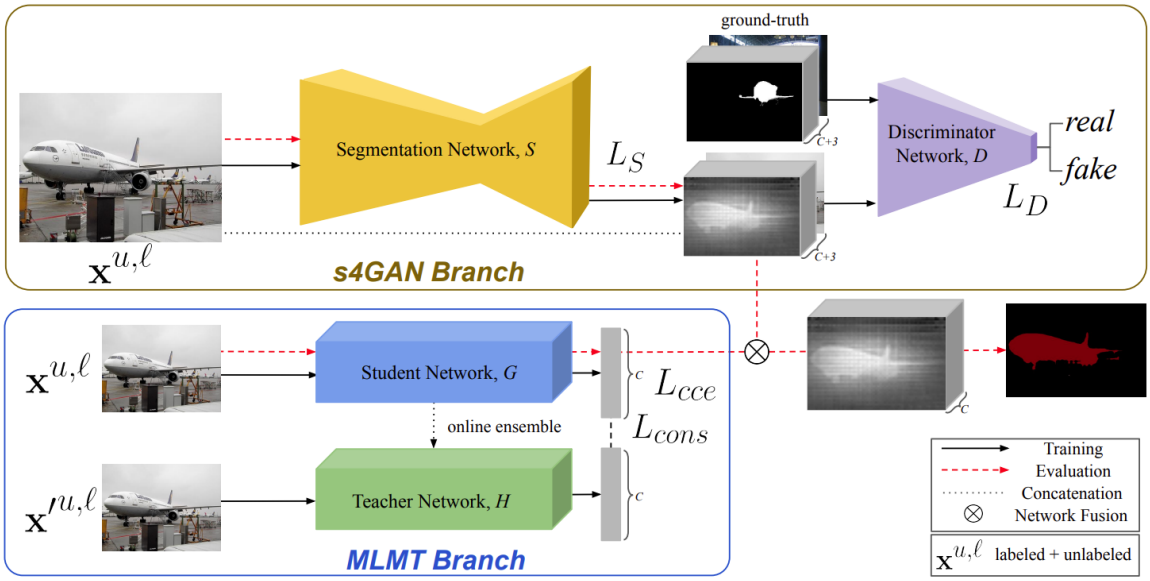


Figure 3.7: Diagram illustrating the semi-supervised semantic segmentation method proposed by Mittal et al. [93] called the s4GAN + Multi-label MeanTeacher classifier.

low-level details in the segmentation output using semi-supervised multi-label classification via the MeanTeacher approach. s4GAN exploits class-level information to remove false-positive predictions from the segmentation map. The s4GAN branch consists of a DeepLabv2 segmentation network and a discriminator network. The segmentation network acts as a generator to produce the segmentation maps from input images and is trained together with a discriminator network responsible for distinguishing the ground truth segmentation maps from the generated ones. The segmentation network is trained with three loss functions: the Cross Entropy loss applied on the pixel level labels; the feature matching loss and self-training loss applied on the unlabelled data. The discriminator network is trained with the original GAN loss: learns to differentiate between the real ground truth maps and the prediction segmentation maps of unlabelled data. The MLMT branch is the semi-supervised classification MeanTeacher network [26] which is backed by ResNet101 and works on both image-level labelled data and unlabelled data. Notably, the s4GAN and Mean Teacher network branches are trained separately.

The output of the Mean Teacher network is the predicted classification score for all classes. This is then used to refine the predicted segmentation maps from s4GAN to produce the final prediction segmentation mask. The prediction segmentation mask for a particular class is removed from the maps if its classification score is less than the predefined threshold. This method achieved 71.4% IoU on the PASCAL VOC data set using 1400 strongly labelled data samples and 9000 unlabelled data samples.

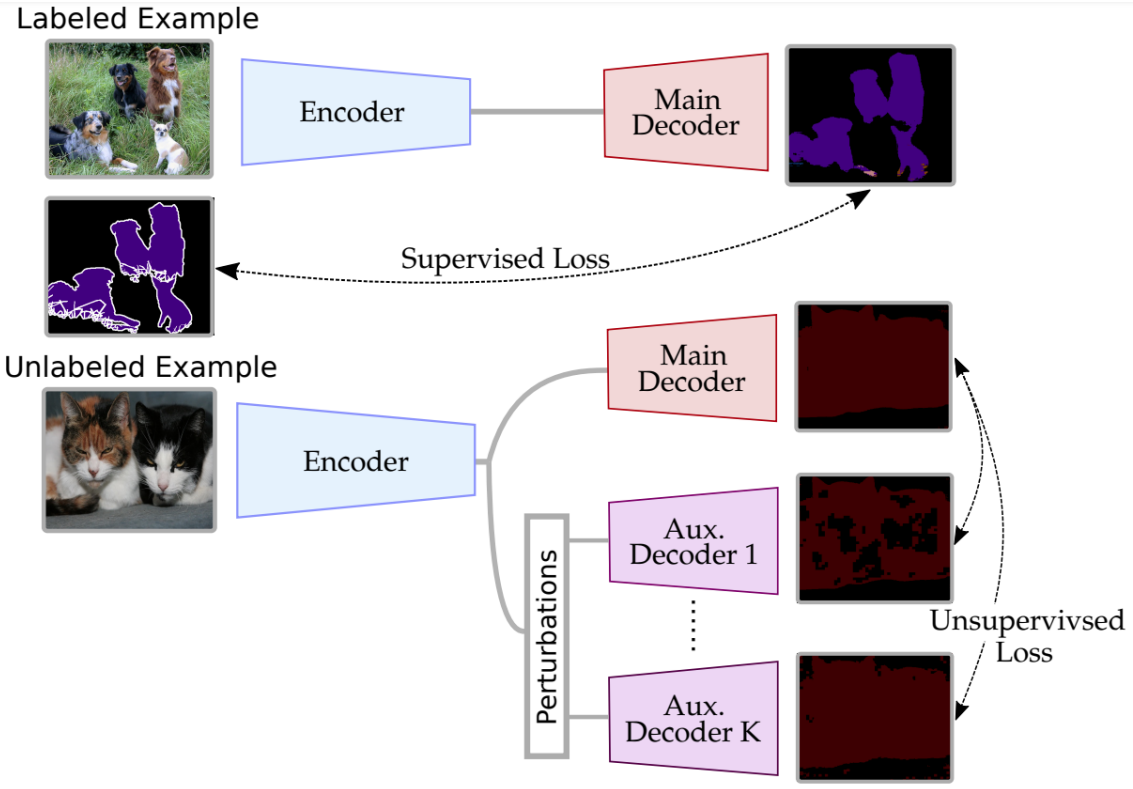


Figure 3.8: Semi-Supervised Encoder-Decoder Semantic Segmentation with Cross-Consistency Training [94].

3.2.5 Semi-Supervised Semantic Segmentation with Cross-Consistency Training (2020)

Ouali et al. [94] presented a cross-consistency based training method for semi-supervised semantic segmentation. The idea is to use the encoder-decoder as the segmentation architecture and perturbation-based approach, discussed in Section 3.1, for semi-supervised learning. Specifically, there is one encoder and one main decoder and multiple auxiliary decoders, as shown in Figure 3.8. The encoder and main decoder are trained using supervised Cross Entropy loss with pixel level targets. For the unlabelled examples, the encoder and all auxiliary decoders are trained using unsupervised loss by applying a consistency loss between the main decoder predictions and all auxiliary decoder predictions which are produced from different types of noise applied to the inputs of the auxiliary decoders. Notably, the unsupervised loss is not back-propagated through the main-decoder. The study established a new state of the result of 69.4% IoU on the PASCAL VOC 2012 dataset.

3.2.6 Semi-supervised semantic segmentation needs strong, varied perturbations (2020)

French et al. [95] proposed a perturbation based semi-supervised semantic segmentation method that uses the MeanTeacher framework [26]. The method used strong varied perturbations such as CutOut, CutMix. The segmentation network used is DeepLabv2 and DenseUnet trained on 2 losses: the supervised Cross Entropy loss on pixel-level labelled data and unsupervised consistency loss on unlabelled data. Specifically for the unlabelled data, the unsupervised consistency loss enforces the consistency between the target, which is the prediction segmentation maps of the Teacher network on the original image, and the predicted segmentation maps of the Student network. The input to the student network is strongly augmented data using methods such as CutOut and CutMix. With this simple semi-supervised framework, the study achieved 67.6% IoU on the PASCAL VOC 2012.

3.3 CNNs for medical image classification

Medical imaging has played a crucial diagnostic role in modern medicine. The common types of medical image can be divided as follows: Ultrasound which is the safest method for the patient's health and is generated by using sound waves; X-Ray which is the oldest technology and generated by using electromagnetic radiation; CT (Computer tomography) which builds the 3D image based on X-Rays; and MRI (Magnetic Resonance Imaging) which is generated by using a strong magnetic field and radio waves 3.9. Besides the four common imaging types listed above, there is another type of medical image used for skin lesions that generate dermoscopy images using skin surface microscopy. In addition, medical image is obtained to visualize the information of the internal human body in order to support pathologists, radiologists or clinicians to make a diagnosis about the disease. The final diagnosis of a patient's health should be made by doctors based on combined evidence from various processes and scans of the patient. Manually analysing medical images for multiple patients is a laborious, repetitive task and error prone task for doctors. Therefore automated diagnosis would help doctors work more efficiently. One of the most common underlying technologies of these systems is deep learning which leverages a substantial amount of medical imaging data. Deep convolutional neural networks (CNNs) are the dominant technique for this task. There are many applications of using deep CNNs in medical imaging tasks including tumor segmentation, cancer detection, cancer classification, image guided therapy, medical image annotation, and image retrieval [96]. To confine the scope in this section we only describe the application and research in the area of cancer classification using supervised CNNs. The result and objective of each study can vary depending on such factors as quantity and quality of the dataset, number of classes

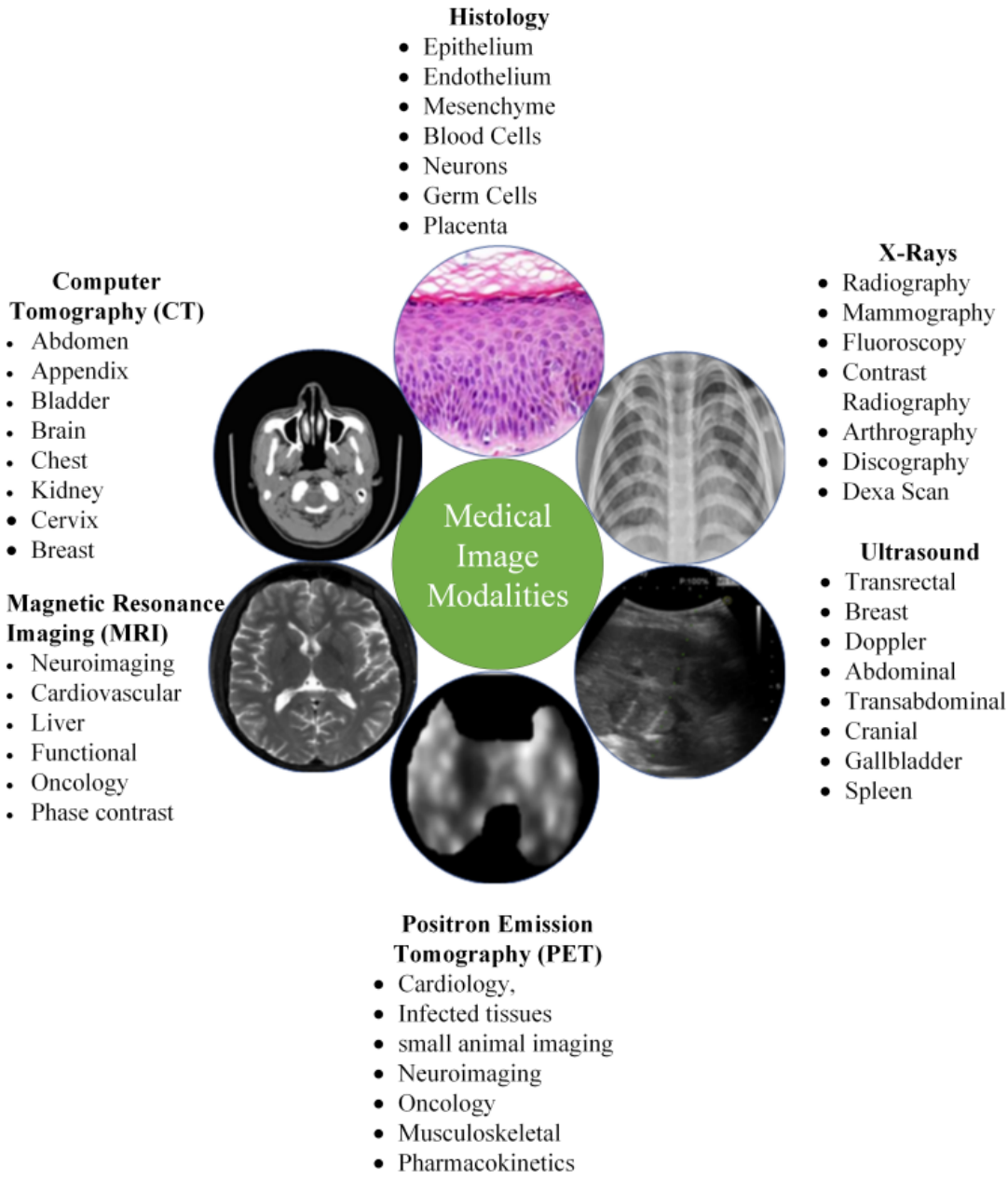


Figure 3.9: Medical imaging modalities [96].

to classify and CNN network architecture used.

3.3.1 Skin cancer classification

Han et al. [3] worked on classifying 12 skin diseases which include basal cell carcinoma, squamous cell carcinoma, intraepithelial carcinoma, actinic keratosis, seborrheic keratosis, malignant melanoma, melanocytic nevus, lentigo, pyogenic granuloma, hemangioma, dermatofibroma and wart. They used the ResNet152 CNN model [97] to train on roughly 19000 clinical images from the Asan, MED-Note and Alas dataset. They tested on the Asan test dataset which achieved 0.91, 86.4, 85.5 for AUC, Sensitivity, Specificity respectively. When using the Edinburgh test dataset, they achieved 0.89, 85.1, 81.3 for AUC, Sensitivity, Specificity respectively.

Amirreza et al. [2] experimented on dermoscopy images ([22]HAM10000: 10015, [98]PH2 : 120) which includes 8 types of skin diseases: melanoma, melanocytic nevi, basal cell carcinoma, benign keratosis, actinic keratosis and intraepithelial carcinoma, dermatofibroma, vascular lesions, and atypical nevi. They tried to use 4 different CNN models pre-trained on ImageNet including: Google's Inception v3 [99], InceptionResNetv2 [100], ResNet152 [97], and DenseNet201 [101]. As a result, they reported that the DenseNet201 model achieved the best accuracy among the 4 models in terms of micro and macro averaged precision (89.01% - 85.24%), F1 - Score (89.01% - 85.13%), and ROC AUC (98.79% - 98.16%), even outperforming expert dermatologists's performance.

Bi et al. [4] worked on the problem of classifying the 3 skin diseases of melanoma, seborrheic keratosis and nevus on 3600 dermoscopy images including 1600 from the ISIC Archive dataset [102]. The final CNN classification model is assembled from 3 separated classification models such that one outputs 3 skin disease classes and the two others output the binary classification of melanoma versus others and seborrheic keratosis versus others. The reported result on AUC is 91.5%.

Achim et al. [5] involved a combination of the deep learning and skin cancer experts to build a superior skin cancer classification algorithm. Firstly, a ResNet CNN model is trained on 11,444 dermoscopic images, obtained from HAM10000 and ISIC Archive, to classify 5 skin diseases. Then both the CNNs trained model and dermatologists of German university hospitals will classify 300 test biopsy-verified images. Finally, a gradient boosting method is used to produce a new classifier from the confident outputs of the CNN model and the dermatologists. This method achieved accuracy, sensitivity and specificity results of (82.95%, 89%, 84%), compared to standalone dermatologists test results of (42.94%, 66%, 62%) and CNN model results of (81.59%, 86.1%, 89.2%).

3.3.2 Digital pathology classification

Digital pathology is a field that uses computer-based technology to manage information generated specimens on slides. With the evolution of Whole Slide Images (WSI) technology which transforms the specimens on glass slides into digital high resolution images, known as histopathology image, digital pathology brings a new way to potentially help pathologist's diagnosis more efficiently by using deep learning. However, one of the challenges in applying deep learning, specifically deep CNNs on WSI, is the very high dimensionality of the input, for example millions of pixels. Therefore, besides configuration of the model, the successful deep CNNs on pathology task also heavily depend on WSI image preprocessing.

Stomach and colon cancers are common cancers that cause high numbers of deaths worldwide, and in a 2018 report, there are 782,685 and 551,269 deaths due to stomach

and colon cancer respectively. In addition, new research by Osamu et al. [6] used deep learning techniques to perform histopathological classification of stomach and colon cancer. There were three classes: adenocarcinoma, adenoma and non-neoplastic. The dataset consists of 4,128 and 4,036 whole slide images (WSIs) of the stomach and colon respectively collected from Hiroshima University and Haradoi Hospital. CNN Google's Inception v3 model is trained on millions of patches extracted from WSIs to classify the three classes above. Then a max-pooling approach and a recurrent neural network (RNN) is used to predict the final label for the particular WSI by aggregating all it's patch predictions. The model achieved the following AUC results, 0.97 and 0.99 for gastric adenocarcinoma and adenoma, 0.96 and 0.99 for colonic adenocarcinoma and adenoma.

