Image Understanding using Convolutional and Graph Neural Networks

Dissertation

submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

by

Amit Lohan
Roll No: 183079033

under the guidance of

Prof. Amit Sethi

Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai - 400076.

2021

Dissertation Approval Sheet

This is to certify that the dissertation titled

Image Understanding using Convolutional and Graph Neural Networks

By

Amit Lohan
(183079033)

is approved for the degree of **Master of Technology**.

Digital Signature
Amit Sethi (i17185)
07-Jul-21 10:40:34 AM

Prof. Amit Sethi
(Guide)

Digital Signature
Vikram M Gadre (i94038)
01-Jul-21 07:11:06 PM

Prof. Vikram Gadre
(Examiner 1)

Digital Signature
Sharat Chandran (051054)
07-Jul-21 08:09:09 PM

Prof. Sharat Chandran
(Examiner 2)

Digital Signature
Vikram M Gadre (i94038)
01-Jul-21 07:36:15 PM

Chairperson

Date : June 30, 2021

Abstract

The work presented in this thesis comprises of the work carried out on two sub-projects, both based on image segmentation, understanding and visual information processing. The work includes observations and analysis from experiments conducted to explore, understand, test and improve the capability and functionality of CNNs and Graph Neural Networks for solving the ploblems that are a part of this thesis project.

The first sub-project has the aim of developing a deep learning based method for automatic segmentation of different regions in whole-slide H&E histopathology images. H&E-stained histopathological whole-slide images contain rich pathological information. As these images have very high resolution and an enormous size, their manual analysis is quite cumbersome and time consuming. So, an automatic segmentation method can be quite useful for extracting the relevant information from histopathological images. The extracted information from the segmentation results may be quite useful in clinical analysis such as predicting risk from the morphological information of regions of Lymph Nodes. Segmentation of whole-slide images can be a challenging task. In this work, some of the challenges that are faced during the whole slide level segmentation of images and their solutions are presented. The contributions include a multi-resolution segmentation CNN architecture, a data augmentation technique and an algorithm to reconstruct the segmentation results effectively at the whole slide level. A number of experiments were conducted to segment the different regions of lymph nodes in whole slide images and the developments made and results obtained so, are presented in chapters 1 to 5.

In some of the computer vision problems where a relationship between objects or object parts present in the image is useful for the solution or can be exploited for it, CNNs alone may not be enough to capture a significant amount of available visual and inter-relationship information. In such problems, graph neural networks may perform better by modeling the dependency between various parts of images. The second half of this thesis deals with one such problem and aims to represent images with limited number of key-points by making use of graph neural networks and explores if graph neural networks assisted by CNNs can be used to represent images effectively. Finally, some potential research directions in this field are presented. Chapters 6-10 focus on this sub-project.

Acknowledgments

I express my sincere gratitude towards my guide **Prof. Amit Sethi** for his constant help, encouragement and inspiration throughout the project work. Without his invaluable guidance, this work would never have been a successful one.

I would like to thank the research scholars **Nikhil Cherian Kurian** and **Abhijeet Tukaram** for their advice and guidance for techniques and implementations used in this work and for their valuable suggestions and helpful discussions.

I would like to thank **Swati Meena** who is a 2018 batch masters student and worked with me on a number of initial experiments of 1st project of this thesis. I would also like to thank **Nimish Dharamshi** for helping me with experiments on Camelyon Dataset.

I would like to thank **Gregory Verghese** who is a PhD student at King's College London for helpful discussions on techniques, methods and dataset related support.

I would also like to thank **Dr. Anita Grigoriadis** of **King's College London** for her pathological and medical advices about the results and **Dr. Swapnil Rane** of **Tata Memorial Hospital** for his help in judging the obtained results and for providing detailed annotations for data preparation.

Last, but not the least, I would like to thank the whole MeDAL family which made my stay at IIT Bombay a memorable one.

Amit Lohan
IIT Bombay
November 26, 2020

Contents

List of Figures

Chapter 1

Introduction: Segmentation of Whole-Slide H&E Images

1.1 H&E Images of Lymph Nodes

Haematoxylin and Eosin stain (H&E stain) is one of the most important tissue stains used in histology and the whole slide images produced from this stain are often assessed for various kinds of medical diagnosis and pathological examinations. For example, when a pathologist assesses the biopsy of a possible cancer, the histological stain to be used is likely to be H&E. Figures 1.1 and 1.2 show zoomed out view of whole slide images of H&E stained lymph node tissues.

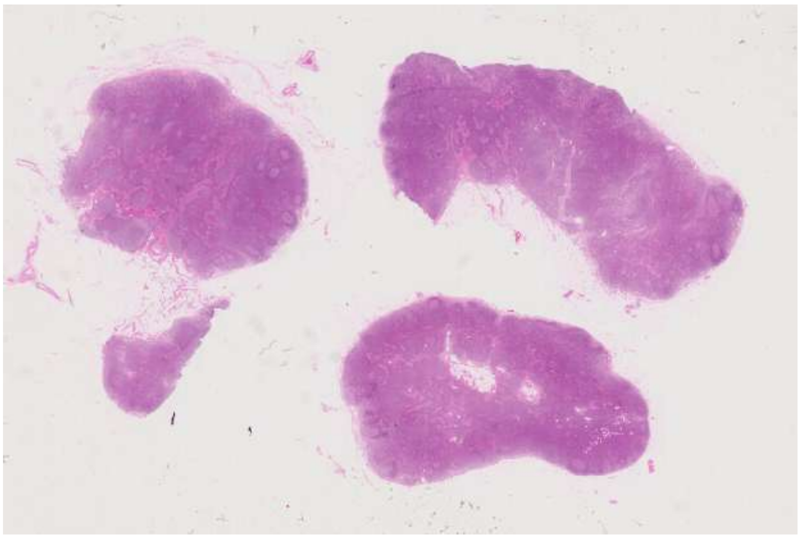


Figure 1.1: Whole Slide H&E Image 1 (Size:126976x84736 Pixels)

H&E staining consists of two histological stains: hematoxylin and eosin. The hematoxylin stains cell nuclei blue, and eosin stains the extracellular matrix and cytoplasm pink, and the other structures take on different shades, hues, and combinations of these colors. This way the staining makes it easy for a pathologist to easily differentiate between the nuclear and cytoplasmic parts of a cell. These whole slide images are often scanned at a 400x resolution and are quite large in size. In digital pathology, the 10x factor is often dropped for the sake of simpler communication. For example 400x is

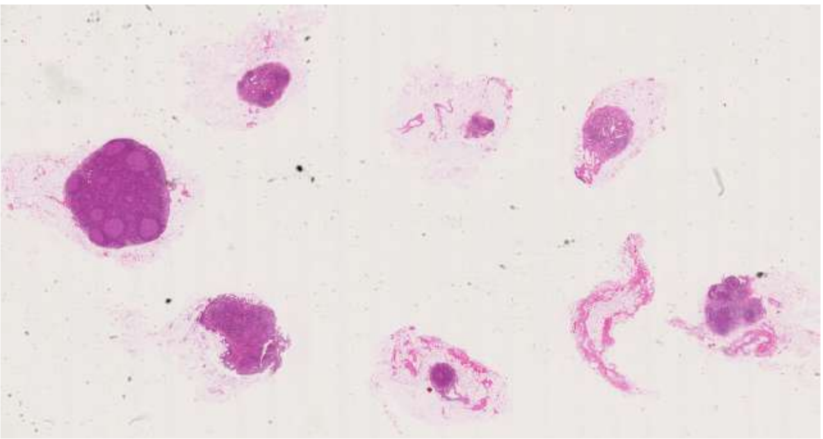


Figure 1.2: Whole Slide H&E Image 2 (Size:98304x51968 Pixels)

referred to as 40x. This terminology has been used in the rest of this thesis for referring to image resolution.

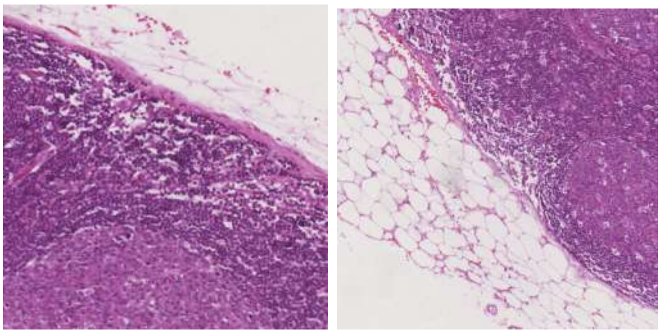


Figure 1.3: Patches extracted from zoomed in regions of H&E Image

Fig. 1.3 shows two zoomed in regions of the H&E whole slide image in Fig. 1.1. This kind of patches are often used as inputs to CNNs in deep learning for classification and segmentation problems as it is generally required to break the H&E WSI into small parts before deep learning algorithms can be used.

1.2 Regions in H&E images of Lymph Nodes

H&E whole slide images can be used to locate the constituent regions of lymph nodes namely, germinal centers, sinus, follicle, adipose(fat), tumour and interfollicular fegions. Figure 1.4 shows the appearance of these regions in a an H&E whole slide image section. These regions are quite variable in appearance and are often difficult to spot and even expert pathologists might sometimes disagree on the presence or absence of these regions in a location.