Lung carcinoma is also a dangerous cancer that causes death to humans and the process of histologic pattern analysis in lung carcinoma is a very significant process in lung cancer diagnoses. Jason et al. [103] collected 422 WSIs from the Dartmouth-Hitchcock Medical Center in Lebanon.They used deep CNN Resnet architecture to classify histologic patterns on resected lung adenocarcinoma WSIs. The CNN model classifies patches extracted from the given WSI into histologic subtype patterns. Then, a heuristic is used to infer predominant and minor histologic patterns for the given WSI by aggregating its patch predictions. At the testing phase combined with experts, this method achieved a kappa score of 0.525 and an agreement of 66.6% with three pathologists, slightly higher than the inter-pathologist kappa score of 0.485 and an agreement of 62.7%.

Another health problem which also receives high attention from researchers is the breast cancer. Kun et al. [104] applied the pipeline framework, which uses the high-light from a heat-map to classify whether the WSI contains breast cancer metastases on the Camelyon-16 grand challenge dataset. Firstly, the deep CNN Google's Inception v3 architecture is trained to classify patches extracted from WSIs as tumor or normal. Then each WSI is slided into many patches and fed into the trained CNN to build the tumor probability heat-map. After that, a Support Vector Machine classifier is trained on WSIs's heat-map to finally produce the label for the entire WSI. As a result, this system achieved 90.23% of AUCs.

3.3.3 X-ray Classification

X-ray image is one of the most frequently used medical imaging modality. It is used for diagnosing patients because it is cheap and quick to obtain. Therefore using deep learning to diagnose disease from X-rays is well motivated.

Worawate et al. [7] performed binary classification on the presence of lung cancer. They used a pre- trained deep CNN DenseNet architecture to firstly retrain on the ChestX-ray14 dataset (112,120 non-cancer chest X-ray images) and then fine-tuned on the JSRT dataset (247 chest X-ray images that have 100 lung cancer) in order to

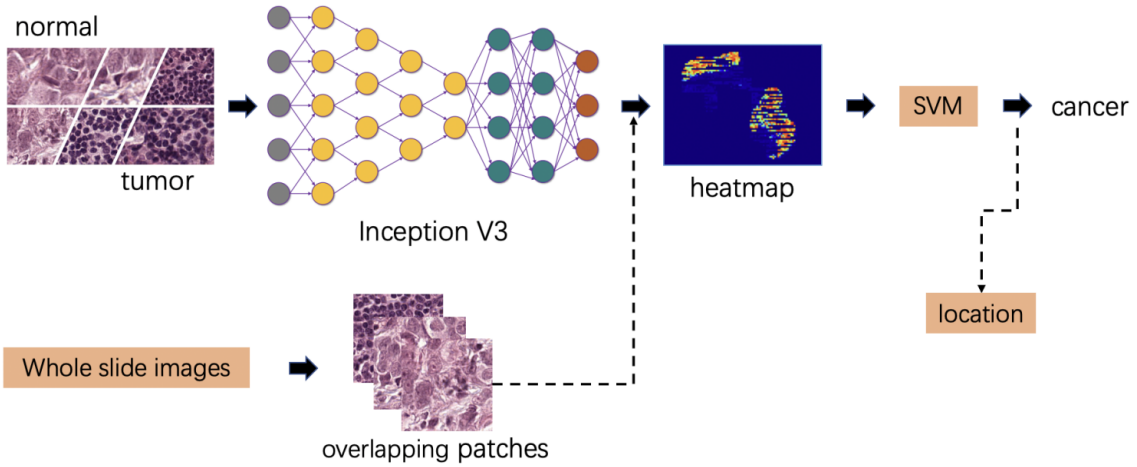


Figure 3.10: [104] The framework of cancer metastases detection.

alleviate the problem of small samples in the target classification task. The proposed method yielded 74.43% of mean accuracy, 74.96% mean specificity and 74.68% mean sensitivity.

Lung pneumonia is a cause of death in humans. In 2017 a group from Stanford University [8] proposed CheXNets, which uses a DenseNet 121 trained on ChestX-ray14 dataset to classify X-ray images into 14 disease classes of lung pneumonia. In the testing phase, the model achieved 0.435 F1 score which outperformed the F1 score of 0.387 achieved from averaging 4 scores from radiologists.

3.4 Semi-supervised learning in medical imaging classification

In the real world, doing annotations for medical imaging usually is very time consuming for the doctors, especially when the diagnosis of multiple doctors needs to be averaged. Therefore, it is practical to apply semi-supervised learning on medical imaging classification that alleviate the problem of small amounts of labelled medical images and high availability of unlabelled medical images.

For skin cancer classification on ISBI and PH2 dataset, Xin et al. [13] used **Categorical Generative Adversarial** networks [89] in a semi-supervised manner to solve two tasks, the first one is to classify the real image into melanoma or benign class, the second one is to generate synthetic images that assist the training. With only 140 labelled images, the method yielded an average precision of 0.424.

In another study of skin cancer classification, Antonia et al. [14] proposed a SSL approach called **Denoising Adversarial AutoEncoder** that combines the Adversarial Net and AutoEncoder Net. Unlabelled samples are utilized in the generative and decoder part to help the model learn the representation of skin cancer such as color, shape and texture. labelled samples are encoded by the encoder and then are

classified as either benign or malignant by the discriminator. Due to the heavy class imbalance (0.9,0.1), the classification loss is weighted for each class. On the ISIC dataset with 5000 labelled samples, the SSL approach achieved 0.82 for sensitivity and 0.85 for specificity.

In the study of nucleus classification task, Hai et al. [15] integrated the current state of the art SSL approach of Mean Teacher [26] and the graph, which is generated by the Label Propagation algorithm [105], that transfers the label information from labelled samples to unlabelled samples. Then the student model is learnt from classification loss, consistency loss and Siames loss which is based on the graph. On the MoNuseg dataset of 22462 nuclei including 4 types of nucleus Epithelial, Inflammatory, Fibroblast and Miscellaneous, the model yielded 75.79% for F1 score using only 10% of the labelled samples from the dataset.

Moreover, Wenkai et al. [16] combined Semi-supervised Generative Adversarial Nets sGANs, Conditional Generative Adversarial Nets cGANs and Semi-Supervised Support Vector Machine S3VM to construct the model called DScGANs (dual-path semi-supervised conditional generative adversarial networks) to solve the thyroid nodule classification task. Notably, the DScGANs is trained under conditional constraint Domain Knowledge (DK) which is acquired from the processed patches called OS, which has 225 pixel value for nodule region and 0 pixel value for non-nodule region, based on consultation with experienced radiologists. The role of DK can be summarized as follows: provide the auxiliary information to help the generator improve image quality; make the connection between information of labelled data and unlabelled data; and acts as a condition to constrain the S3VM to classify the lesion image as benign or malignant. With only 35% labelled samples from 3090 ultrasonography thyroid nodule lesion images, the proposed method achieved 90.5% for accuracy and 91.4% for AUC.

3.5 Semi-supervised learning in medical semantic segmentation

For semantic segmentation of medical images, obtaining pixel-level annotation of 2D images or 3D volumes by medical experts is costly and time-consuming, leading there to be much more unlabelled data compared to labelled data. Therefore, semi-supervised learning is a promising learning framework for medical images that leverages a limited amount of pixel-level labelled data and a large amount of unlabelled data. There have been many semi-supervised frameworks proposed for semantic segmentation of medical images. They can be grouped into 2 broad types: adversarial training [106, 107, 108, 17] and consistency training [109, 110, 111]. These methods use the same basic methodology as those used for general semantic segmentation methods that were described in Sections 3.2.3, 3.2.4 for adversarial training and Sec-

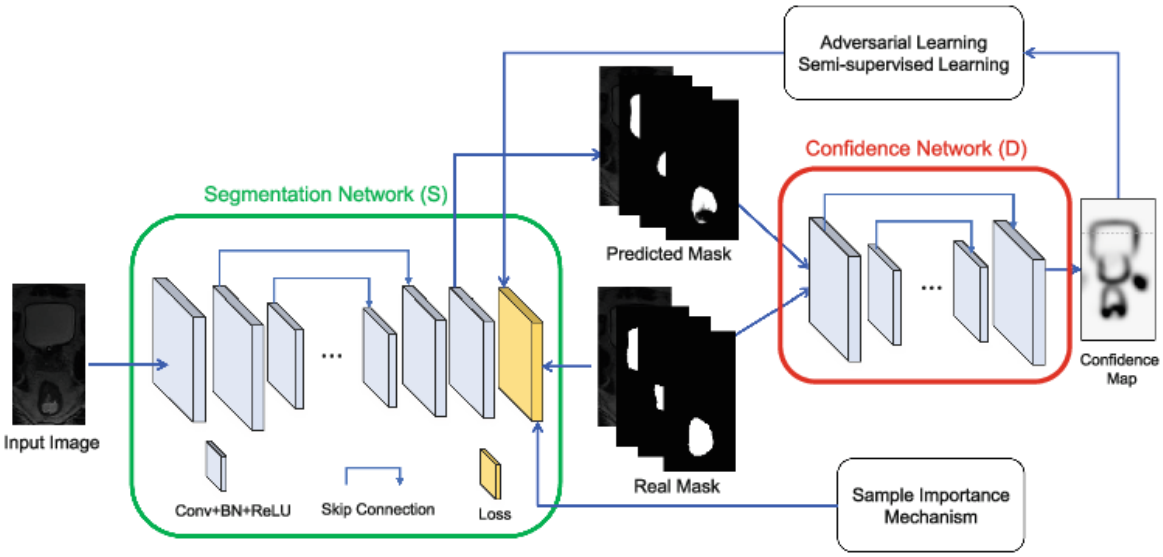


Figure 3.11: Illustration of ASDNet [106].

tions 3.2.5, 3.2.6 for consistency training.

3.5.1 Semi-supervised adversarial training in medical semantic segmentation

Nie et al. [106] proposed Attention Based Semi-supervised Deep Networks (ASDNet). Their experiments were conducted on 50 labelled and 20 unlabelled T2-weighted MRI images of prostate cancer patients from a cancer hospital. The problem was to segment the prostate, bladder and rectum. As shown in Figure 3.11, the solution consists of two networks: the segmentation network and the confidence network. The segmentation network is a simplified V-Net that outputs a predicted segmentation mask. The confidence network is a fully convolutional discriminator. The Segmentation Network is trained on three losses: the supervised loss which is a newly proposed Sample Attention Multi-class Dice loss that is designed to alleviate the problem of class imbalance and dominance of easy samples; the Binary Cross Entropy adversarial loss which improves the Segmentation Network to produce segmentation masks that are more consistent with ground-truth segmentation masks in order to fool the Confidence Network; the semi-supervised loss on unlabelled data which is the multi-class Dice loss on the segmentation map and its confident map processed by the Confident Network. The confidence network is trained on the Binary Cross Entropy adversarial loss using the segmentation output from the Segmentation Network and its corresponding ground truth.

Li et al. [17] proposed a shape aware semi-supervised segmentation method on 3D atrial gadolinium-enhanced MRIs. The term "shape aware" is used to describe the segmentation network's ability to capture the global shape of each object class more effectively. To do so, the authors designed a segmentation network with 2 heads, one head produces a segmentation map and the other head produces a signed distance map

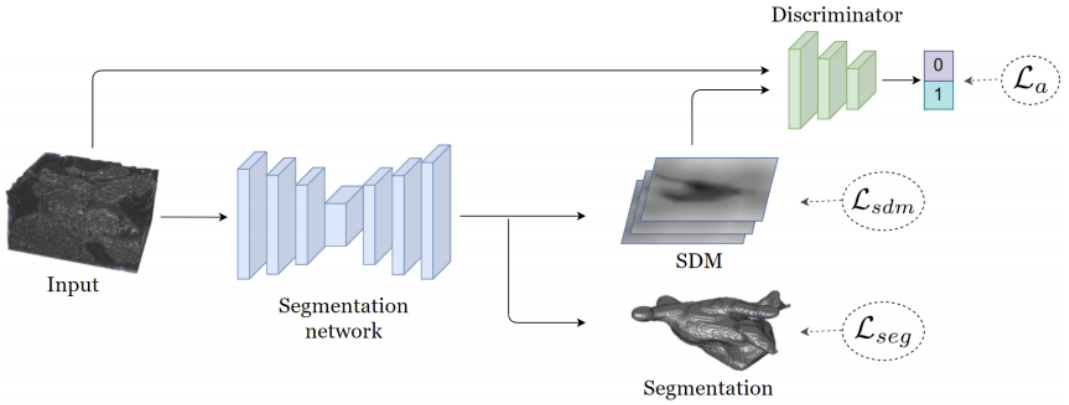


Figure 3.12: Shape aware semi-supervised semantic segmentation of medical images [17].

(SDM) which assigns the signed distance value to the nearest boundary of a target object for each pixel. The segmentation network with V-Net backbone is trained on two losses, shown in Figure 3.12 and described as follows:

- For labelled samples, the supervised loss consists of the Dice loss on the predicted segmentation map and its corresponding ground truth mask; the Mean Squared Error on the predicted SDM and its corresponding ground truth SDM. Notably, the ground truth SDM is generated from the ground truth segmentation mask.
- The adversarial loss induced from the fixed discriminator network aims to distinguish the predicted SDM from labelled samples or unlabelled samples. In particular, the discriminator network (CNN binary classification like) receives the fusion of predicted SDM and its 3D original images from labelled or unlabelled data and then produces the class probability of being labelled data. As a result, learning from adversarial loss helps the segmentation network obtain better shape-aware features that generalize well to the unlabelled data.

3.5.2 Semi-supervised consistency training in medical semantic segmentation

Bortsova et al. [109] proposed a semi-supervised semantic segmentation method that uses consistency training to segment left and right lung fields, left and right clavicles and the heart from chest X-rays. The segmentation network uses a U-Net backbone and follows a Siamese architecture with two identical branches, as shown in Figure 3.13. Each labelled or unlabelled sample is transformed by 2 random data augmentation methods. Then each transformed version is fed into two branches of the segmentation network to obtain two predicted segmentation maps. The network is trained using the supervised loss on labelled data and unsupervised consistency loss on both labelled and unlabelled data. To perform the consistency loss, the first

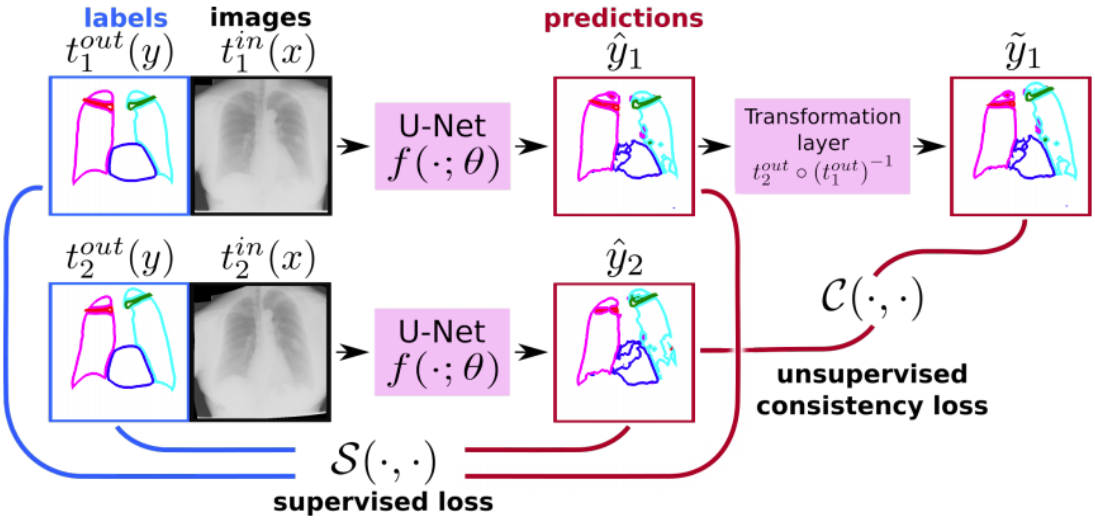


Figure 3.13: The training pipeline of the semi-supervised semantic segmentation method of [109] that uses consistency training.

predicted segmentation map is processed by a differentiable transformation layer in order to align it with the second predicted segmentation map.

Yu et al. [111] proposed another semi-supervised consistency training model for 3D Left Atrium Segmentation with the uncertainty-aware feature. The authors used Mean Teacher as the semi-supervised framework and V-Net as the segmentation network, as illustrated in Figure 3.14. The student segmentation network is learnt from the supervised loss which is based on labelled samples and the proposed uncertainty-aware consistency loss which is based on both labelled and unlabelled samples. The uncertainty-aware consistency loss enforces the consistency between predicted segmentation map from the student network and estimated uncertainty segmentation map with Monte Carlo Dropout technique from the teacher network as the target. Using estimated uncertainty on the prediction of the teacher network helps the student network learn from the more reliable targets.

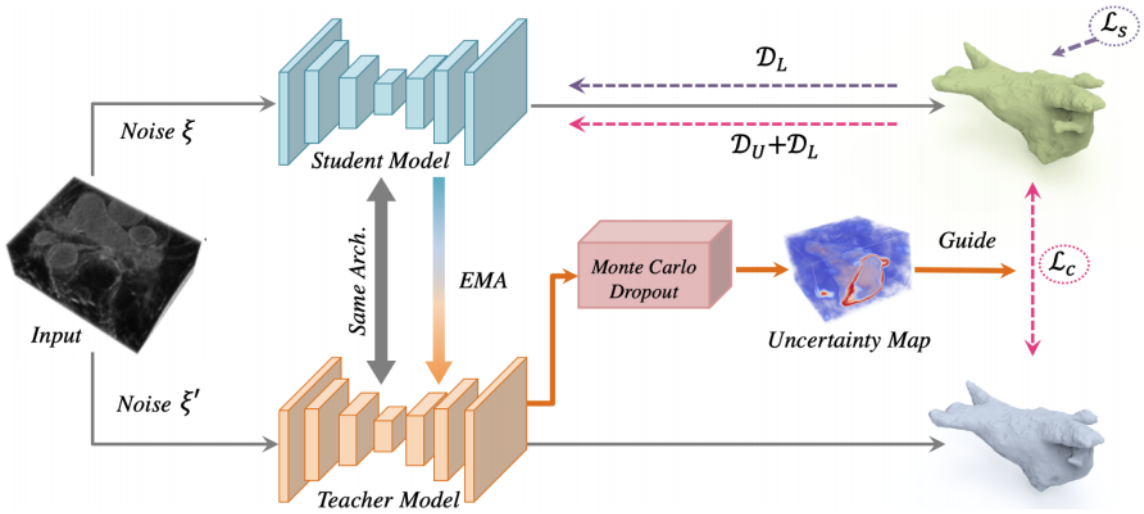


Figure 3.14: Diagram illustrating semi-supervised the consistency training model with the uncertainty-aware feature [111].

Chapter 4

Problem Definition

As a result of our thorough literature review, we found there is no prior work addressing the class imbalance problem in semi-supervised learning (SSL) of medical images for either classification or segmentation. Therefore in this thesis we will address the class imbalance problem for both semi-supervised medical image classification and segmentation.

Semi-supervised learning for medical images is of practical importance. Large numbers of labelled training images are necessary for training highly accurate deep learning classification and segmentation models. However, it is costly and time-consuming for medical experts to manually annotate the required number of images at either image-level or pixel-level. Hence, semi-supervised learning is an ideal method to tackle this obstacle. It only requires a small portion of all scanned images to be labelled and automatically exploits statistical patterns from the rest. Medical image datasets often contain studies that involve imbalanced datasets which the model trained on often favour the majority image class (usually negative cases), such as for the HAM10000 skin cancer dataset [22] where the majority of the images are classified as benign versus a small number of malignant images, for ChestX-ray14 dataset [112] where the majority of the images are for healthy lungs versus unhealthy lungs, or for the Nerve Ultrasound segmentation dataset [24] where the number of background pixels is heavily larger than foreground pixels. Hence in this study we address the class imbalance issue for semi-supervised classification and segmentation of medical images.

As stated earlier our problem of class imbalance in semi-supervised learning for medical images has not been explored directly in the past. There are still several studies of SSL medical images on skewed data distributions [13, 14, 106, 108] but they are not focused on addressing the class imbalance problem. There has also been work addressing the class imbalance problem for supervised learning such as resampling [65, 66, 67, 68, 69, 70, 47, 84], distribution-based loss [72, 73], region-based loss [85, 86], cost-sensitive learning [74, 75, 76], threshold moving [77, 78, 79] and using a pre-compute weight map for ground truth [46], but none of these works study SSL.

We found one study [82] that proposed a new loss function for addressing the class imbalance issue for SSL. However, the authors conduct their experiments on standard image datasets and as such the efficacy of their proposed approach remains unproven on medical image datasets where minimising the number of false negative predictions is extremely important. Furthermore they artificially made their datasets imbalanced by introducing synthetic skew into the class distribution. In contrast, our study will use a medical image dataset which contains a real skewed class distribution.

More formally we define our problem as follows. Given a set of n labelled training examples $L = \{l_1, l_2, ...l_n\}$ and a set of m unlabelled training examples $U = \{u_1, u_2, ...u_m\}$, with $m >> n$. The class distribution of labels in L and U are highly skewed, so both class distributions have low entropy. We further assume that samples in both L and U are drawn from the same population, and hence the class distributions of L and U are similar.

For the image classification task, the aim is to train a model using both L and U such that the model achieves high average recall for all classes (the standard metric for judging the success of models on class imbalanced datasets). Hence we aim to maximize the unweighted average recall (UAR) metric on a test data set of z examples $T = \{t_1, t_2, ...t_z\}$. UAR is defined as follows:

$$\text{UAR} = \frac{\sum_{C=1}^{C} Recall_C}{C}$$

where C is the number of classes.

For the image segmentation task, the aim is to train a model using both L and U such that the model achieves high Dice Coefficient score for minor class on a test data set of z examples $T = \{t_1, t_2, ...t_z\}$. Dice Coefficient is defined as follows:

$$\text{Dice} = \frac{2 * TP}{2 * TP + FP + FN}$$

Further details of the UAR and Dice Coefficient can be found in 6.3.

We assume that T has the same class distribution as L and U. By using the UAR metric, we evaluate the model based on the contribution of all classes equally rather than favoring the major classes. Furthermore, it will make sure the minor classes which are really important for us (normally ones with disease) get an equal share of the error metric. Similarly the Dice Coefficient fairly measures performance of every class regardless of class distribution skew.

Justification for assuming labelled and unlabelled data have the same class distribution: Our new loss function mainly depends on the information of the predicted class's frequency of unlabelled data. We can not measure the class distribution of the unlabelled dataset. Therefore, in our study, we assume that the class distribution of the unlabelled dataset is the same as the labelled dataset. This assumption is also supported by statistical theory. That is, the data distribution of a

population can be represented by the data distribution of a uniform random sample from the population. In our problem, the labelled dataset is a sample taken from the population. We can use its class distribution to represent the population. Hence, it is reasonable to assume the unlabelled data has the same class distribution as the labelled data for a sufficiently large number of samples.

Chapter 5

Methodology

5.1 Semi-supervised learning architecture

We will base our research on perturbation based semi-supervised learning (SSL) methods as described in Section 3.1 since all state of the art SSL methods [1, 26, 27, 28, 29] use this approach. The solution we developed for this study can be applied to any perturbation based SSL that uses the consistency loss. To simplify our analysis we will focus on a particular perturbation based SSL called the Unsupervised Data Augmentation [1] (UDA) method. We describe how UDA works in detail in Section 5.1.1. In our experiments we also implemented our proposed method for addressing class skew on top of the Mean Teacher SSL method. Hence in Section 5.1.2 we describe how the Mean Teacher method works in detail.

5.1.1 Unsupervised Data Augmentation (UDA)

UDA is one of the best performing recent SSL methods. Figure 5.1 shows a diagram of how UDA works. The model is trained using two losses: a supervised loss (cross-entropy loss) and an unsupervised loss (consistency loss). The aim of consistency loss is to enforce the consistency of two prediction distributions. The key idea of UDA is to use optimal data augmentation on unlabelled samples to increase the effectiveness of the consistency loss. To obtain optimal data augmentation they applied an algorithm called RandAugmentation [63] on the labelled dataset. The total loss for the UDA architecture consists of two terms (see Figure 5.1): supervised loss for labelled data and consistency loss for unlabelled data. The loss formula can be summarized as follows:

$$L = L_S(p_\theta(y|l)) + L_{con}(p_\theta(y|u), p_\theta(y|\hat{u})) \tag{5.1}$$

Where L_S is a supervised cross entropy loss function that takes input as the predicted probability distribution $p_\theta(y|l)$ of y for a labelled sample l produced by the model M with parameters θ. L_{con} is a consistency loss function that uses the Kullback-Leibler divergence to steer the predicted class distribution of the augmented unlabelled image

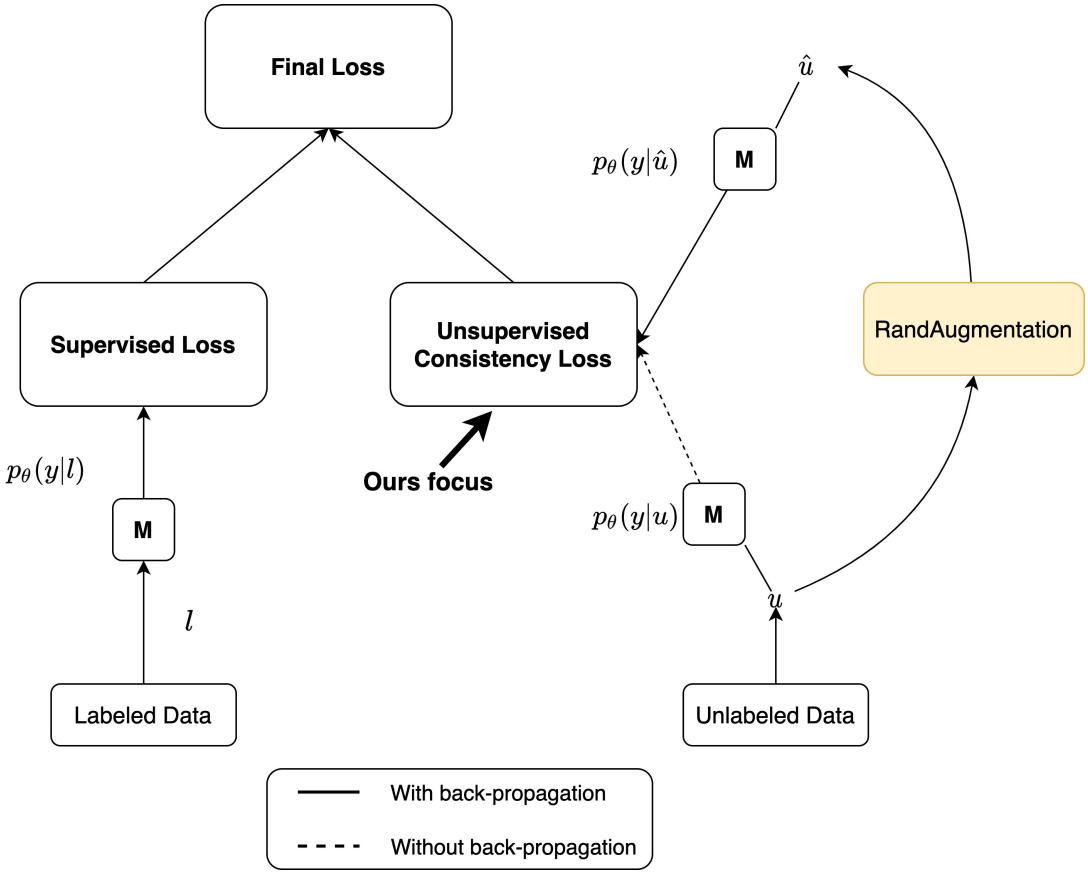


Figure 5.1: The architecture used by the Unsupervised Data Augmentation (UDA) [1] perturbation SSL method. In the diagram M is the shared CNN model used for classifying both the labelled and unlabelled images.

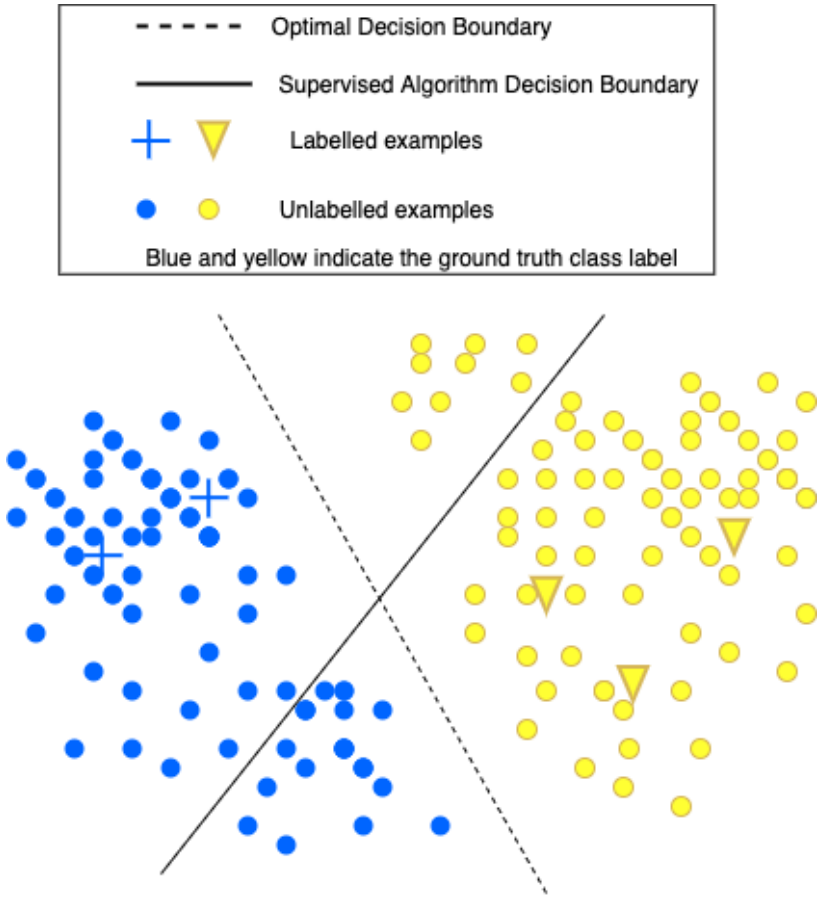


Figure 5.2: [33] Illustration of the decision boundary based on the smoothness and low-density assumption.

$\hat{u}$ towards the target predicted class distribution of the original unlabelled image u.

As we discussed in the related works chapter, perturbation based methods such as UDA are based on the smoothness and low-density assumption. With the smoothness assumption, two data points which are close in the input space should have the same label. That means, a labelled sample and an unlabelled sample who are close to each other will have the label information propagated to the unlabelled sample. Another important property of the smoothness assumption is that the original input and the augmented input should be close to each other in the embedded space and hence should be assigned the same label. This idea is captured by the consistency loss L_{con} which ensures the model produces the same predicted probabilities between the original and noise input, leading to the model being robust to noise.

The low-density assumption is implicitly followed by the perturbation method due to the relation with the smoothness assumption [33]. Figure 5.2 illustrates the decision boundary based on the smoothness and low-density assumptions. Data points having the same label should be relatively close to each other, thus becoming a cluster with high density. Therefore, under the low-density assumption, the decision boundary should not lie in the high density region of the embedding.

There have been various existing approaches [65, 66, 67, 68, 72, 73, 74, 75, 76, 85, 86, 46, 47, 84] to tackle class imbalance for the supervised learning problem. These

approaches are complementary to our solutions since they work on improving supervised loss while ours works to improve the unsupervised loss component of the overall loss. The unsupervised loss is particularly important since the number of unlabelled examples is typically much larger than labelled examples. As shown in Figure 5.1, UDA uses the consistency loss to exploit the information from the unlabelled loss by making the class distribution of the augmented unlabelled data match the original unlabelled data. Hence to alleviate the class imbalance problem in UDA, we focus on modifying UDA's consistency loss (shown in Equation 5.1). In particular, UDA's consistency loss is replaced by our new novel loss function, Adaptive Blended Consistency Loss (ABCL). Hence the total loss is reformulated with ABCL replacing the unsupervised term:

$$L = L_S(p_\theta(y|l)) + ABCL(p_\theta(y|u), p_\theta(y|\hat{u})) \tag{5.2}$$

5.1.2 Mean Teacher

Mean Teacher [26] is another well-known perturbation based SSL method which is the underlying framework for many semi-supervised classification and segmentation studies. As discussed in Section 3.1, the framework consists of the dual teacher with weights θ_T and student with weights θ_S models which produce two predictions from all labelled and unlabelled inputs $X = L \cup U$ by applying two types of stochastic noise. Following UDA and the study of semi-supervised semantic segmentation using the Mean Teacher method [95], the teacher model takes original examples x as the input and the student model takes augmented examples $\hat{x}$ as the input. Figure 3.2 illustrates the learning framework. The weights θ_S of the student model is updated by the supervised loss on only labelled data of student's output and the consistency loss which is the distance between student's output from the input x and teacher output from the input $\hat{x}$. Formally, the total loss of student model is formulated as follows:

$$L = L_S(p_{\theta_S}(y|l)) + L_{con}(p_{\theta_T}(y|x), p_{\theta_S}(y|\hat{x})) \tag{5.3}$$

Where L_S is a supervised loss function that receive the predicted $p_{\theta_S}(y|l)$ probability of y when given labelled samples l produced by the student model with parameters θ_S. L_{con} is a consistency loss function that minimizes the difference between the target probability distribution $p_{\theta_T}(y|x)$ of y for given original samples x produced by the teacher model with parameter θ_T and the $p_{\theta_S}(y|\hat{x})$ probability distribution of y, given augmented samples $\hat{x}$ predicted by the student model with parameter θ_S. Notably, the teacher model with parameter θ_T is not learnable, instead its parameter is updated by the exponential moving average of the student's parameter over each batch training, which computed as follows:

$$\theta_T^t = \alpha\theta_T^{t-1} + (1-\alpha)\theta_S^t \tag{5.4}$$

Where α is a predefined smoothing coefficient hyperparameter and t is the training step. This consistency loss can also be replaced with our Adaptive Blended Consistency Loss (ABCL) to tackle class imbalance in semi-supervised learning. Hence the total loss is reformulated as follows:

$$L = L_S(p_{\theta_S}(y|l)) + ABCL(p_{\theta_T}(y|x), p_{\theta_S}(y|\hat{x})) \tag{5.5}$$

5.2 Issues with existing consistency loss formulations

In this section, we analyse problems with standard consistency loss (CL) in UDA [1] and the state-of-the-art Suppressed Consistency Loss (SCL) [82] when training on datasets with imbalanced class distributions. The problems will be analysed in the context of the original sample's prediction (OSP) and augmented sample's prediction (ASP) which represent the probability distributions $p_\theta(y|u)$ and $p_\theta(y|\hat{u})$ respectively in Equation 5.1

Figure 5.3 illustrates how CL and SCL works. In UDA, the CL is a function that sets the OSP as the target for ASP. That is the CL always pushes the class distribution of ASP towards the class distribution of OSP. The idea behind SCL is to suppress the minor class‘s consistency loss to move the decision boundary such that it passes through a low-density region of the latent space. In practical terms, SCL suppresses the CL when the OSP is the minor class and applies the CL when the OSP is the major class. Like CL, SCL uses the OSP's class distribution as the target irrespective of whether OSP and ASP class distributions belong to the major or minor class.

Both standard CL and SCL have two shortcomings. Firstly they are both biased towards targeting the major class in the presence of imbalanced training data. Secondly they do not target a blend of OSP and ASP but instead always target OSP only, and thus do not exploit the augmented example to improve the model's behaviour for the original example.