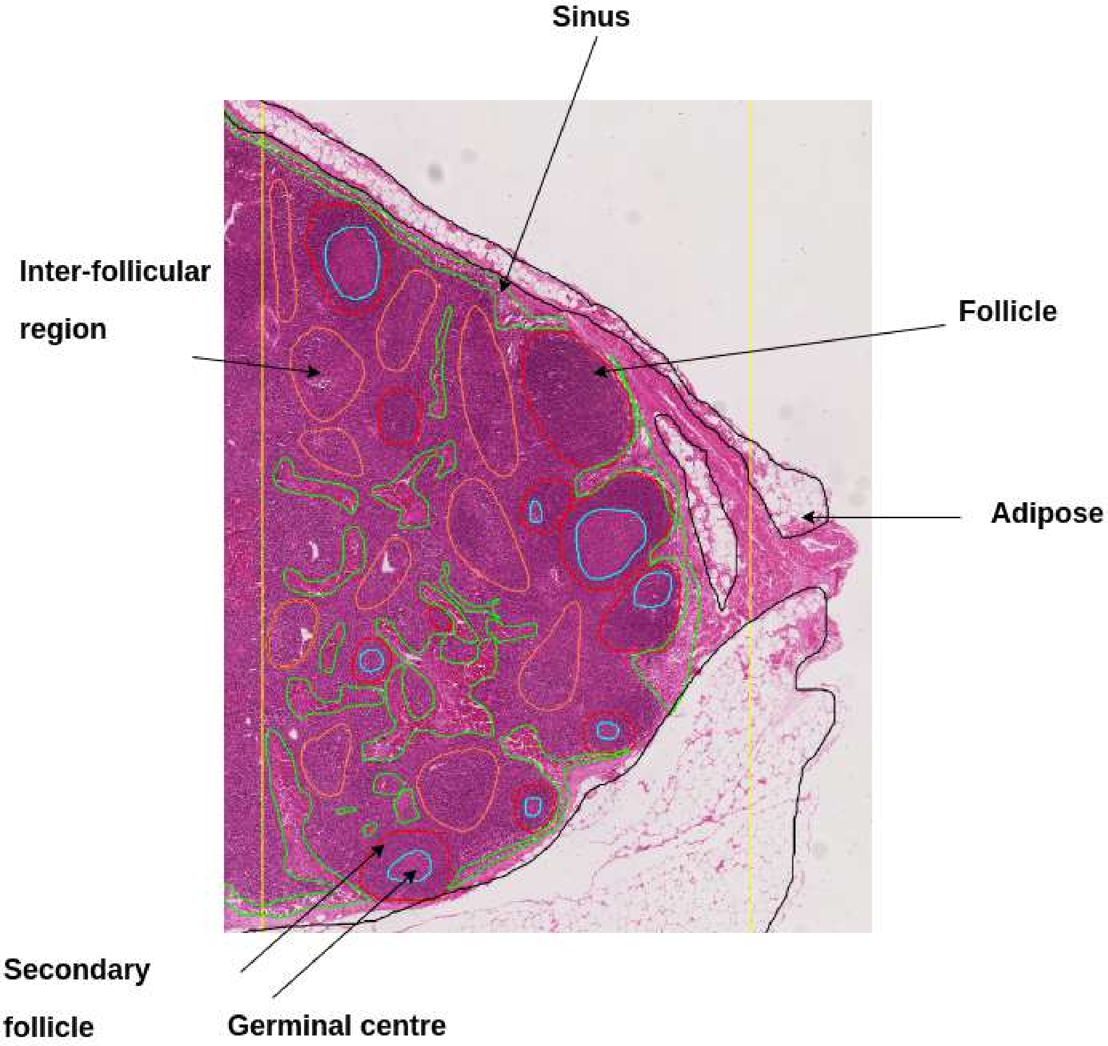


Figure 1.4: Different regions in lymph nodes

1.3 Challenges in Segmentation of Histopathology Whole-Slide Images

- **Large size:** Histopathology whole-slide images are enormous in size e.g. 200000x200000 pixels. It is not possible to feed the whole images to neural networks because of hardware constraints. So, appropriate techniques to segment the images virtually at a whole-slide level are to be employed.

- **Imprecise annotations:** It is quite difficult to mark the exact boundaries of regions and the annotations marked by pathologists are imprecise and not always on the correct boundaries. So, a mechanism to deal with these incorrect boundaries is required to be incorporated in training.

- **Large variation of color, shapes and sizes:** The regions to be segmented vary largely in size, shape and color etc. Some of the regions such as Germinal Centers in lymph nodes are small and relatively easier to capture whereas regions such as sinus may range to a few thousand pixels and have high variation of shape and color also, which makes the segmentation task difficult.

- **Resolution dependence:** The whole-slide images are digitised at a very high resolution and are stored in .ndpi format at multiple resolutions, all being the downsampled version of the same highest resolution image. After observing some of the results, it was observed that the segmentation results are dependent on the resolution of whole-slide image used for training. Given a fixed architecture of the segmentation neural network, some regions are best segmented a one resolution and other at a different resolution. This dependence on resolution makes it difficult to find the best resolution for segmentation.

1.4 Related Work

The performance of semantic segmentation pipelines significantly progressed with the advent of deep supervised learning models in general computer vision. The end to end automatic feature extraction capabilities of deep learning coupled with novel encoder-decoder architectures made it possible to completely outperform the previous baselines in semantic segmentation. The remarkable success however came at the expense of providing numerous high-quality supervised masks that serves to train these supervised training networks. In literature many innovations were reported to address these issues and a popular subset of deep learning architectures that reported the most improved results falls into the category of fully convolution networks (FCNs) originally proposed in [cite FCN]. Some of the popular modifications in FCN architecture includes SegNet, Deep CRFs and UNet.

The U-Net model [1], which uses an encoder-decoder architecture with skip connections across various downsampling and upsampling blocks enables precise determination of fine object boundaries and is reported to be the most widely accepted choice in medical image segmentation tasks. U-Net like architectures are used in various forms and are quite effective at identifying complex pixel patterns in images.

Khened et al. in [9] have developed a framework which uses UNet as well as two more convolutional networks in parallel as an ensemble for segmentation of patches. Their method effectively segments the patches but lacks mechanisms to handle the dependence of results on resolution from which patches are extracted. Also the method uses multiple CNNs parallelly which increases the number of trainable parameters, hence the large amount data requirements as well as more hardware requirements.

Some of the approaches [11],[12],[13] are based on patch-wise classification, where the model predicts a class label for each patch. This results in a coarser segmentation of the whole-slide level mask. These methods are not effective for finding the precise boundaries, which may be acceptable for large tumour segmentation, but may not be acceptable for segmentation of other tissue regions which may vary in size from very small to very large. Nevertheless, it has been used in tumor segmentation methods.

Most of the discussed methods focus on segmentation of tumour regions, which are typically large in size and show less dependence on the resolution of input patch. A general whole-slide segmentation method, however, should be able to segment all kinds of regions in whole slide images (e.g. Germinal Centers, Sinus and Adipose in lymph node H&E images). These regions could be very small, limited to clusters of a few hundred pixels as well as large enough to range across regions of tens of thousands of pixels. These region may also show dependence on the resolution of input patches. The overall framework for segmentation should be able to deal with these challenges and variations. Our method deals with effective segmentation of regions in lymph nodes, in addition to the segmentation of large tumour regions. The associated statistics such as number, shapes, sizes and locations of these regions (germinal centers, sinus, follicles etc.), relative area and amount of sinus in lymph nodes may be extracted from a corpus of lymph node images and may be investigated for a correlation with the presence or absence of signs of cancer in the host. The work presented in rest of this thesis focuses on experiments and observations based on methods that can be used to extract such features and investigate the correlation of these features with absence or presence of signs of cancer.

Chapter 2

Data & Dataset Preparation

2.1 Whole Slide Image Data

The dataset using which most of the experiments in this project were conducted is the dataset provided by Guy's Hospital, London through our KCL Collaborators. The Guy's Hospital Dataset consists of 77 whole slide images of Lymph nodes. Each slide is of variable size, is in ndpi image format and contains the following tissue regions, germinal centres, follicles, sinus histiocytosis, and adipose. The annotations for each slide are available in qpdata and XML files which are marked by pathologists. An example of an annotated whole slide image is shown in Fig 1.4 where various colour boundaries represent the marked regions of interest and tissue regions.

Some of the experiments during the last stage of the project have also been conducted on the Camelyon-16 dataset. This dataset has a total of 403 whole-slide images, out of which 191 were used for training, 82 for validation and 130 whole-slide images were used for testing. Only Tumour regions are annotated in Camelyon dataset.