Shortcoming 1: CL and SLC are more likely to target the major class. When in doubt the model will more often predict the major class since the model is trained on labelled data which is skewed towards major class samples. Consequently, samples of the minor class are more likely to be mispredicted as the major class than vice versa. In particular, when the original sample is incorrectly predicted as major class and the augmented sample is correctly predicted as minor class, the CL and SCL erroneously encourages the model to predict the sample as major class. As a result, the model's performance will degrade for minor class samples.

Shortcoming 2: CL and SLC do not target a blend of OSP and ASP but instead always targets OSP only. Targeting a blend of OSP and ASP reduces the harmful effects of targeting OSP only when OSP predicts the wrong class.

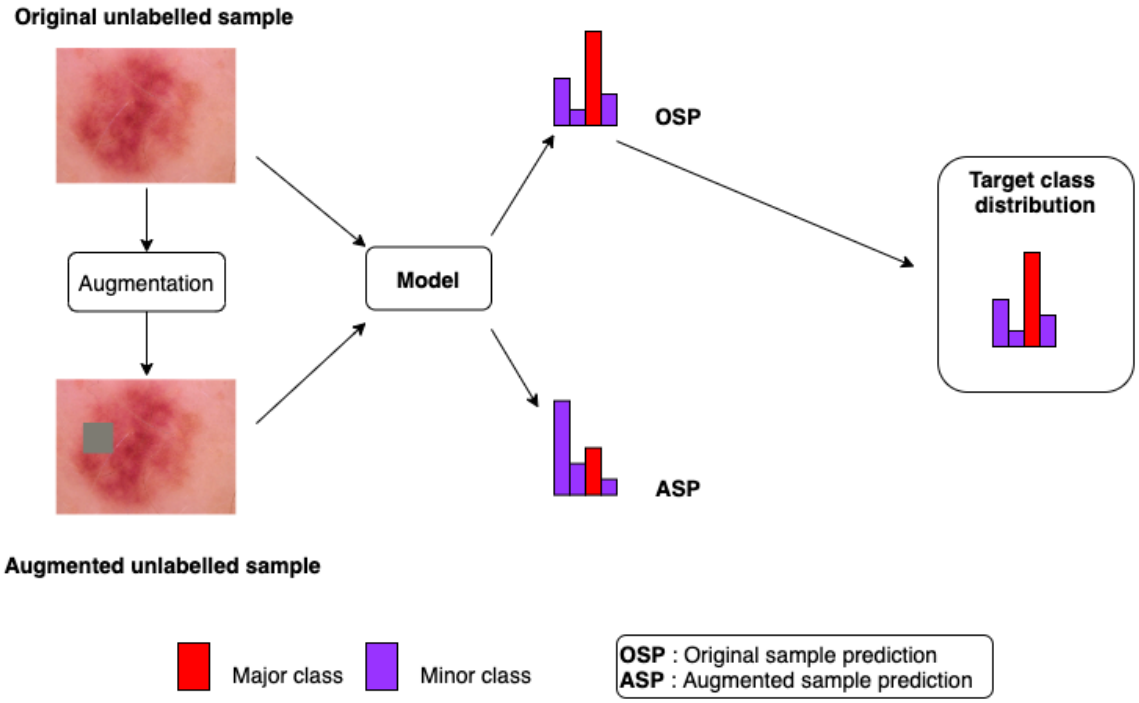


Figure 5.3: An illustration showing how CL and SCL works. Note that the target class distribution is always the class distribution for OSP. In addition, SCL down-weights the loss if the predicted class of OSP is the minor class.

Unlike existing methods, we do not make the assumption that the model's prediction for the original sample is more accurate than for the augmented sample. Instead we set the target as a blend of OSP and ASP to potentially average out some of the harmful effects of a wrong OSP prediction. This is in some ways similar to the benefit of ensembling the prediction of multiple models to minimize prediction error and the established technique of test-time data augmentation [35].

The shortcomings mentioned above are all centered around what we should set as the target distribution for the CL as a function of OSP and ASP. Hence, to provide a more detailed analysis of the desired target class distribution we divide the analysis into four separate cases depending on whether major or minor classes were predicted by the OSP and ASP.

See Table 5.1 and Figure 5.4 for an illustration of the 4 different OSP and ASP cases. In cases 1 and 2, OSP and ASP are in agreement. Unlike CL and SCL, we forgo the assumption that the OSP is in any way more valid than the ASP. As a result, we posit that it may be better to move the desired target class distribution to be a blend of OSP and ASP instead of just to OSP. This is because we would like the target to contain information in the predictions of both OSP and ASP. In case 3 CL and SCL may unintentionally encourage the bias towards the major class to be stronger since it is moving the desired target class distribution towards the major class although there is no consensus between the two predictions. There is already a natural tendency to predict the major class, a bias which is induced by the dataset skew. CL/SCL does nothing to counteract this bias as it does not consider the frequencies of the predicted

Case	OSP	ASP	CL target	SCL target	ABCL target
1	Major class	Major class	OSP (major class)	OSP (major class)	Near the middle between OSP and ASP
2	Minor class	Minor class	OSP (minor class)	OSP (minor class)	Near the middle between OSP and ASP
3	Major class	Minor class	OSP (major class)	OSP (major class)	Closer towards ASP
4	Minor class	Major class	OSP (minor class)	OSP (minor class)	Closer towards OSP

Table 5.1: The analysis of the target class distribution of CL (standard consistency loss), SCL (suppressed consistency loss), and our ABCL (adaptive blended consistency loss) for 4 prediction cases. Figure 5.4 gives a diagrammatic illustration of the 4 cases.

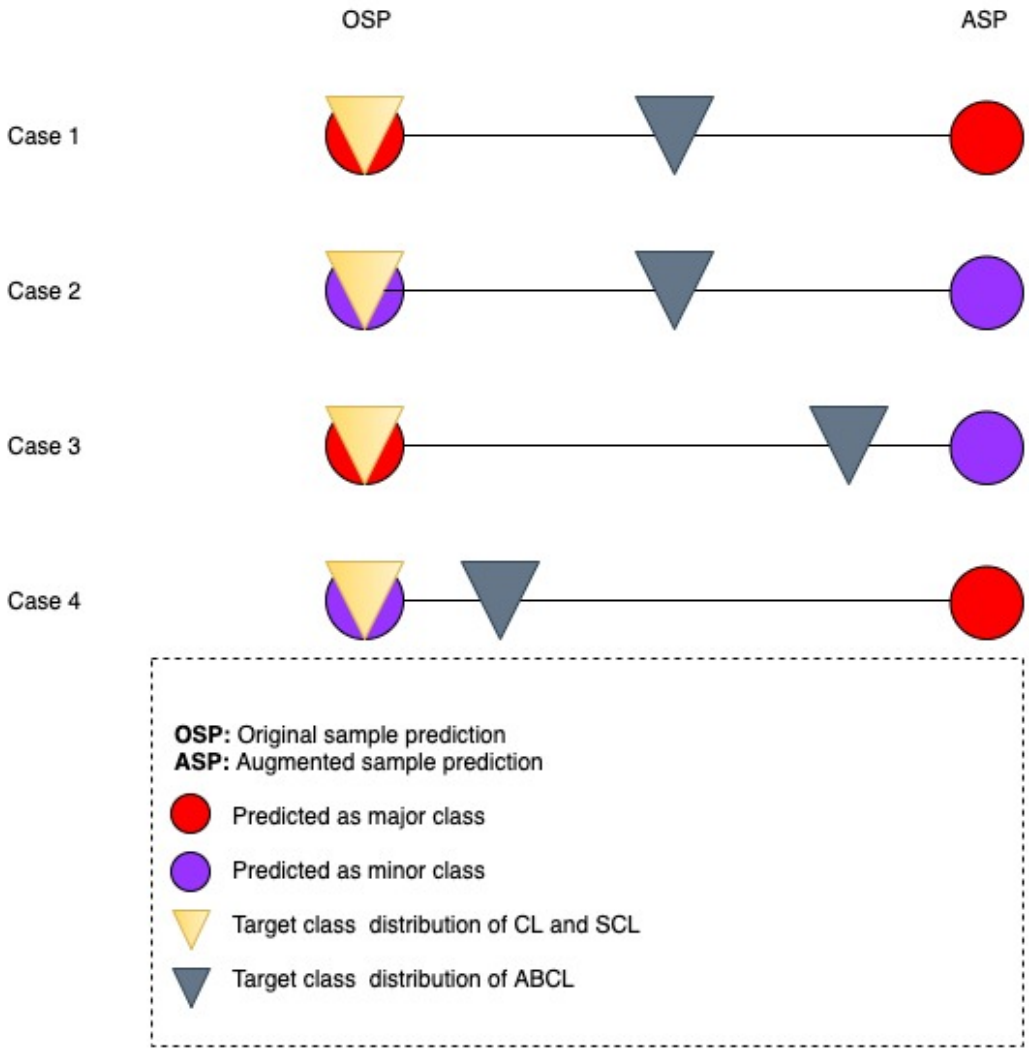


Figure 5.4: An illustration of the target class distribution of CL, SCL and ABCL for the 4 cases shown in Table 5.1.

classes. Case 3 presents an opportunity to counteract this bias, as the model has already indicated via the ASP that the minor class is likely correct since it made that prediction despite its natural tendency to predict the major class. Unfortunately, using CL/SCL in this situation is likely to encourage the model to mispredict the minor class sample as the major class. It would be preferable to instead move the desired target class distribution towards ASP, thus counteracting the dataset bias. In case 4, OSP is the minor class, so there is no bias towards the major class and so both CL and SCL do a good job of rewarding the minor class prediction in this case.

5.3 Adaptive Blended Consistency Loss (ABCL)

To address the drawbacks of CL and SCL, the desired target class distribution for the consistency loss should be adaptively adjusted to be somewhere between OSP and ASP depending on which of the 4 cases in Table 5.1 and Figure 5.4 has occurred. In cases 1 and 2, OSP and ASP both are either the major or minor class. In this case the desired target class distribution should be in the middle of OSP and ASP. In case 3 and 4, to discourage the bias towards predicting the major class, the desired target distribution should be closer to the minor class depending on whether OSP or ASP is the minor class.

We proposed a new consistency loss function called Adaptive Blended Consistency Loss (ABCL) which captures the desirable properties listed above. Figure 5.5 illustrates how it works. ABCL uses the following loss function to generate a new target class distribution which is a blend of OSP and ASP.

$$ABCL(z, \hat{z}) = L_{con}(z_{blended}, z) + L_{con}(z_{blended}, \hat{z}) \tag{5.6}$$

$$z = p_\theta(y|u); \hat{z} = p_\theta(y|\hat{u}) \tag{5.7}$$

L_{con} is the Kullback-Leibler divergence between the blended target probability distribution $z_{blended}$ and either OSP (z) or ASP ($\hat{z}$). Hence ABCL pulls both the original and augmented predictions towards the target distribution. For a model with parameters θ, $p_\theta(y|u)$ and $p_\theta(y|\hat{u})$ denote the predicted class probability distribution for an original unlabelled example u and its augmented version $\hat{u}$ respectively. The gradient of the loss is not back-propagated through $z_{blended}$ during parameter optimisation. This aims to enforce the augmented embedding closer to the $z_{blended}$ target embedding.

The blended target class distribution $z_{blended}$ is defined as follows:

$$z_{blended} = (1 - k) * z + k * \hat{z} \tag{5.8}$$

Where k is the weighting value [0,1] that determines the proportion to which the blended target moves towards OSP versus ASP. When k equals 0, the target will

become OSP meaning ABCL will become the same as CL. On the other hand, as k approaches 1, the target approaches ASP. k is calculated based on predicted class frequencies with respect to the training dataset using the following formula:

$$k = max(0, min(\gamma * (N_{original} - N_{augmented}) + 0.5, 1)) \tag{5.9}$$

$N_{original}$ and $N_{augmented}$ are the class frequencies of the predicted class (class with the highest predicted probability) for the OSP and ASP respectively. $\gamma \in (0, 1]$ is the class imbalance compensation strength that controls how strong the new blended target class distribution skews towards either OSP or ASP. The value of k can be interpreted as follows:

- When the OSP is the minor class and the ASP is the major class (case 4 in Table 5.1 and Figure 5.4), the value of $N_{original}$ will be smaller than the value of $N_{augmented}$ so the value of k will be smaller than 0.5. This indicates that the new blended target class distribution will skew towards the minor class side (OSP).

- When the OSP is the major class and the ASP is the minor class (case 3 in Table 5.1 and Figure 5.4), the value of $N_{original}$ will be larger than the value of $N_{augmented}$ so the value of k will be larger than 0.5. This indicates that the new blended target distribution will skew towards the minor class side (ASP).

- When OSP and ASP both are the major class or both are the minor class (cases 1 and 2 in Table 5.1 and Figure 5.4), the value of $N_{original}$ and $N_{augmented}$ are similar, so the value of k will be around 0.5. In this case OSP and ASP contribute equally to the blended target distribution.

Class imbalance compensation strength γ. In ABCL (Equation 5.9), the γ term is used to control how strongly the new blended target distribution is pushed towards either OSP or ASP. As γ approaches 0, the impact of $N_{original} - N_{augmented}$ on the value of k will be smaller, and k will therefore be close to 0.5. This indicates that the new blended target class distribution will skew only slightly to either OSP or ASP. On the other hand, as γ approaches 1, the impact that $N_{original} - N_{augmented}$ has on k will be bigger. This indicates that the new blended target class distribution will skew strongly to either the OSP or ASP. When γ is set to a high value, ABCL approaches the standard CL in the case that $N_{augmented} >> N_{original}$, since $z_{blended} \approx z$. This corresponds to CL, where OSP is wholly responsible for defining the target of the unsupervised loss term.

ABCL for Mean Teacher The above ABCL is initially designed for the consistency loss of UDA. Here, we present a different version of ABCL for the consistency loss of Mean Teacher with only one consistency loss:

$$ABCL(z, \hat{z}) = L_{con}(z_{blended}, \hat{z}) \tag{5.10}$$

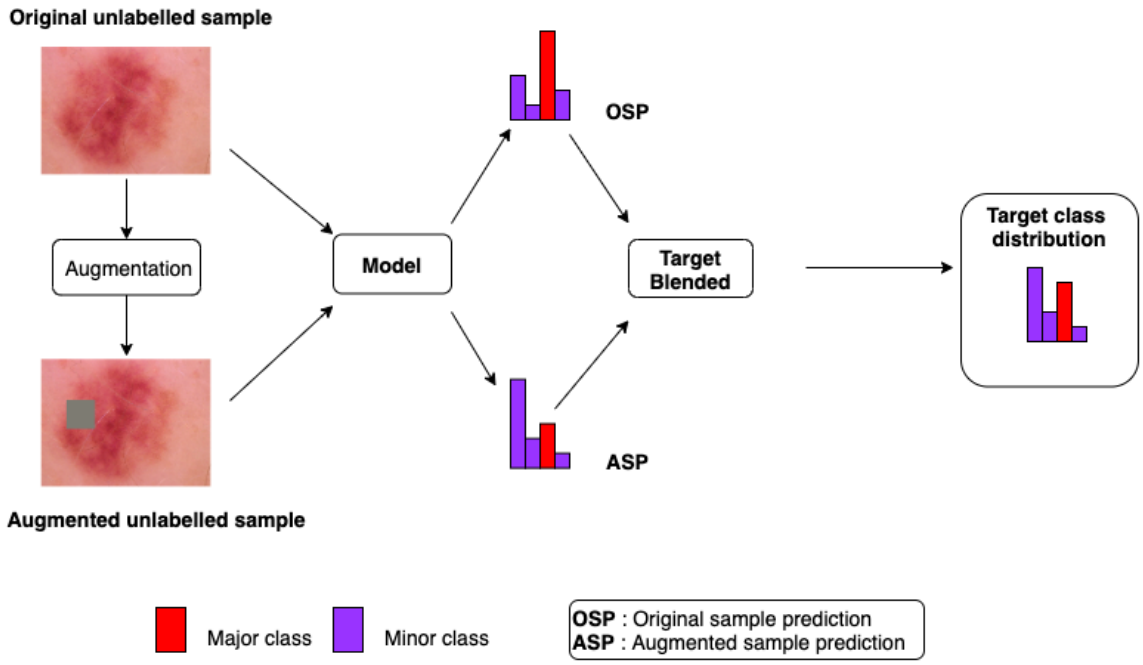


Figure 5.5: Diagram showing how ABCL works. The target distribution is blended more towards the minor class, although it still retains some of OSP's distribution.

$$z = p_{\theta_T}(y|x); \hat{z} = p_{\theta_S}(y|\hat{x}) \tag{5.11}$$

Where z and $\hat{z}$ is the probability distribution output of the teacher and student model respectively. There is only one consistency loss function, which is used to move $\hat{z}$ towards the blended target class distribution $z_{blended}$, compared to two consistency loss functions within ABCL for UDA. This is because the teacher model is not learnable hence the consistency loss can not be applied on its output.

5.3.1 Semi-supervised learning for segmentation

The UDA (described in Section 5.1.1) and Mean Teacher (described in Section 5.1.2) frameworks were initially designed for the semi-supervised classification problem, however we can easily adapt them to the semi-supervised semantic segmentation problem which is actually just classification applied to the pixel instead of whole image grain.

Geometric consistency for applying noise (affine transformations). In UDA, the consistency loss is applied on image-level class predictions of original and augmented images. However, we need to be more careful when applying consistency loss at the pixel level for semantic segmentation. In particular we need to correctly map each pixel in the original image to the corresponding pixel in the augmented image. For example, if the data augmentation rotates an image we need to perform the inverse transformation to the model output before we can apply consistency loss. This is because affine transformation will change the location of pixels from its original space. Therefore, to make the class prediction of original and augmented images at

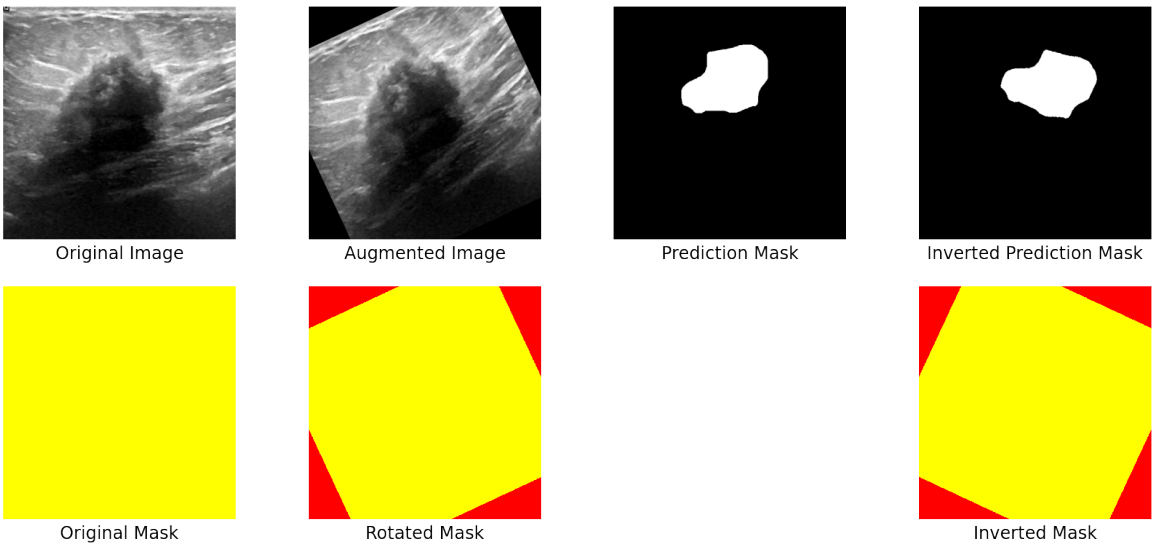


Figure 5.6: An example of our "inverse transformation" approach to make consistency loss geometrically consistent for the segmentation problem. In this example the data augmentation operation is a 25 degree anti-clockwise rotation. The bottom row shows the valid pixel mask in yellow and the invalid pixels in red. The image and mask are both rotated by 25 degrees. Then the prediction mask and the rotated mask are rotated by -25 degrees.

the pixel level geometrically consistent, we apply a masked inverse transformation before computing the loss. This process is shown in Figure 5.6 and is described as follows:

- Initialize a valid pixel mask that has the same size as the original image with all its pixel values set to 1. This is used later to mask out invalid pixels (pixels whose new locations are outside the original image size bounds after applied affine transformation) when computing the loss. Apply the same affine transformation that was applied to the original image (for data augmentation) to the valid pixel mask. Set regions of the resultant mask that do not map to the original image to invalid using a value of zero.
- Apply inverse affine transformation on the predicted segmentation map of augmented image. (inverted prediction segmentation map)
- Apply inverse affine transformation on the transformed valid pixel mask.

The inverted prediction segmentation map of the augmented image and the class prediction segmentation map of the original image are used to compute the consistency loss. Then the portions of the loss corresponding to the invalid regions are canceled out using the mask.

5.4 CNN architecture used

5.4.1 For classification problem

We opted to use ResNet-34 as the backbone of our classification network, initialising the model with weights that were pre-trained on the ImageNet [36] dataset. ResNet-34 is the 34-layer variant of the popular ResNet CNN architecture [97], which achieves excellent classification performance by alleviating a common issue with deep neural networks known as the "vanishing gradient problem". The vanishing gradient problem is commonly observed when training deep models with back-propagation and results in impeded training as gradients shrink to zero after passing through a large number of layers. ResNet includes skip connections which permit back-propagated gradients to flow more easily between distant layers, thus mitigating the vanishing gradient problem.

5.4.2 For segmentation problem

We used the DeepLabv2 with ResNet101 backbone for our semantic segmentation solution which was described in detail in Section 2.4.1. We chose this backbone since it is the same backbone used in the semi-supervised semantic segmentation method in [95].

Chapter 6

Experiment Setup

6.1 Dataset

We evaluate our proposed methodology on several medical imaging datasets selected to cover a variety of diseases.

For classification:

- HAM10000 [22] (main dataset): contains 10015 RGB dermatoscopic skin lesions images of size (450, 600) multiclass, divided into 7 classes: Pigmented Bowen's (AKIEC), Basal Cell Carcinoma (BCC), Pigmented Benign Keratoses (BKL), Dermatofibroma (DF), Melanoma (MEL), Nevus (NV), Vascula (VASC).
- REFUGE Challenge [23] (supporting dataset): contains 1200 RGB retinal fundus glaucoma images of various sizes. Each image is assigned a binary label of glaucoma or non-glaucoma.

Figure 6.1 shows example images for each class. As is shown in Table 6.1, the class distributions of both datasets are highly imbalanced.

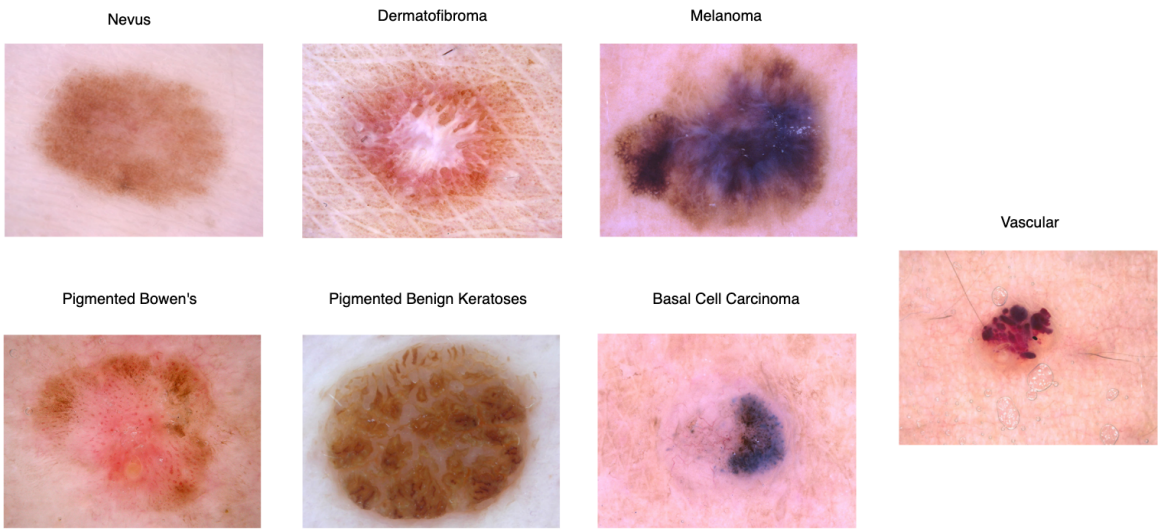
For segmentation:

- Nerve Ultrasound segmentation dataset [24]: contains 5635 nerve ultrasound

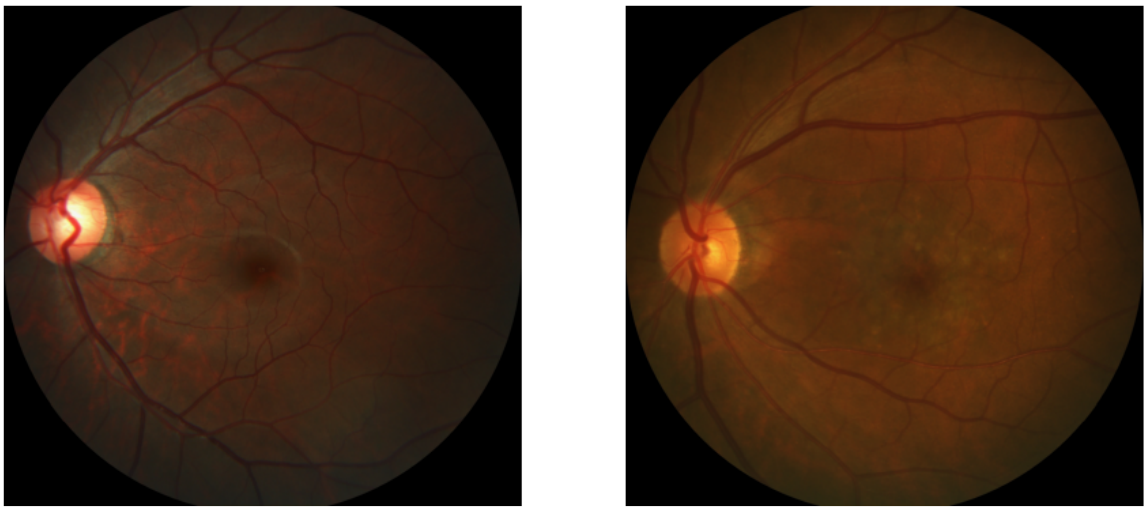
	MEL	**NV**	**BCC**	**AKIEC**	**BKL**	**DF**	**VASC**	**Total**
Skin cancer dataset	1113	6705	514	327	1099	115	142	10015
	0.11	0.67	0.06	0.03	0.11	0.01	0.01	1

	Glaucoma	**Non-Glaucoma**	**Total**
Retinal fundus glaucoma dataset	120	1080	1200
	0.1	0.9	1

Table 6.1: The class distribution of 2 experimental classification dataset. Both datasets are very imbalanced. The number in brackets indicates the fraction of samples for the given class. The most and least class frequency is (0.67,0.01) for skin cancer dataset and (0.9,0.1) for retinal fundus glaucoma dataset.

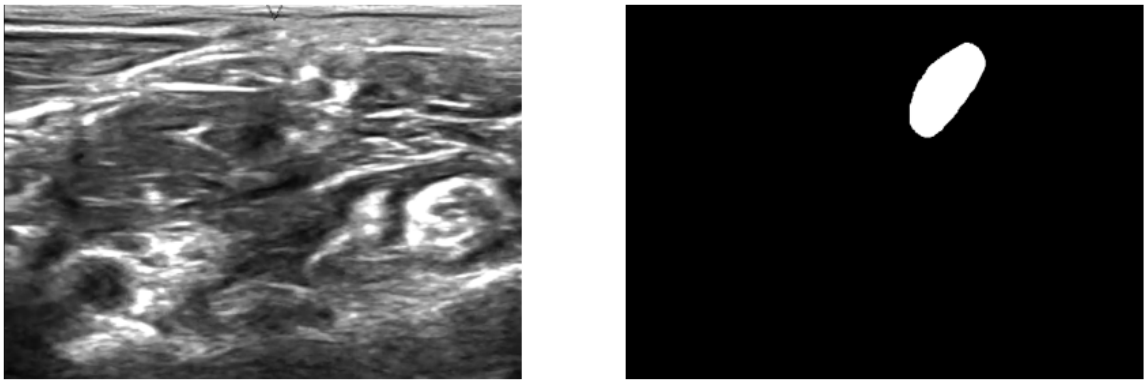


(a)

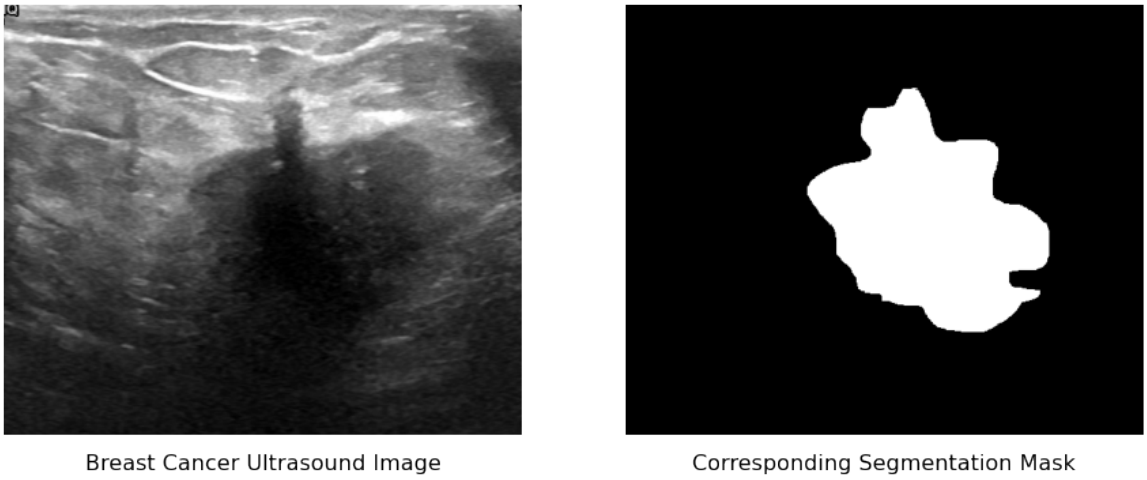


(b)

Figure 6.1: Example images of the dermatoscopic skin lesions and retinal fundus glaucoma dataset for each class.



(a)



(b)

Figure 6.2: Example images of the Nerve Ultrasound and Breast Cancer Ultrasound dataset, with their corresponding segmentation mask.

images and its corresponding segmentation masks of size (580,420) binary class, which is whether an image has nerve or not.

- Breast Cancer Ultrasound segmentation dataset [25]: contains 780 breast ultrasound images and its corresponding segmentation masks of average size (500,500) binary class, which is whether an image has breast tumor or not.

Figure 6.2 shows example images. Table 6.2 shows the class distribution of the dataset in pixels which are highly imbalanced.

We split the datasets as follows: 70% training images, 20% test images and 10% validation images. The validation and test dataset are used separately for the early

Dataset	**Background pixel**	**Foreground pixel**
Nerve Ultrasound segmentation	0.98	0.02
Breast Cancer Ultrasound segmentation	0.9	0.1

Table 6.2: The class distribution of 2 experimental segmentation dataset. Both datasets are very imbalanced in total pixels of background versus foreground.

stopping to select the best model, otherwise they will be combined to evaluate the model at final training epoch. For the classification task, 2/7 of the training dataset is labelled images and 5/7 is unlabelled images. But for the segmentation task, we will experiment with different amounts of labelled data in the training dataset. Notably, the unlabelled training dataset is larger than the labelled training dataset because it simulates the semi-supervised problem we stated earlier. That is, in medical image data, it is often the case there is a small amount of labelled images with a much larger set of unlabelled images. Additionally, we will ensure the class distribution across each of the data splits are the same.

6.2 Data processing

6.2.1 For classification

For the HAM10000 dataset, the model is trained with the original image size of (450, 600), in contrast, images of the REFUGE challenge dataset are resized to (512,512). All images of both datasets have their color normalised. In particular, the red and blue color channel will be normalised based on the green color channel. The formula is described as follows:

$$R = R * (\frac{G_{mean}}{R_{mean}}); B = B * (\frac{G_{mean}}{B_{mean}})$$

6.2.2 For segmentation

For the nerve segmentation dataset, the model is trained with the original image size of (580,420) and images of the breast cancer segmentation dataset are resized to (500,500)

6.3 Evaluation metrics

In this study, all metrics used to evaluate the model's performance are derived from the confusion matrix shown in Table 6.3.

6.3.1 For classification

Conventionally, a model's performance is usually measured by the accuracy metric indicating how many test samples are predicted correctly. In terms of confusion matrix, it can be calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

		Predicted Class	
		Positive Class	**Negative Class**
Actual Class	**Positive Class**	True Positive	False Negative
	Negative Class	False Positive	True Negative

Table 6.3: An example of a confusion matrix of the binary classification problem. True Positive (TP) means the number of positive class samples are correctly predicted as the positive class. False Positive (FP) means the number of negative class samples are mispredicted as the positive class. False Negative (FN) means the number of positive class samples are mispredicted as the negative class. True Negative (TN) means the number of negative class samples are correctly predicted as the negative class.

Unweighted Average Recall. Due to the class imbalance in the testing dataset, the model usually performs really well on the major classes and much worse on the minor classes. Since overall accuracy is dominated by the major class, considering this metric alone may conceal such poor performance on minor classes. However, minor class accuracy is really important, especially for medical image analysis since often the samples with disease belong to the minor class. Hence, the Skin Lesion Analysis towards Melanoma Detection (ISIC 2018) [113] competition used an evaluation metric that gives equal weight to all classes (called unweighted average recall (UAR)) to rank the performance of the different models on the HAM10000 dataset (our experimental dataset). Therefore, we also use UAR to measure our model's performance across all classes in a fair way. UAR is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{UAR} = \frac{\sum_{C=1}^{C} Recall_C}{C}$$

where TP is the number of true positives, FN is the number of false negatives, and C is the number of classes. The recall indicates the percentage of samples correctly classified (TP) over the total number of samples ($TP+FN$). By using this more class balanced evaluation metric, we actually evaluate the model based on the contribution of all classes equally rather than favoring the major classes. Since we are studying class imbalance specifically, we additionally make use of more granular metrics to further investigate the performance of our models, including the area under the receiving operating charac- teristic curve and geometric mean. Additionally, these two metrics are proposed because they were mostly used in studies of classification with class imbalance which were reported in the review [114].

Geometric mean. The geometric mean (G-mean) is the n-th root of the product of n-class recall. This is one of the most popular metrics for measuring the performance of classification models on class imbalanced data. G-Mean score is maximized when

the recall of all classes is balanced. The formula can be described as follows:

$$\text{G-mean} = \sqrt[C]{Recall_1 * Recall_2 * ...Recall_C}$$

Where Recall is defined as above, C is the number of classes.

The Receiving Operating Characteristic (ROC) curve. The ROC curve is a graph plotting the performance of a classification model at different classification thresholds as a curve. Furthermore, this graph is plotted from 2 parameters:

- True Positive Rate (TPR), is a synonym for recall. TPR = $\frac{TP}{TP+FN}$
- False Positive Rate (FPR). FPR = $\frac{FP}{FP+TN}$

Intuitively, the ROC curve indicates how well the model classifies when classification threshold is varied.