2.1.1 Resolution Levels in Whole Slide Images

Each ndpi file contains the same whole slide image at 10 different resolutions, all downsampled versions of the highest resolution which is 40x (400x absolute magnification). So level 0 is the highest resolution, level 1 is once downsampled to half the resolution of level 0, i.e. if level 0 is 20000X20000 then level 1 is 10000X10000 and level 2 is 5000X5000 and so on. As a convention, level 4, which is most convenient to handle for viewing purpose is referred to as 2.5X resolution, level 3 as 5X and so on.

2.2 Dataset Preparation

For the dataset preparation for training deep learning models to segment H&E images, the first step is to find a region of interest in the whole slide image. As most of the area inside a whole slide image is empty white background around 20% of the area actually contains the tissues and concerned regions of lymph nodes. In some of the whole slide images, the pathologist marks a region of interest and provides detailed annotations inside this region only. But in other cases, a region of interest is decide with the help of a program by finding the leftmost, rightmost, topmost and bottommost points and then deciding the rectangular region. Now all the area inside this region is extracted and also a coloured mask (different colours for different classes) is generated with the help of a program and the annotation points provided by the pathologists.

Figure 2.1 shows the process of extraction of tissue regions within regions of interest from a whole slide image. The rectangular regions bounding the annotated area are saved as images and overlapping patches for the corresponding tissue region along with their segmentation masks are saved in the dataset directory with the same names but in different directories named patches and masks. The same name helps loading data at the time of training.

One-vs-all class Segmentation

For one vs all class segmentation, while creating the mask, only the annotations corresponding to that class are used to draw mask, with a white colour and rest all the annotations are ignored, merging them with black background as shown in Fig. 2.1.

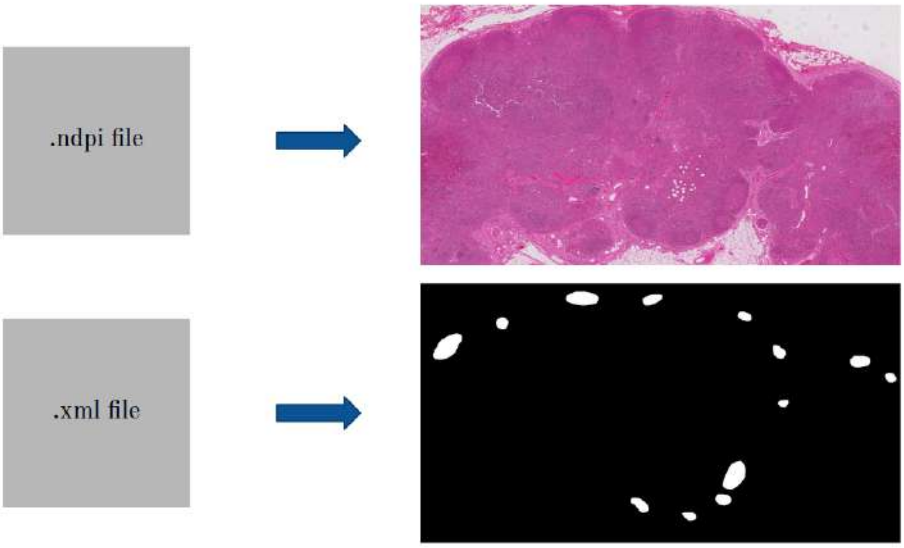


Figure 2.1: Data Preparation for one vs all classification.

After extracting the the whole region of interest and the corresponding class mask, the whole image is divided into small, overlapping patches and the corresponding patch masks are also saved. At the time of training, a patch from ROI and its corresponding mask patch are loaded by the dataloader to compute the loss.

Multi-class Segmentation

Similar to one-vs-all class segmentation, a colored mask, with a different colour for each class is generated using the annotation points in qpdata or xml file at the time of ROI extraction. After that the ROI image is divided into small patches and the corresponding patch masks from coloured mask are also saved. At the time of training, each coloured pixel is given a predefined class number by the dataloader, to compare the prediction of network with it.

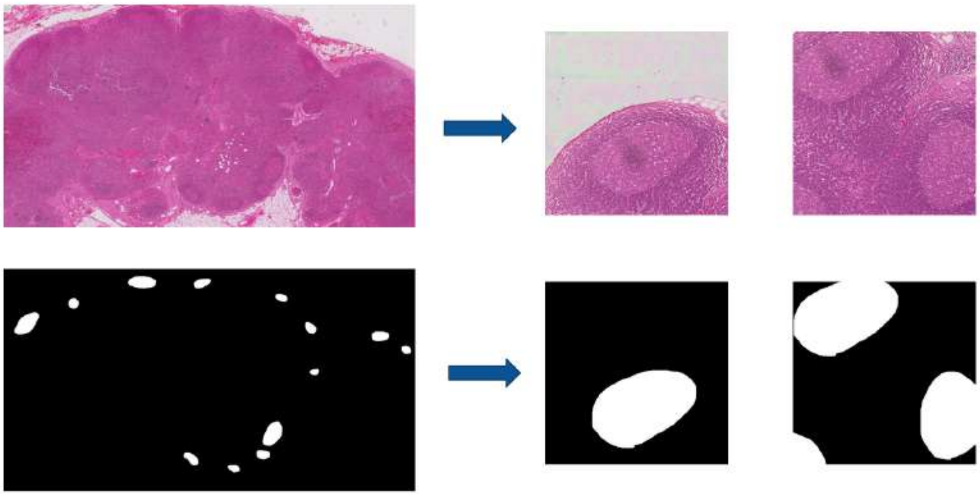


Figure 2.2: Data Preparation for one vs all classification.