The Area Under the ROC curve (AUC). AUC is the entire area underneath the ROC curve which is between 0 and 1 and the ideal value is 1. AUC summarises a model's performance across all classification thresholds. Furthermore AUC can be interpreted as the probability that the model ranks a random positive example higher than a random negative example [115].

6.3.2 For segmentation

Dice Coefficient. Dice Coefficient (Dice) is a very common metric to measure a segmentation model's performance. The Dice Coefficient score is a value between 0 and 1 which represents the similarity between the prediction segmentation mask and the ground truth mask. Specifically, Dice Coefficient is described as the overlap of the object in prediction segmentation mask and the object in the ground truth mask, divided by the total size of the two objects, as calculated as follows:

$$Dice = \frac{2 * TP}{2 * TP + FP + FN}$$

Notably, we use the Dice score of the foreground class to represent the overall performance of experiments.

Sensitivity. The sensitivity, known as recall, indicates the percentage of the number of foreground class pixels correctly classified (TP) over the total number of foreground class pixels ($TP + FN$), as calculated as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

6.4 Model training and model selection for classification

All models were trained for 200 epochs with the static learning rate of 10^{-4}. Furthermore, the Stochastic Gradient Descent (SGD) optimizer with momentum 0.9 and weight decay 5 x 10^{-4} was used. The model was trained using batches of 30 examples per iteration, 8 of which were labelled and 22 of which were unlabelled.

We apply the early stopping technique to select the best model's state. The model will be evaluated with the validation dataset in every epoch. Then, the best model's state with the highest main metrics will be picked as the model evaluated on the test set. Applying early stopping rather than always selecting the best model's state at the final epoch can prevent the chosen model's state from overfitting. Eventually, the best model's state will be evaluated with the test dataset to obtain proposed measurement metrics.

6.5 Model training and model selection for segmentation

All models are trained for 20 epochs on the nerve segmentation dataset and 15 epochs on the breast cancer segmentation dataset with the static learning rate of 3 x 10^{-5}. Furthermore, the Adam optimizer was used. For the nerve segmentation dataset, a batch size of 20 training samples was used, 10 of the samples are labelled and 10 were unlabelled. For the breast cancer segmentation dataset, a batch size of 4 training samples was used, 2 of the samples are labelled and 2 were unlabelled. For experiments on both datasets, the model at final epoch is selected for the report.

6.6 Hardware and software

We considered Pytorch which is an open source machine learning library backed by Facebook and TensorFlow which is an open source machine learning framework backed by Google for our experimental software. The main difference between the two is that TensorFlow uses static computation graphs versus Pytorch, which uses dynamic computation graphs. That means in TensorFlow, we need to define the whole computation graph before running the model. Whereas, Pytorch allows us to flexibly modify the computation graph at runtime, hence it is easier to debug. Therefore, we decided to use Pytorch to do our experiments.

Moreover, training a deep network is very time-consuming because it requires doing a vast amount of mathematical operations. Therefore it is important to perform the processing in parallel to reduce the training time. Indeed, Graphical Processing Units

Task	Dataset	Data augmentation method
Classification	HAM10000 and REFUGE	Random horizontal flips, rotations(between 0 and 180 degree), erasing[26] (a random proportion (0.02-0.33) of input image will be erased) and color (jitter brightness, saturation and contrast with a random value in range of 0.9 to 1.1).
Segmentation	Nerve Ultrasound segmentation	Crop (321,321), horizontal flip, rotation (between 0 and 90)
	Breast Cancer Ultrasound segmentation	Crop (490,490), horizontal flip, rotation (between 0 and 90)

Table 6.4: Data augmentation methods for both tasks.

(GPUs) and CPUs with multiple cores can handle the parallel processing. Training deep learning tasks on GPUs run much faster than CPUs due to the much higher number of cores in GPUs compared to CPUs. Hence, we decided to run our experiment on a NVIDIA GPU GeForce RTX 2080 TI.

6.7 Data Augmentation

Strong data augmentation via RandAugmentation [63] is a key component of the UDA method, and the authors show that it can significantly improve the sample efficiency by encouraging consistency on a diverse set of augmented examples. However, experiments presented in Section 7.1.3 show ABCL performs poorly when strong data augmentation is used. We attribute the poor performance of ABCL in the presence of strong data augmentation to increased distance between the augmented and unaugmented examples in embedding space. ABCL assumes in cases 3 and 4 of Table 5.1 (one of the predictions belongs to a major class and the other belongs to a minor class), it is more likely the actual class is the minor class. However, when using strong data augmentation, the augmented major class sample may be moved far away from the original example in feature space. As a consequence, there is a risk the augmented sample may be predicted to be a different class. That is, samples belonging to the major classes are more likely to be predicted wrongly since the strong augmentation can make very significant changes to the appearance of the images. Consequently, the main assumption behind ABCL is violated when strong data augmentation is used. Hence, we use weak data augmentation for both labelled and unlabelled samples in both tasks. Details of our data augmentation methods are listed in Table 6.4.

6.8 Algorithms used in experimental study

6.8.1 Classification task

UDA baseline

The original UDA method is used as the baseline method using the standard cross entropy loss as the supervised loss and the standard consistency loss (Kullback-Leibler Divergence) as the unsupervised loss. This baseline is not designed to handle class imbalance. All other methods implemented modify this baseline UDA method. As default we use weak data augmentation for all algorithms implemented in this paper including UDA baseline. However in Section 7.1.3 we show an experiment where we compare the performance of strong versus weak data augmentation for UDA and UDA+ABCL. The results show that ABCL with weak data augmentation is able to outperform UDA using strong data augmentation. This shows that benefits of using ABCL outweigh the performance loss from using weak data augmentation.

Sampling based method for the labelled dataset

As we mentioned in Section 2.6, undersampling the major class may lose information from the major class samples and oversampling the minor class too much might cause the model to overfitting. Therefore, to balance this trade-off, we can combine oversampling the minor class and undersampling the major class to make a class balanced dataset. We will resample the skewed labelled dataset by using the intelligent methods Synthetic Minority Oversampling Technique (SMOTE) [68] and random undersampling. SMOTE is a statistical method that oversample the minor class in order to create a balanced dataset. Instead of just duplicating the existing sample, SMOTE generates a new synthetic sample based on the feature of the target class and its neighbours. On the other hand, the random undersampling method randomly removes instances of the major class. In our labelled HAM10000 dataset, the number of images of the most and least frequency class is 1320 and 22, respectively. Hence, we balance the labelled dataset to 500 images for each class.

Weighted Cross Entropy Loss (WeightedCE)

This is an extension of the Cross Entropy loss function which weights the loss of each class based on a given set of weights. The equation for this function loss is as follows:

$$WeightedCE(weight, p_t) = -weight[c] * log(p_t)$$

Where p_t is the predicted probability of a sample, $weight$ is a given set of class weight (0,1), $weight[c]$ is the given weight of the predicted class c (class with highest predicted probability). We define the set of class weight as follows:

- Class frequency: [0.11,0.67,0.06,0.03,0.11,0.01,0.01] for the HAM10000 dataset and [0.1,0.9] for the REFUGE dataset
- weight = 1 - Class frequency

The objective of WeightedCE is to give higher loss for the minor class and lower loss for the major class. Intuitively, it helps the model reduce the effect of the major class.

Focal Loss [72]

This loss function is used for supervised learning to reduce the effect of the easy samples. The idea is the major class are easy examples and hence should be given lower weight in the loss function. The equation for focal loss is as follows:

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

where $-(1-p_t)^{\gamma}$ is a modulating factor, p_t is the predicted probability of a sample, γ is a focusing hyper parameter [0,5]. When $\gamma = 0$, the loss function is identical to regular cross entropy loss, therefore the higher value of γ the increased effect of the modulating factor. Intuitively, if the sample is misclassified with small p_t, the loss of the sample is applied as normal. Whereas, if the sample is well-classified with high confidence of p_t then it is more likely to belong to the major class, therefore the loss of the sample is reduced. The default γ value used was 1 because we found this value for γ gave the best UAR result when we did a hyperparameter search.

Suppressed Consistency Loss (SCL) [82]

This is the state-of-the-art semi-supervised learning method used to address class skew that suppresses the consistency loss when the minor class is predicted. We implemented this approach on top of UDA. The loss is formulated as follows:

$$L_{SCL}(X_i) = g(N_c) * L_{con}(X_i),$$

$$\text{where } c = argmax(f_\theta(X_i))$$

$g(N_c)$ is any function that is inversely proportional to N_c. The authors proposed it as:

$$g(N_c) = \beta^{1-\frac{N_c}{N_{max}}}$$

where $\beta \in (0,1]$, N_c is the number of unlabelled examples of class c and N_{max} is the number of unlabelled examples of the major class. The objective of $g(N_c)$ is described as follows: if the model predicts the sample as a major class, the consistency loss of this sample is applied as normal however if the model predicts the sample as a minor class, the consistency loss of this sample is suppressed. The default value used was 0.8 because we found this value for gave the best UAR result when we did a hyperparameter search.

Adaptive Blended Consistency Loss (ABCL)

This is our method for addressing the class imbalance problem for semi-supervised learning. Our new method has only one hyperparameter that is class imbalance compensation γ. The default γ value used was 0.4 because we found this value gave the best overall result when considering UAR and the individual recall results for each class.

UDA-WeightedCE-SCL and UDA-WeightedCE-ABCL

It is necessary to experiment with combinations of supervised and unsupervised class imbalance methods to explore whether our ABCL method is beneficial alongside established supervised methods. Therefore, we will experiment Weighted Cross Entropy with ABCL and SCL.

6.8.2 Semantic segmentation task

UDA and Mean Teacher baseline

The original UDA and Mean Teacher method were both used as the baseline method using the standard cross entropy loss as the supervised loss and the standard consistency loss (Kullback-Leibler Divergence) as the unsupervised loss. All other methods implemented modify the baseline UDA and Mean Teacher methods. The α smoothing coefficient hyperparameter of Mean Teacher was set to 0.99 for all experiments, since we found this gave the best results. All solutions for the segmentation task used weak data augmentation.

Weighted Cross Entropy Loss (WeightedCE)

The WeightedCE was described in Section 6.8.1. For the segmentation task, we applied the loss on each pixel's predicted class distribution. After hyperparameter

searching, we found the class weight [1,1.5] gives the best performance for both segmentation datasets.

Dice Coefficient Loss (Dice) [85]

Inspired from Dice Coefficient metric, this is a region-based loss which maximizes the similarity between two images, The equation for the loss is as follows:

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1}$$

Where y is the one-hot encoded ground truth for each pixel, $\hat{p}$ is the prediction probability for each pixel. 1 is added to avoid the function becoming undefined.

Adaptive Blended Consistency Loss (ABCL)

The default γ value used was 0.4 because we found this value gave the best overall Dice score for the foreground class.

WeightedCE-ABCL and Dice-ABCL

We experimented with combinations of supervised and unsupervised class imbalance methods to explore whether our ABCL method is complementary to supervised methods. Therefore, we combined ABCL with WeightedCE and Dice loss.

Chapter 7

Experimental Results

7.1 Classification results

7.1.1 Results for the skin cancer (HAM10000) dataset

Table 7.1 shows the test set results (UAR as the main metric and G-mean, average AUC as supporting metrics) comparing our ABCL with other methods for the HAM10000 dataset. All semi-supervised methods are based on the UDA method. Our ABCL achieved the highest UAR, G-mean and average AUC values of 0.67, 0.62 and 0.95 respectively among all experimented methods. Existing class imbalance methods designed for labelled data (Sampling and Focal) performed worse than the baseline UDA method for the UAR and G-mean metric meaning they were not effective at addressing the class imbalance problem in SSL. This is because these methods can only make a small contribution to overall model quality since they can only be applied to the labelled data which in SSL is just a small portion of the total dataset. On the other hand, class imbalance methods designed for unlabelled data (SCL and our ABCL) outperformed the baseline UDA for the UAR and G-mean metrics meaning these methods are more effective than methods which are only applicable to labelled data. This is because these methods modify the consistency loss which is used for the unlabelled data (occupies a larger proportion of all data for SSL).

When WeightedCE was used with ABCL and SCL, the UAR performance was boosted from 0.67 to 0.68 and from 0.61 to 0.65 respectively. This can be explained by the fact addressing class imbalance for the labelled data helps the model produce lower bias of the major class for the pseudo target distribution in the unsupervised consistency loss. Additionally, although the SCL outperformed the baseline UDA, its performance was still worse than our ABCL in all experiments that involved both methods. This is because as discussed in Section 5.2, SCL has the two problems of bias towards major class and ignoring the augmented sample prediction when determining

Algorithms	UAR	G-mean	Average AUC
Supervised learning	0.75	0.71	0.98
UDA baseline	0.59	0.56	0.91
UDA-Sampling	0.51	0.44	0.89
UDA-WeightedCE	0.59	0.55	0.92
UDA-Focal	0.55	0.50	0.92
UDA-SCL	0.61	0.58	0.92
UDA-WeightedCE-SCL	0.65	0.64	0.94
UDA-ABCL (ours)	0.67	0.62	0.95
UDA-WeightedCE-ABCL (ours)	**0.68**	**0.66**	**0.96**

Table 7.1: Test set results of supervised learning, UDA baseline and various methods designed for handling class imbalance on top of UDA for the HAM10000 dataset. The results in bold show the best result for each column among the SSL methods.

Algorithms	MEL (0.11)	NV (0.67)	BCC (0.06)	AKIEC (0.03)	BKL (0.11)	DF (0.01)	VASC (0.01)	UAR
UDA baseline	0.43	0.93	0.81	0.35	0.55	0.50	0.56	0.59
UDA-Sampling	0.44	0.84	0.69	0.52	0.53	0.36	0.12	0.51
UDA-Weighted Loss	0.39	0.94	**0.83**	0.40	0.53	0.41	0.59	0.59
UDA-Focal	0.34	**0.96**	0.81	0.32	0.51	0.32	0.62	0.55
UDA-SCL	0.40	0.94	0.78	0.46	0.57	0.45	0.65	0.61
UDA-ABCL (ours)	**0.73**	0.88	**0.83**	0.46	0.64	0.36	**0.76**	0.67
UDA-WeightedCE-SCL	0.5	**0.96**	0.73	**0.58**	0.59	**0.55**	0.65	0.65
UDA-WeightedCE-ABCL (ours)	0.68	0.9	0.79	0.51	**0.67**	0.45	0.74	**0.68**

Table 7.2: Test set recall results of the algorithms for each class of the HAM10000 dataset. The number in brackets under each class name shows the fraction of all samples belonging to that class. Therefore the major class with most examples is NV.

the target class distribution.

To further understand the performance of ABCL against the rival methods, we also reported the test set recall of each class in Table 7.2. This allows us to see how the major and minor classes contribute to the UAR. ABCL performed the best among all methods for almost all of the minor classes (MEL, BCC, BKL and VASC) and performed worse than most methods for the major class NV. This result shows ABCL is doing what it was designed to do, namely, it alleviates the bias towards the major class unlike CL and SCL.

The importance of high recall for minor classes. In the medical domain it is usually better to mistakenly report a false positive than to miss a true positive disease diagnosis. This is because manual assessment or further testing can be used to correct the misdiagnosis. However, missing a positive disease diagnosis may leave a potentially fatal condition untreated. Additionally, in medical data, the amount

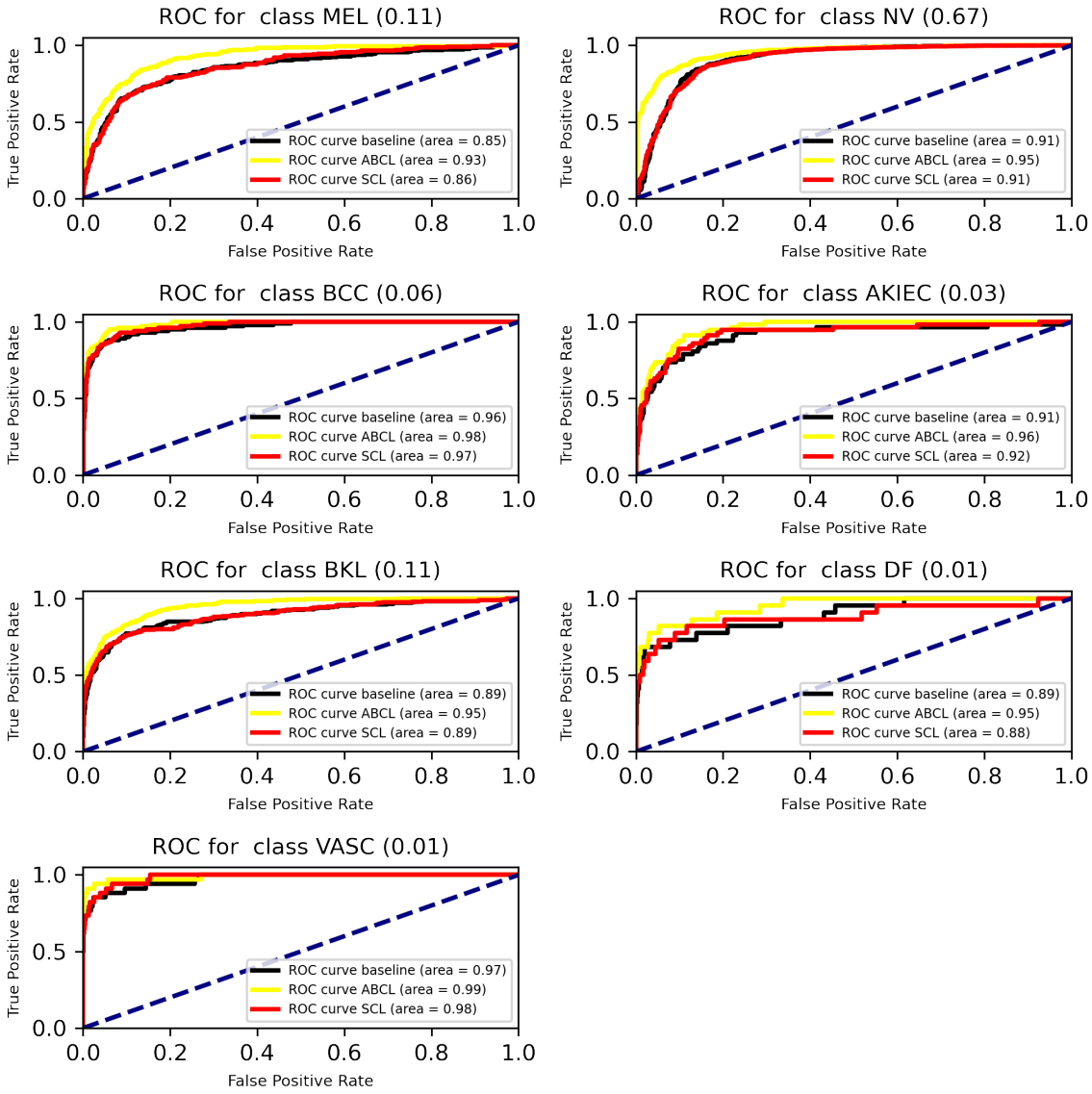


Figure 7.1: Test set ROC and AUC results for each class of the HAM10000 dataset using UDA baseline, SCL and ABCL methods. The number in the brackets next to the class name is the fraction of examples that belong to that class.

of healthy cases is usually higher than the amount of diseased cases implying the major class is usually the healthy case. In the skin cancer dataset, the NV class is the benign class, which is also the only major class and the rest are minor classes. Based on this principle, ABCL gives better performance than all its competitors. The test set results in Table 7.2 shows that ABCL when compared to its nearest non-ABCL rival achieves higher recall for the minor classes MEL, BCC, BKL and VASC. This means ABCL is less likely to miss disease diagnosis than alternative losses.