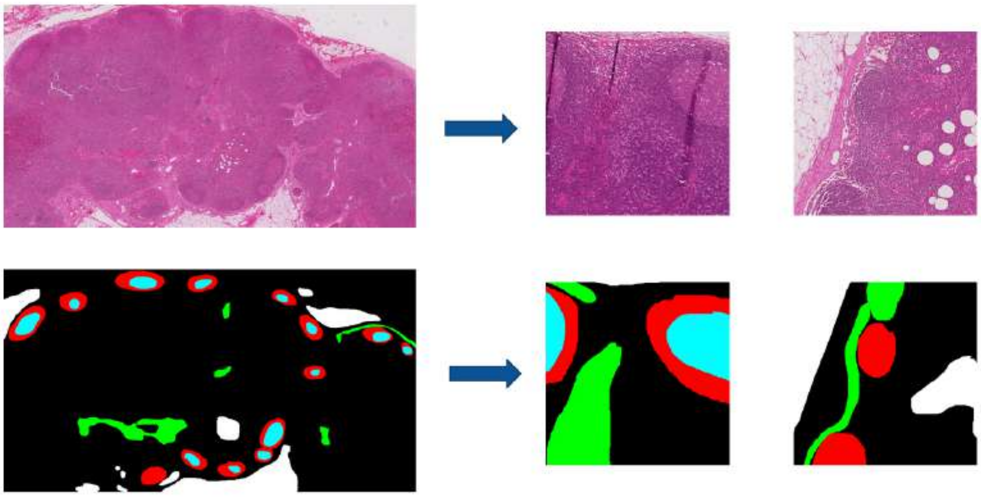


Figure 2.3: Data Preparation for multi class classification.

Chapter 3

Segmentation Methods and Tools

3.1 Overview

This chapter gives an overview of various techniques, network architectures, loss functions, data augmentations and evaluation metrics that were used in training experiments discussed in rest of this thesis. First I give a brief explanation of the two most important CNN architectures that were used. The first one is the widely used UNet architecture and the 2nd one is a multi-resolution architecture based on UNet and was developed by us. The multi-resolution model was found to perform better than a plain UNet in most of the experiments. In section 3.4, I discuss various loss functions that were used for training the models and for obtaining the results discussed in the next 3 chapters. In section 3.5 of this chapter I also discuss various data augmentation strategies, and especially the random rotations augmentation, the implementation of which required a slightly different way than the conventionally used techniques.

3.2 UNet Model

The UNet is a widely used network architecture for segmentation of biomedical images. It uses an encoder-decoder architecture. The most popular approach that is used for segmentation task is to use the encoder-decoder architecture. The encoder part successively downsamples the feature maps while increasing number of features per location. This way the features become increasingly meaningful for the purpose of semantic segmentation. The decoder part, along with skip connections, builds up the image segmentation map back to the size of image. The learned features can be used to reconstruct the image from the vector. Three main components/parts of UNet are- the contraction, the bottleneck, and the expansion. The contraction part effectively learns the complex features, the number of feature maps doubles after each block. The bottleneck part connects contraction and expansion part. The main important part of this architecture is the expansion layer which consists of CNN and upsampling layers. The feature maps decrease in number after each block and in addition to that features maps from the contraction layers are also added to reconstruct the output (with the help of skipped connections). The number of blocks on contraction side of UNet is same as the blocks in the expansion side.

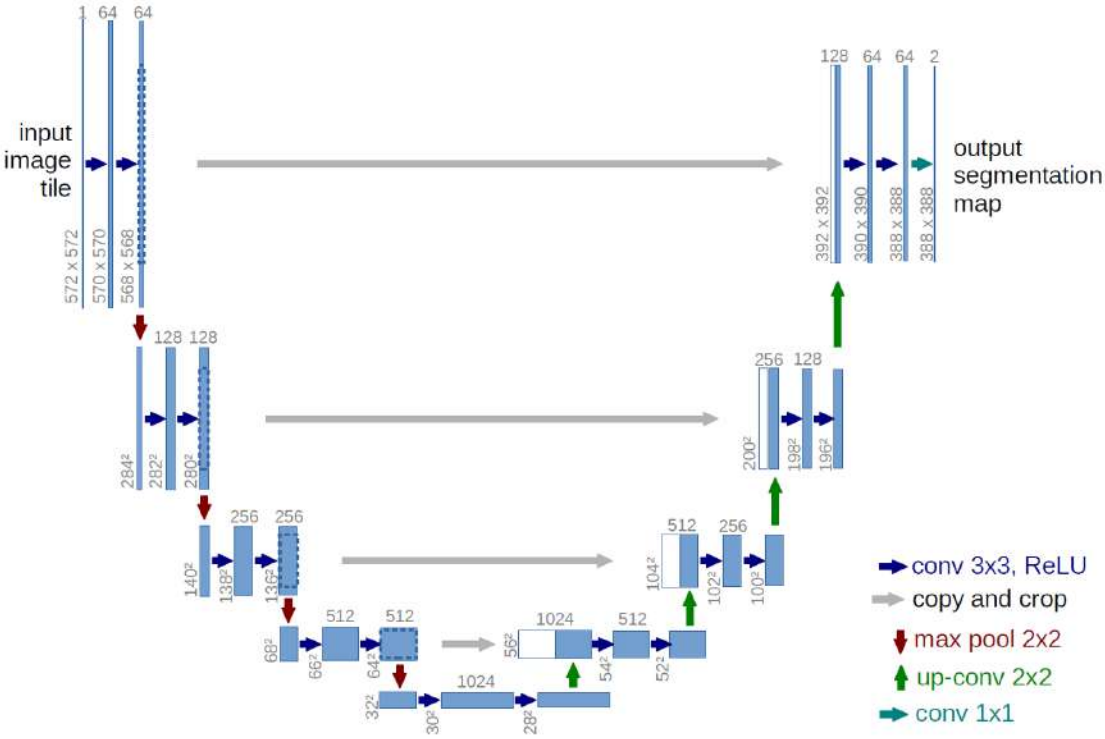


Figure 3.1: UNet Architecture [1]

3.3 The Multi-resolution Network Architecture

The architecture of the downsampling blocks of the multi-resolution model that was developed for the experiments in this project is shown in Fig 3.5. The high-level overall architecture is similar to that of a UNet (as shown in figure 3.2) but each downsampling block on the encoder side contains the modification capable of multi-resolution processing. In a normal UNet there are two convolutional layers and the output feature map is passed through a max-pooling layer. In the multi-resolution architecture, each block contains three sets of kernels in the first convolutional layer, each set of kernels for a different resolution, one set for the resolution of input to the block, one set with dilation 2 and stride 2 and hence processing at a half resolution of the input and 3rd set with dilation 4 and stride 4 and hence a processing at $1/4^{th}$ resolution of the input. A downsampling block of UNet can be seen in figure 3.4 and that of a multi-resolution network can be seen in figure 3.5. The outputs of three sets of kernels are passed through one more convolutional layer and are interpolated with bilinear interpolation to make them of same size and concatenated to make a single set of feature maps. This set of feature maps is then passed to the next downsampling block where the same sequence of operations is repeated after passing the feature maps through a max-pool layer.