Figure 7.1 shows our ABCL method compared against the SCL and UDA baseline using the ROC and the corresponding AUC. ABCL's AUC results are better than the UDA baseline and SCL for all classes according to Table 7.3. This means ABCL is better at separating between the positive and negative classes than the alternatives.

Algorithms	**MEL (0.11)**	**NV (0.67)**	**BCC (0.06)**	**AKIEC (0.03)**	**BKL (0.11)**	**DF (0.01)**	**VASC (0.01)**	**Average AUC**
UDA baseline	0.85	0.91	0.96	0.91	0.89	0.89	0.97	0.91
UDA-ABCL (ours)	**0.93**	**0.95**	**0.98**	**0.96**	**0.95**	**0.95**	**0.99**	**0.95**
UDA-SCL	0.86	0.91	0.97	0.92	0.89	0.88	0.98	0.92

Table 7.3: Test set AUC results for each class of the HAM10000 dataset and its average using UDA baseline, SCL and ABCL methods. The number in the brackets next to the class name is the fraction of examples that belong to that class.

γ **value**	**MEL (0.11)**	**NV (0.67)**	**BCC (0.06)**	**AKIEC (0.03)**	**BKL (0.11)**	**DF (0.01)**	**VASC (0.01)**	**UAR**
0.2	0.66	**0.92**	0.82	0.47	0.65	0.32	0.68	0.65
0.4	0.73	0.88	0.83	0.46	0.64	0.36	0.76	0.67
0.5	0.74	0.86	0.84	0.51	0.64	0.41	0.71	0.67
0.6	0.73	0.83	0.84	0.51	**0.66**	0.36	0.76	0.67
0.8	0.75	0.78	**0.86**	0.49	0.63	**0.45**	0.76	0.68
1	**0.78**	0.76	0.83	**0.53**	0.64	0.41	**0.82**	**0.68**

Table 7.4: Test set recall results for each class of the HAM10000 dataset when the γ hyper parameter value of ABCL is varied.

7.1.2 The effects of varying γ value

In this section we analyse the important γ parameter of ABCL according to reported recall results for each class in Table 7.4. Important observations include the following:

- As γ approaches 1, the recall of the major class NV decreases and the recall of almost all minor classes increases.
- As γ approaches 0, the recall of the major class NV increases and the recall of almost all minor classes decreases.

This implies that as γ is increasing, the model is compensating more towards the minor classes than the major class, leading to the degradation of major class performance. This can be explained by the case when the model correctly predicts the OSP as the major class and mispredicts the ASP as the minor class, the blended target class distribution is then skewed towards the ASP. As γ approaches 1, the blended target class distribution moves closer to ASP. Therefore the γ term in ABCL can be used to tradeoff decreased major class performance for increased minor class performance.

From Table 7.4 we decided to choose a γ value of 0.4 as our default value since it provided a good balance of recall for the minor classes while retaining most of the recall for the major class NV.

Algorithms	γ value	MEL (0.11)	NV (0.67)	BCC (0.06)	AKIEC (0.03)	BKL (0.11)	DF (0.01)	VASC (0.01)	UAR
ABCL (always-on blending)	0.2	0.66	**0.92**	0.82	0.47	0.65	0.32	0.68	0.65
	0.4	0.73	0.88	0.83	0.46	0.64	0.36	0.76	0.67
	0.5	0.74	0.86	0.84	0.51	0.64	0.41	0.71	0.67
	0.6	0.73	0.83	0.84	0.51	**0.66**	0.36	0.76	0.67
	0.8	0.75	0.78	**0.86**	0.49	0.63	0.45	0.76	0.68
	1	**0.78**	0.76	0.83	0.53	0.64	0.41	**0.82**	**0.68**
ABCL (selective blending)	0.2	0.52	0.91	0.77	0.49	0.60	0.45	0.65	0.63
	0.4	0.65	0.83	0.83	0.49	0.62	**0.55**	0.74	0.67
	0.5	0.64	0.80	0.85	0.50	0.61	0.53	0.72	0.66
	0.6	0.65	0.69	0.85	0.46	0.55	0.45	0.76	0.63
	0.8	0.69	0.53	**0.86**	0.53	0.58	0.32	0.76	0.61
	1	0.59	0.50	0.82	**0.56**	0.59	0.41	0.71	0.60

Table 7.5: Test set recall results selective target blending versus always-on blending for each class of the HAM10000 dataset.

7.1.3 Ablation study

Selective versus always-on target blending

Here we compare ABCL with selective target blending versus ABCL with always-on target blending using the HAM10000 dataset. In our experiments we use ABCL with always-on target blending as our default method. This means even when the original and augmented samples both are predicted as the minor or major class (cases 1 and 2 of Table 5.1), we still blend the two samples to produce the targets. However in selective target blending we do not blend the original and augmented predictions when they both predict the minor or major class, instead in this situation we resort to the standard UDA method of just setting the target as the predicted output from the original sample.

The results in Table 7.5 show that the version of ABCL that always blends targets gives higher UAR (between 0.65 and 0.68) across the entire range of γ values. In contrast, ABCL with selective target blending performs poorly for high γ values, especially for the major class NV. ABCL improves the model's performance on minor classes by compensating more towards minor classes. As a consequence, in case 2 of Table 5.1 (the original and augmented samples both are predicted as the minor class), there is a harmful effect to major classes that the actual major class might be mispredicted as the minor class. Therefore, in this case, always selecting the original distribution as the target distribution might exacerbate this harmful effect. On the other hand, ABCL with always-on blending can mitigate this harmful effect, leading to less degradation in the recall for the major class.

Weak versus strong data augmentation

In this section we compare the effects of different data augmentation strategies for the baseline UDA method and our ABCL method on the HAM10000 dataset.

Algorithms	UAR	G-mean	Average AUC
UDA baseline + WeakAug	0.59	0.56	0.91
UDA baseline + StrongAug	0.61	0.56	0.92
ABCL + WeakAug	**0.67**	**0.62**	**0.95**
ABCL + StrongAug	0.50	0.42	0.91

Table 7.6: Test set UAR, G-mean and average AUC results of the HAM10000 dataset comparing UDA baseline and ABCL with strong data augmentation and weak data augmentation.

Algorithms	UAR	G-mean	Average AUC
UDA baseline	0.55	0.37	0.82
UDA-WeightedCE	0.57	0.43	**0.83**
UDA-Focal	0.57	0.43	0.82
UDA-SCL	0.55	0.37	0.81
UDA-WeightedCE-SCL	0.55	0.37	0.82
UDA-ABCL (ours)	0.55	0.37	0.82
UDA-WeightedCE-ABCL (ours)	**0.67**	**0.61**	**0.83**

Table 7.7: Test set UAR, G-mean and AUC results for the REFUGE dataset.

Table 7.6 shows the results of this experiment. The results show that for the UAR, G-mean and average AUC metrics, strong data augmentation works better than weak data augmentation for the UDA baseline whereas the opposite is true for ABCL. As discussed in Section 6.7, we believe the reason for this is when using strong data augmentation, even samples belonging to major classes may be predicted wrongly since the augmentation can make very significant changes to the appearance of the images. Consequently, the main assumption behind ABCL is violated when strong data augmentation is used. The results also show that ABCL with weak data augmentation is able to outperform the baseline UDA using the strong data augmentation. This is an important result since it shows ABCL is able to overcome any negative consequence of not being able to use strong data augmentation.

7.1.4 Results for the retinal fundus glaucoma (REFUGE) dataset

Table 7.7 shows the test set results of comparing ABCL with the other competing methods for the glaucoma (REFUGE) dataset. ABCL does not improve the UDA-baseline model's performance when the standard cross entropy loss is used for the supervised loss. However, using a combination of WeightedCE for the supervised loss and ABCL for unsupervised loss, UDA-WeightedCE-ABCL is able to significantly outperform UDA-WeightedCE and UDA-WeightedCE-SCL. As explained in Section 7.1.1, WeightedCE can help the model produce lower bias of towards the major class for the pseudo target distribution in the unsupervised consistency loss which

is of benefit to both ABCL and SCL methods. However, UDA-WeightedCE-ABCL outperforms UDA-WeightedCE-SCL because ABCL is able to adaptively blend the target distribution between the original and augmented samples's predicted distributions depending on which predicted the minor class. This helps ABCL better address the class imbalance problem in the unlabelled data.

7.2 Segmentation results

7.2.1 Results for the Nerve Ultrasound segmentation dataset

Table 7.8 shows the test set dice score result for experiments on top of Mean Teacher and UDA SSL methods with three different amounts of labelled data. For UDA experiments ABCL methods consistently outperformed non-ABCL methods. These results are consistent with our classification results. An interesting observation for the UDA experiments is that ABCL methods outperform non-ABCL methods by a larger margin when there is a smaller amount of labelled data. This may be attributed to the fact that directly addressing class imbalance in the consistency loss becomes more important with a higher ratio of unlabelled compared to labelled data.

The Mean Teacher results show a less clear picture. For 500 and 980 labelled data sizes ABCL methods perform either the same or a little better than its nearest competitor. For 200 labelled data a non-ABCL method, MT-WeightedCE is the best performer. The reason that ABCL does not perform as well when Mean Teacher SSL is used compared to UDA can be explained by the violation of ABCL's assumption. In the scenario where OSP and ASP do not agree (case 3 and 4 of Table 5.1), ABCL assumes it is more likely that the true class is the minor class. However, in the Mean Teacher method, OSP and ASP are generated from two different models, Teacher and Student respectively. For some examples, these two models might have different latent representations for the original sample and augmented sample. This introduces an additional source of mispredictions on true major class examples which ABCL is unable to account for the behavioural difference between models. In such cases ABCL will assume that the discrepancy is a result of data bias and attempt to correct it by encouraging prediction of the minor class, which may be incorrect and lead to increased false positives.

We also observe that replacing standard cross-entropy with Dice loss typically has a positive or neutral effect on ABCL's performance, with the exception of UDA on 980 labelled examples. This is consistent with our earlier discussion in Section 7.1.1 and illustrates that methods dealing with class imbalance for labelled examples are complementary to ABCL.

Our ABCL method is designed to boost the recall of minor classes which is very important in the medical domain. However, a higher dice score (our main metric)

Algorithms	200	500	980
MT baseline	0.54	0.59	0.61
MT-WeightedCE	**0.56**	0.59	0.61
MT-Dice	0.54	**0.60**	0.61
MT-ABCL (ours)	0.55	0.59	0.61
MT-WeightedCE-ABCL (ours)	0.53	0.59	**0.62**
MT-Dice-ABCL (ours)	0.55	**0.60**	**0.62**
UDA baseline	0.44	0.53	0.58
UDA-WeightedCE	0.51	0.54	0.57
UDA-Dice	0.50	0.54	0.58
UDA-ABCL (ours)	0.55	**0.58**	**0.61**
UDA-WeightedCE-ABCL (ours)	0.55	**0.58**	0.60
UDA-Dice-ABCL (ours)	**0.57**	**0.58**	0.59

Table 7.8: Test set dice score result for experiments on top of Mean Teacher and UDA methods with various amounts of labelled data for the Nerve Ultrasound segmentation dataset. The numbers in bold indicate the best results for each column separated into Mean Teacher and UDA methods.

does not guarantee a higher recall score. Hence in Table 7.9 we report the recall scores for the various methods built on top of the UDA and Mean Teacher SSL. These recall results show the combination of WeightedCE and ABCL achieved the highest recall among all competing methods. This can be attributed to the fact that ABCL is designed to reduce the bias towards the major class hence the higher recall on the minor class. Also as mentioned in the results of Section 7.1.1 ABCL is able to successfully work in a complementary fashion with the WeightedCE supervised loss function.

7.2.2 Results for the Breast Cancer Ultrasound segmentation dataset

Table 7.10 shows the test set dice score results of experiments on top of the Mean Teacher and UDA on the Breast Cancer Ultrasound segmentation dataset. For all three different amounts of labelled data, experiments with the ABCL method always outperform experiments without the ABCL method for both Mean Teacher and UDA SSL methods. The advantage of using ABCL is particularly significant for the UDA SSL method. These results are consistent with the results for the Nerve Ultrasound segmentation dataset presented in Section 7.2.1. Namely ABCL is clearly better than competing methods for the UDA SSL but the advantage is less significant for the Mean Teacher dataset. On 50 and 150 labelled examples, the combination of ABCL with Dice or WeightedCE achieved the highest dice score among all experiments on top of the UDA method. These results once again indicate ABCL and methods designed to work on unlabelled data can be complementary to effectively tackle the class imbalance problem in semi-supervised learning.

Algorithms	**200**	**500**	**980**
MT baseline	0.62	0.63	0.62
MT-WeightedCE	0.66	0.65	0.65
MT-Dice	0.49	0.59	0.60
MT-ABCL (ours)	0.69	0.66	0.63
MT-WeightedCE-ABCL (ours)	**0.71**	**0.67**	**0.66**
MT-Dice-ABCL (ours)	0.66	0.66	0.64
UDA baseline	0.33	0.45	0.51
UDA-WeightedCE	0.43	0.48	0.52
UDA-Dice	0.43	0.47	0.52
UDA-ABCL (ours)	**0.70**	0.57	**0.62**
UDA-WeightedCE-ABCL (ours)	**0.70**	**0.66**	**0.62**
UDA-Dice-ABCL (ours)	0.62	0.52	0.52

Table 7.9: Test set recall results for experiments on top of UDA and Mean Teacher with various amounts of labelled data for the Nerve Ultrasound segmentation dataset.

Algorithms	**50**	**150**	**300**
MT baseline	0.68	0.75	0.78
MT-WeightedCE	0.69	0.75	0.78
MT-Dice	0.57	0.73	0.78
MT-ABCL (ours)	**0.71**	**0.76**	**0.79**
MT-WeightedCE-ABCL (ours)	0.70	**0.76**	0.78
MT-Dice-ABCL (ours)	0.70	0.75	**0.79**
UDA baseline	0.53	0.63	0.66
UDA-WeightedCE	0.58	0.66	0.68
UDA-Dice	0.51	0.65	0.67
UDA-ABCL (ours)	0.59	0.70	**0.72**
UDA-WeightedCE-ABCL (ours)	0.64	**0.71**	**0.72**
UDA-Dice-ABCL (ours)	**0.65**	0.70	0.71

Table 7.10: Test set dice score results for experiments on top of the Mean Teacher and UDA with various amounts of labelled data on the Breast Cancer Ultrasound segmentation dataset. The numbers in bold indicate the best results for each column separated into Mean Teacher and UDA methods.

Datasets	γ **value**					
		200		**500**		**980**	
Nerve Ultrasound segmentation dataset	0.4	**0.55**	0.69	**0.59**	0.66	**0.61**	0.63
	0.9	0.46	**0.74**	0.57	**0.70**	0.61	**0.66**
		50		**150**		**300**	
Breast Cancer Ultrasound segmentation dataset	0.4	**0.71**	0.72	**0.76**	0.74	**0.79**	0.75
	0.9	0.68	**0.78**	0.75	**0.76**	0.78	**0.76**

Table 7.11: Test set dice score (on the left) and corresponding recall (on the right) results for our ABCL method on top of the Mean Teacher SSL when γ is varied across different amounts of labelled data for both segmentation datasets.

7.2.3 The effects of varied γ value of our ABCL method to the segmentation task

In this section, we will analyse the effects of varying the γ parameter on the performance of our ABCL method in terms of both dice score and recall. Table 7.11 shows the test set dice score result along with the recall of ABCL with γ set to 0.4 and 0.9. This result is reported for both segmentation datasets with three different amounts of labelled data. We can make the following observations from the results in Table 7.11:

- The dice score at $\gamma = 0.9$ is lower than at $\gamma = 0.4$
- The recall at $\gamma = 0.4$ is lower than at $\gamma = 0.9$

As discussed in Section 7.1.2, the γ parameter in ABCL can be used to tradeoff decreased recall of the major class for increased recall of the minor class for the classification task. The lower dice score when γ is at the higher value of 0.9 can be explained as follows. The dice score penalises for false positives. At the higher γ value of 0.9 the benefits of the high recall of the minor class is outweighed by the even higher penalty of false positives for the minor class. Hence we choose a γ value of 0.4 as our default setting for the segmentation tasks since it provides a better balance between recall of the minor class and false positive rate.