3.4 Loss Functions

The loss functions used for various training experiments are the commonly used wieghted cross-entropy loss, L1 loss for noisy label segmentation and dice loss [6]. Among all these losses, including a combination of dice and cross-entropy, weighted cross-entropy loss was found to perform the best.

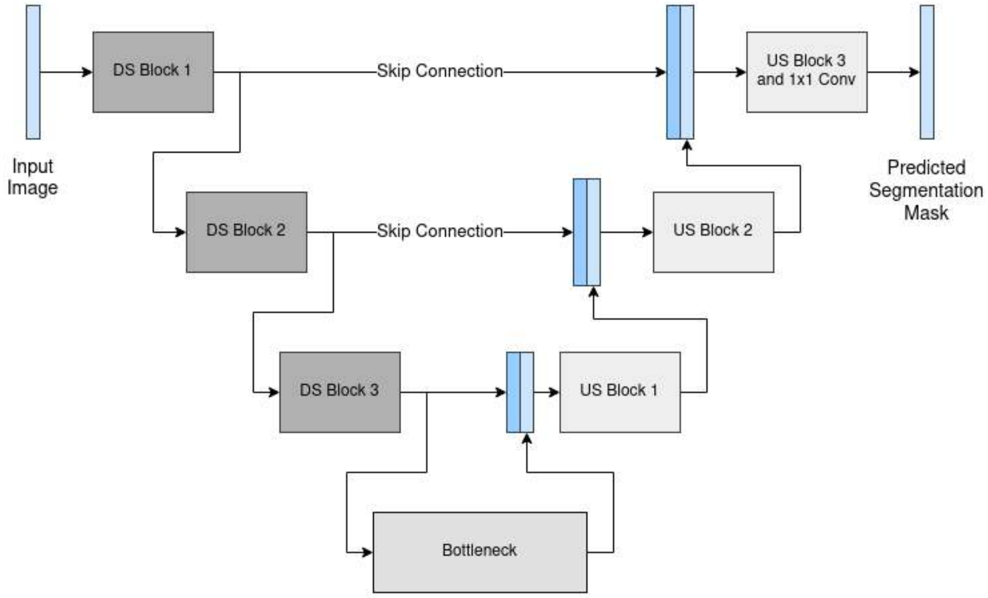


Figure 3.2: Block Level View of encoder-decoder segmentation architecture.

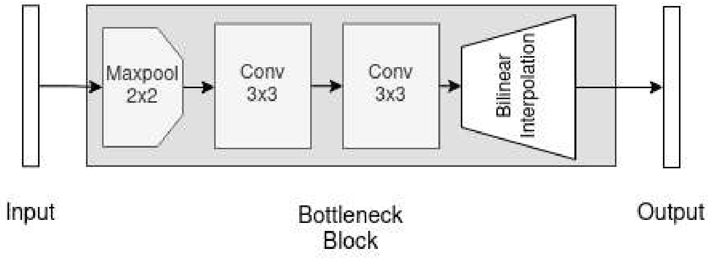


Figure 3.3: Inside view of bottleneck block

3.4.1 Weight Calculation Procedure for Weighted Cross Entropy Loss

To calculate the classwise weights for the weighted cross entropy loss, firstly the number of pixels belonging to each class are calculated and then total number of pixels in the whole dataset is divided by the number of pixels of each class. This way a weight inversely proportional to the amount of pixels of each class is obtained and is used to calculate the final cross entropy loss.

3.5 Data Augmentations

Following is a brief explanation of various data augmentations used in the training experiments discussed in this thesis:

- Random Horizontal and Vertical Flips: For lymph node image segmentation, if a patch is flipped horizontally or vertically along with its segmentation mask, then it is still a good data sample and can help learn the CNN kernels better.
- Color Jitter: As the H&E images may have a wide range of variability of colors, a small amount of Color Jitter data augmentation was used in the segmentation

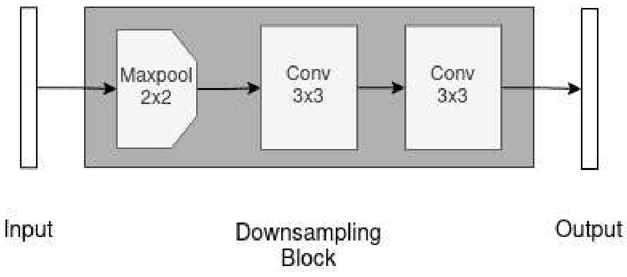


Figure 3.4: Inside view of a regular single-resolution segmentation network downsampling block

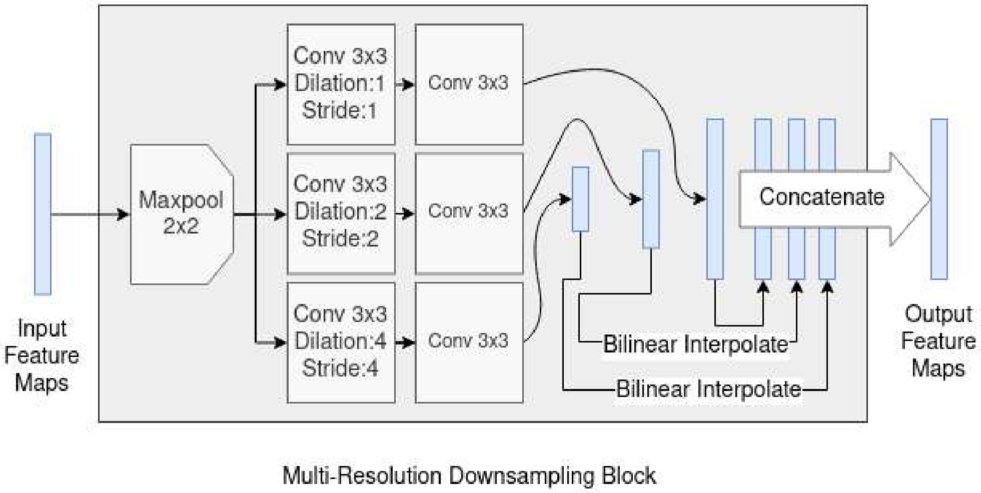


Figure 3.5: Inside view of a multi-resolution segmentation downsampling block

and was found to improve the segmentation results significantly.

- Random Rotations: Random rotation augmentation was also found to be useful in improving the segmentation results. For whole slide images, as the training is done at the patch level and final prediction is required to be done at the whole slide image level, it is important to feed the patches-without-any-artificial-background to the network, but when we rotate a square patch a significant amount of plain black or white background is added in the corner areas as shown in figure 3.6(c). Also we cannot just load the whole image and take a rotated patch from it as that would require an enormous amount of memory. So for the implementation of random rotations in the experiments, a slightly different approach of creating the 1.41 time larger patches (as shown in figures 3.6(a) and 3.6(e)) than the actual used for training was taken. With respect to the center of the larger patch (1.41 times the actual patch size), after randomly deciding a rotation angle, four coordinates of the rotated patch are calculated and the rotated patch is then extracted as square patch using a function such as affine warp .This way the actual patch is created by extracting a randomly rotated patch from the center of larger patches.

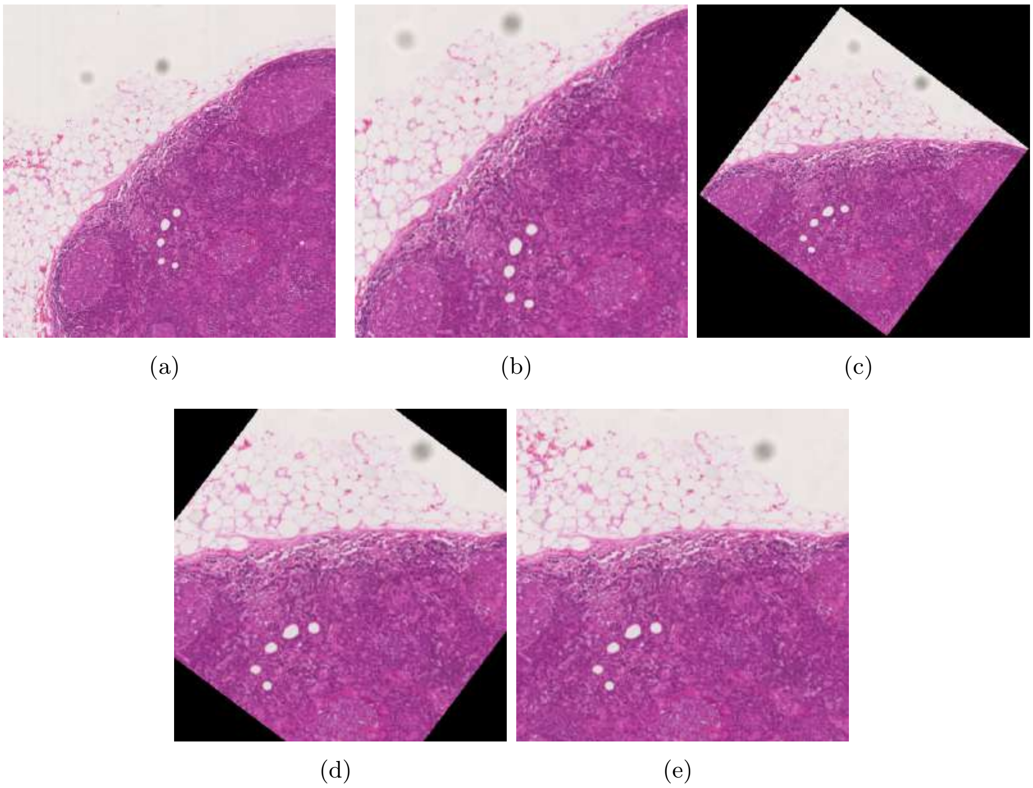


Figure 3.6: (a) 1.41 times larger patch than the actual patch to be extracted, (b) Actual patch used for training without rotations, (c) Rotate and resize approach of random rotations, (d) Rotate and center crop approach for random rotations, (e) Effective random rotations (ours)

3.6 Evaluation Metrics

3.6.1 IOU Score

IOU metirc also called Jaccard index. This metric helps to analyze the overlapping area percentage between prediction and ground truth mask in segmentation task.

The value of this metric ranges from 0-1. High overlapping regions gives IOU score close to 1 and 0 denotes no overlapping between ground truth and prediction.

$$IOU\ score = \frac{Area\ of\ overlap}{Area\ of\ Union} \tag{3.1}$$

3.6.2 Pixel-wise accuracy

This metric is simply calculated by the number of correctly classified pixels. The results of this metric can be mis-leading in case of class imbalance (where number of pixels of one class is higher than the other class).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Where, TP is True Positives, TN is True Negatives, FP is False Positive, FN is False Negatives. The numerator in the eq. 3.2 gives the correctly clasified pixels and the denominator is the total pixels.

3.6.3 Dice Score

This measures the overlap between the target and the predicted mask. Higher the value of dice-coefficient, lesser will be the loss acc. to eq 3.4. In case of class imbalance, weighted dice loss can also be used. The weights for this can be calculated same as the cross entropy weights discussed in section 3.2.1.

$$Dice\,Coefficient = \frac{2 \times Area\,of\,overlap}{Total\,area} \tag{3.3}$$

The best dice coefficient would have a value of 1.

$$Dice\,loss = 1 - Dice\,coefficient \tag{3.4}$$

3.6.4 Mean Average Precision

Calculation of IOU score requires a threshold value. It is difficult to find out an optimum threshold value that gives best result for a trained model and dataset. Mean Average Precision Score is a metric which is independent of threshold that is used to calculate the final results.

After calculating the precision and recall values for all the thresholds in the range 0-1 with a small step size, e.g. 0.01, we can draw a precision vs recall curve.