Chapter 8

Conclusion

In this study, we identified an important gap in the literature. Namely the need to address the class imbalance problem within the context of semi-supervised classification and segmentation for medical images. This is an important problem to study since medical image datasets often have skewed distributions and missing positive disease diagnosis can have fatal consequences. We address this gap by proposing a new consistency loss function called Adaptive Blended Consistency Loss (ABCL). To demonstrate the effectiveness of ABCL, we applied it to the perturbation based SSL algorithm UDA and Mean Teacher. ABCL overcame the problem of the standard consistency loss by generating a new blended target class distribution from the mix of original and augmented sample's class distribution in accordance to class frequency. Extensive experiments showed ABCL consistently outperforms baseline SSL implementations such as UDA and Mean Teacher and methods designed to address the class imbalance problem. For the skin cancer classification task, our proposed ABCL method was able to improve the performance of the UDA baseline from 0.59 to 0.67 UAR, outperform methods that address the class imbalance problem for labelled data (between 0.51 and 0.59 UAR) and the SCL method for addressing skew in unlabelled data (0.61 UAR). On the imbalanced retinal fundus glaucoma dataset, combining with Weighted Cross Entropy loss, ABCL achieved 0.67 UAR as compared to 0.57 to its nearest rival. For the two segmentation tasks, our ABCL method clearly outperformed rival methods for the dice score when the UDA SSL was used. When the Mean Teacher SSL was used ABCL performed the best or near the best depending on the amount of labelled data used.

Overall the results show the effectiveness of ABCL to alleviate the class imbalance problem for both semi-supervised classification and semi-supervised segmentation for medical images.

8.1 Future work

As future work we would like to explore the following ideas:

- Apply our ABCL method on top of other SSL methods that use the consistency loss and in other domains apart from medical imaging.
- Test the effectiveness of our ABCL method when used with visual transformer network backbone.
- Report the experimental results for segmentation algorithms using the Hausdorff (95%) distance and specificity metrics.
- Compare the effect of our ABCL method on the convergence of model training to the baselines.
- Apply our ABCL method with patch-based methods.
- Apply our ABCL method to 3D medical datasets.
- The value k in equation 5.9 is calculated based on a small number of labelled data, which might not well represent the real distribution of the entire dataset in some cases. Explore whether the proposed loss is still stable under the situations when the class frequency in the labelled training dataset is not close to the entire dataset.

Bibliography

[1] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. "Unsupervised data augmentation". In: *arXiv preprint arXiv:1904.12848* (2019).

[2] Amirreza Rezvantalab, Habib Safigholi, and Somayeh Karimijeshni. "Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms". In: *arXiv preprint arXiv:1810.10348* (2018).

[3] Seung Seog Han, Myoung Shin Kim, Woohyung Lim, Gyeong Hun Park, Ilwoo Park, and Sung Eun Chang. "Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm". In: *Journal of Investigative Dermatology* 138.7 (2018), pp. 1529–1538.

[4] Lei Bi, Jinman Kim, Euijoon Ahn, and Dagan Feng. "Automatic skin lesion analysis using large-scale dermoscopy images and deep residual networks". In: *arXiv preprint arXiv:1703.04197* (2017).

[5] Achim Hekler, Jochen S Utikal, Alexander H Enk, Axel Hauschild, Michael Weichenthal, Roman C Maron, Carola Berking, Sebastian Haferkamp, Joachim Klode, Dirk Schadendorf, et al. "Superior skin cancer classification by the combination of human and artificial intelligence". In: *European Journal of Cancer* 120 (2019), pp. 114–121.

[6] Osamu Iizuka, Fahdi Kanavati, Kei Kato, Michael Rambeau, Koji Arihiro, and Masayuki Tsuneki. "Deep Learning Models for Histopathological Classification of Gastric and colonic epithelial tumours". In: *Scientific Reports* 10.1 (2020), pp. 1–11.

[7] Worawate Ausawalaithong, Arjaree Thirach, Sanparith Marukatat, and Theerawit Wilaiprasitporn. "Automatic lung cancer prediction from chest X-ray images using the deep learning approach". In: *2018 11th Biomedical Engineering International Conference (BMEiCON)*. IEEE. 2018, pp. 1–5.

[8] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning". In: *arXiv preprint arXiv:1711.05225* (2017).

[9] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. "Medical transformer: Gated axial-attention for medical image segmentation". In: *arXiv preprint arXiv:2102.10662* (2021).

[10] Jeya Maria Jose Valanarasu, Vishwanath A Sindagi, Ilker Hacihaliloglu, and Vishal M Patel. "Kiu-net: Overcomplete convolutional architectures for biomedical image and volumetric segmentation". In: *arXiv preprint arXiv:2010.01663* (2020).

[11] Nikhil Kumar Tomar, Debesh Jha, Michael A Riegler, Håvard D Johansen, Dag Johansen, Jens Rittscher, Pål Halvorsen, and Sharib Ali. "Fanet: A feedback attention network for improved biomedical image segmentation". In: *arXiv preprint arXiv:2103.17235* (2021).

[12] Dinh Viet Sang, Tran Quang Chung, Phan Ngoc Lan, Dao Viet Hang, Dao Van Long, and Nguyen Thi Thuy. "AG-CUResNeSt: A Novel Method for Colon Polyp Segmentation". In: *arXiv preprint arXiv:2105.00402* (2021).

[13] Xin Yi, Ekta Walia, and Paul Babyn. "Unsupervised and semi-supervised learning with Categorical Generative Adversarial Networks assisted by Wasserstein distance for dermoscopy image Classification". In: *arXiv preprint arXiv:1804.03700* (2018).

[14] Antonia Creswell, Alison Pouplin, and Anil A Bharath. "Denoising adversarial autoencoders: classifying skin lesions using limited labelled training data". In: *IET Computer Vision* 12.8 (2018), pp. 1105–1111.

[15] Hai Su, Xiaoshuang Shi, Jinzheng Cai, and Lin Yang. "Local and Global Consistency Regularized Mean Teacher for Semi-supervised Nuclei Classification". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 559–567.

[16] Wenkai Yang, Juanjuan Zhao, Yan Qiang, Xiaotang Yang, Yunyun Dong, Qianqian Du, Guohua Shi, and Muhammad Bilal Zia. "DScGANS: Integrate Domain Knowledge in Training Dual-Path Semi-supervised Conditional Generative Adversarial Networks and S3VM for Ultrasonography Thyroid Nodules Classification". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 558–566.

[17] Shuailin Li, Chuyu Zhang, and Xuming He. "Shape-aware semi-supervised 3d semantic segmentation for medical images". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 552–561.

[18] Fengbei Liu, Yaqub Jonmohamadi, Gabriel Maicas, Ajay K Pandey, and Gustavo Carneiro. "Self-supervised Depth Estimation to Regularise Semantic Segmentation in Knee Arthroscopy". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 594–603.

[19] Shumeng Li, Ziyuan Zhao, Kaixin Xu, Zeng Zeng, and Cuntai Guan. "Hierarchical Consistency Regularized Mean Teacher for Semi-supervised 3D Left Atrium Segmentation". In: *arXiv preprint arXiv:2105.10369* (2021).

[20] Anneke Meyer, Suhita Ghosh, Daniel Schindele, Martin Schostak, Sebastian Stober, Christian Hansen, and Marko Rak. "Uncertainty-aware temporal self-learning (UATS): Semi-supervised learning for segmentation of prostate zones and beyond". In: *Artificial Intelligence in Medicine* 116 (2021), p. 102073.

[21] Yanwen Li, Luyang Luo, Huangjing Lin, Hao Chen, and Pheng-Ann Heng. "Dual-Consistency Semi-Supervised Learning with Uncertainty Quantification for COVID-19 Lesion Segmentation from CT Images". In: *arXiv preprint arXiv:2104.0322* (2021).

[22] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5 (2018), p. 180161.

[23] José Ignacio Orlando, Huazhu Fu, João Barbosa Breda, Karel van Keer, Deepti R Bathula, Andrés Diaz-Pinto, Ruogu Fang, Pheng-Ann Heng, Jeyoung Kim, JoonHo Lee, et al. "Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs". In: *Medical image analysis* 59 (2020), p. 101570.

[24] Halyard Health. *Ultrasound Nerve Segmentation*. 2016. URL: `https://www.kaggle.com/c/ultrasound-nerve-segmentation/data`.

[25] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. "Dataset of breast ultrasound images". In: *Data in brief* 28 (2020), p. 104863.

[26] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in neural information processing systems*. 2017, pp. 1195–1204.

[27] Samuli Laine and Timo Aila. "Temporal ensembling for semi-supervised learning". In: *arXiv preprint arXiv:1610.02242* (2016).

[28] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993.

[29] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. "Mixmatch: A holistic approach to semi-supervised learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5050–5060.

[30] Seeger Matthias. "Learning with labeled and unlabeled data". In: *Inst. Adapt. Neural Comput* (2001).

[31] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[32] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. "Realistic evaluation of deep semi-supervised learning algorithms". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3235–3246.

[33] Jesper E Van Engelen and Holger H Hoos. "A survey on semi-supervised learning". In: *Machine Learning* 109.2 (2020), pp. 373–440.

[34] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[37] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[38] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. "Handwritten digit recognition with a back-propagation network". In: *Advances in neural information processing systems*. 1990, pp. 396–404.

[39] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[41] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision". In: *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE. 2010, pp. 253–256.

[42] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9.4 (2018), pp. 611–629.

[43] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[44] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. "Panoptic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413.

[45] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[47] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. "Deep neural networks segment neuronal membranes in electron microscopy images". In: *Advances in neural information processing systems* 25 (2012), pp. 2843–2851.

[48] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[49] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. "The pascal visual object classes challenge 2012 (voc2012) results (2012)". In: *URL http://www. pascal-network. org/challenges/VOC/voc2011/workshop/index. html*. 2011.

[50] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[51] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[52] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[53] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[54] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[55] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[56] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012).

[57] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. "Random erasing data augmentation". In: *arXiv preprint arXiv:1708.04896* (2017).

[58] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019), p. 60.

[59] Agnieszka Mikołajczyk and Michał Grochowski. "Data augmentation for improving deep learning in image classification problem". In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, pp. 117–122.

[60] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[61] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A neural algorithm of artistic style". In: *arXiv preprint arXiv:1508.06576* (2015).

[62] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 113–123.

[63] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *arXiv preprint arXiv:1909.13719* (2019).

[64] Justin M Johnson and Taghi M Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (2019), p. 27.

[65] Inderjeet Mani and I Zhang. "kNN approach to unbalanced data distributions: a case study involving information extraction". In: *Proceedings of workshop on learning from imbalanced datasets*. Vol. 126. 2003.

[66] Miroslav Kubat, Stan Matwin, et al. "Addressing the curse of imbalanced training sets: one-sided selection". In: *Icml*. Vol. 97. Citeseer. 1997, pp. 179–186.

[67] Ricardo Barandela, Rosa M Valdovinos, J Salvador Sánchez, and Francesc J Ferri. "The imbalanced training sample problem: Under or over sampling?" In: *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer. 2004, pp. 806–814.

[68] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[69] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning". In: *International conference on intelligent computing*. Springer. 2005, pp. 878–887.

[70] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2009, pp. 475–482.

[71] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. "Training deep neural networks on imbalanced data sets". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 4368–4374.

[72] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[73] Huy Phan, Martin Krawczyk-Becker, Timo Gerkmann, and Alfred Mertins. "DNN and CNN with weighted and multi-task loss functions for audio event detection". In: *arXiv preprint arXiv:1708.03211* (2017).

[74] Haishuai Wang, Zhicheng Cui, Yixin Chen, Michael Avidan, Arbi Ben Abdallah, and Alexander Kronzer. "Predicting hospital readmission via cost-sensitive deep learning". In: *IEEE/ACM transactions on computational biology and bioinformatics* 15.6 (2018), pp. 1968–1978.

[75] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. "Cost-sensitive learning of deep feature representations from imbalanced data". In: *IEEE transactions on neural networks and learning systems* 29.8 (2017), pp. 3573–3587.

[76] Chong Zhang, Kay Chen Tan, and Ruoxu Ren. "Training cost-sensitive deep belief networks on imbalance data problems". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 4362–4367.

[77] Bartosz Krawczyk and Michał Woźniak. "Cost-sensitive neural network with roc-based moving threshold for imbalanced classification". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2015, pp. 45–52.

[78] JJ Chen, C-A Tsai, H Moon, H Ahn, JJ Young, and C-H Chen. "Decision threshold adjustment in class prediction". In: *SAR and QSAR in Environmental Research* 17.3 (2006), pp. 337–352.

[79] Hualong Yu, Changyin Sun, Xibei Yang, Wankou Yang, Jifeng Shen, and Yunsong Qi. "ODOC-ELM: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data". In: *Knowledge-Based Systems* 92 (2016), pp. 55–70.

[80] Haibo He and Edwardo A Garcia. "Learning from imbalanced data". In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1263–1284.

[81] Qi Dong, Shaogang Gong, and Xiatian Zhu. "Imbalanced deep learning by minority class incremental rectification". In: *IEEE transactions on pattern analysis and machine intelligence* 41.6 (2018), pp. 1367–1381.

[82] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. "Class-Imbalanced Semi-Supervised Learning". In: *arXiv preprint arXiv:2002.06815* (2020).

[83] Saeid Asgari Taghanaki, Yefeng Zheng, S Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. "Combo loss: Handling input and output imbalance in multi-organ segmentation". In: *Computerized Medical Imaging and Graphics* 75 (2019), pp. 24–33.

[84] Fausto Milletari, Seyed-Ahmad Ahmadi, Christine Kroll, Annika Plate, Verena Rozanski, Juliana Maiostre, Johannes Levin, Olaf Dietrich, Birgit Ertl-Wagner, Kai Bötzel, et al. "Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound". In: *Computer Vision and Image Understanding* 164 (2017), pp. 92–102.

[85] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[86] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks". In: *International workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387.

[87] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[88] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[89] Jost Tobias Springenberg. "Unsupervised and semi-supervised learning with categorical generative adversarial networks". In: *arXiv preprint arXiv:1511.06390* (2015).

[90] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1742–1750.

[91] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7268–7277.

[92] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. "Adversarial learning for semi-supervised semantic segmentation". In: *arXiv preprint arXiv:1802.07934* (2018).

[93] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. "Semi-supervised semantic segmentation with high-and low-level consistency". In: *IEEE transactions on pattern analysis and machine intelligence* (2019).

[94] Yassine Ouali, Céline Hudelot, and Myriam Tami. "Semi-supervised semantic segmentation with cross-consistency training". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12674–12684.

[95] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. "Semi-supervised semantic segmentation needs strong, varied perturbations". In: *British Machine Vision Conference*. 31. 2020.

[96] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. "Medical image analysis using convolutional neural networks: a review". In: *Journal of medical systems* 42.11 (2018), p. 226.

[97] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[98] Teresa Mendonça, Pedro M Ferreira, Jorge S Marques, André RS Marcal, and Jorge Rozeira. "PH 2-A dermoscopic image database for research and benchmarking". In: *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE. 2013, pp. 5437–5440.

[99] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[100] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[101] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[102] David Gutman, Noel CF Codella, Emre Celebi, Brian Helba, Michael Marchetti, Nabin Mishra, and Allan Halpern. "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC)". In: *arXiv preprint arXiv:1605.01397* (2016).

[103] Jason W Wei, Laura J Tafe, Yevgeniy A Linnik, Louis J Vaickus, Naofumi Tomita, and Saeed Hassanpour. "Pathologist-level classification of histologic patterns on resected lung adenocarcinoma slides with deep neural networks". In: *Scientific reports* 9.1 (2019), pp. 1–8.

[104] Kun Fan, Shibo Wen, and Zhuofu Deng. "Deep Learning for Detecting Breast Cancer Metastases on WSI". In: *Innovation in Medicine and Healthcare Systems, and Multimedia*. Springer, 2019, pp. 137–145.

[105] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[106] Dong Nie, Yaozong Gao, Li Wang, and Dinggang Shen. "ASDNet: attention based semi-supervised deep networks for medical image segmentation". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2018, pp. 370–378.

[107] Yizhe Zhang, Lin Yang, Jianxu Chen, Maridel Fredericksen, David P Hughes, and Danny Z Chen. "Deep adversarial networks for biomedical image segmentation utilizing unannotated images". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 408–416.

[108] Han Zheng, Lanfen Lin, Hongjie Hu, Qiaowei Zhang, Qingqing Chen, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, Ruofeng Tong, and Jian Wu. "Semi-supervised segmentation of liver using adversarial learning with deep atlas prior". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 148–156.

[109] Gerda Bortsova, Florian Dubost, Laurens Hogeweg, Ioannis Katramados, and Marleen de Bruijne. "Semi-supervised medical image segmentation via learning consistency under transformations". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 810–818.

[110] Xiaomeng Li, Lequan Yu, Hao Chen, Chi-Wing Fu, and Pheng-Ann Heng. "Semi-supervised skin lesion segmentation via transformation consistent self-ensembling model". In: *arXiv preprint arXiv:1808.03887* (2018).

[111] Lequan Yu, Shujun Wang, Xiaomeng Li, Chi-Wing Fu, and Pheng-Ann Heng. "Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 605–613.

[112] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2097–2106.

[113] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)". In: *arXiv preprint arXiv:1902.03368* (2019).

[114] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. "A systematic review on imbalanced data challenges in machine learning: Applications and solutions". In: *ACM Computing Surveys (CSUR)* 52.4 (2019), pp. 1–36.

[115] James A Hanley and Barbara J McNeil. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1 (1982), pp. 29–36.