The average precision (AP) can be used to summarize the precision-recall curve into a single value which represents the average of all precisions. Average Precision is given by the following equation:

$$AP = \sum_{k=0}^{k=n-1} (Recalls(k) - Recalls(k+1)) * Precisions(k) \tag{3.5}$$

Where,
n = number of Thresholds
Recalls(n)=0, Precisions(n)=1

Chapter 4

Experiments and Results

4.1 Overview

A simple approach to solve the segmentation problem is to use a widely used CNN architecture such as a UNet with Cross-entropy loss for pixel-wise classification. A whole slide image can be divided into small patches and then each patch can be segmented. The segmentation masks of patches can then be combined to reconstruct the whole-slide level segmentation map. The initial experiments in the project began with that strategy. In the first set of experiments, discussed in section 4.2, all the experiments are based on UNet architecture with a variation of loss functions and other tweaks in training procedure. Section 4.3 discusses the experiments based on the multi-resolution architecture and center cropped loss calculation strategies.

4.2 Experiments-I: UNet with Cross-entropy and L1 loss

The results discussed in this section were obtained by using a UNet as the main architecture with several sizes of input patches. Random horizontal and vertical flip and color-jitter data augmentations were used for this set of experiments. Out of 45 images that were initially available, training was done on patches extracted from 35 image of lymph nodes, 5 were used for validation set and 4 were used for final testing. Different training experiments were conducted with patches taken from different resolutions to identify the resolution at which model can most successfully identify the concerned regions.

4.2.1 Segmentation at 2.5x resolution with UNet

Patches of size 256X256 were extracted from regions of interest of whole slide images and corresponding patches of masks of same size were also saved. UNet model was trained using a weighted cross entropy loss.The weights were calculated as explained in section 3.4.1. After the network was trained, testing was done on patches extracted from 4 test WSIs and final results were produced by stitching back to obtain the large ROI masks. Following figures contain the images of some results for Germinal Centre predictions.

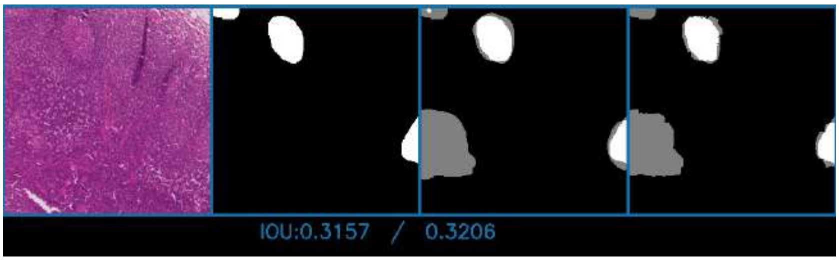


Figure 4.1: Germinal Center Prediction - Patch Level Sample-1
Leftmost (Patch from WSI), Middle-Left (Ground Truth), Middle-Right (Prediction with grey region indicating region of mismatch with GT), Rightmost(Prediction with boundary corrected with superpixel approach)

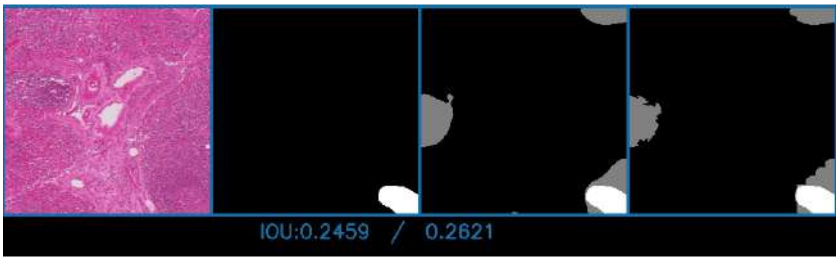


Figure 4.2: Germinal Center Prediction - Patch Level Sample-2
Leftmost (Patch from WSI), Middle-Left (Ground Truth), Middle-Right (Prediction with grey region indicating region of mismatch with GT), Rightmost(Prediction with boundary corrected with superpixel approach)

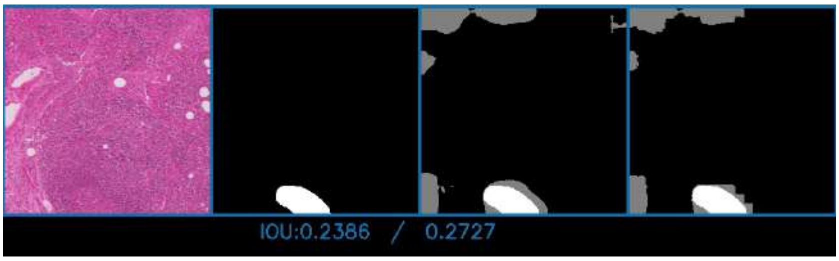


Figure 4.3: Germinal Center Prediction - Patch Level Sample-3
Leftmost (Patch from WSI), Middle-Left (Ground Truth), Middle-Right (Prediction with grey region indicating region of mismatch with GT), Rightmost(Prediction with boundary corrected with superpixel approach)

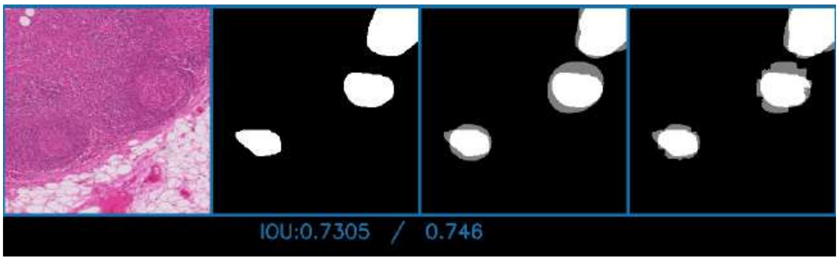


Figure 4.4: Germinal Center Prediction - Patch Level Sample-4
Leftmost (Patch from WSI), Middle-Left (Ground Truth), Middle-Right (Prediction with grey region indicating region of mismatch with GT), Rightmost(Prediction with boundary corrected with superpixel approach)

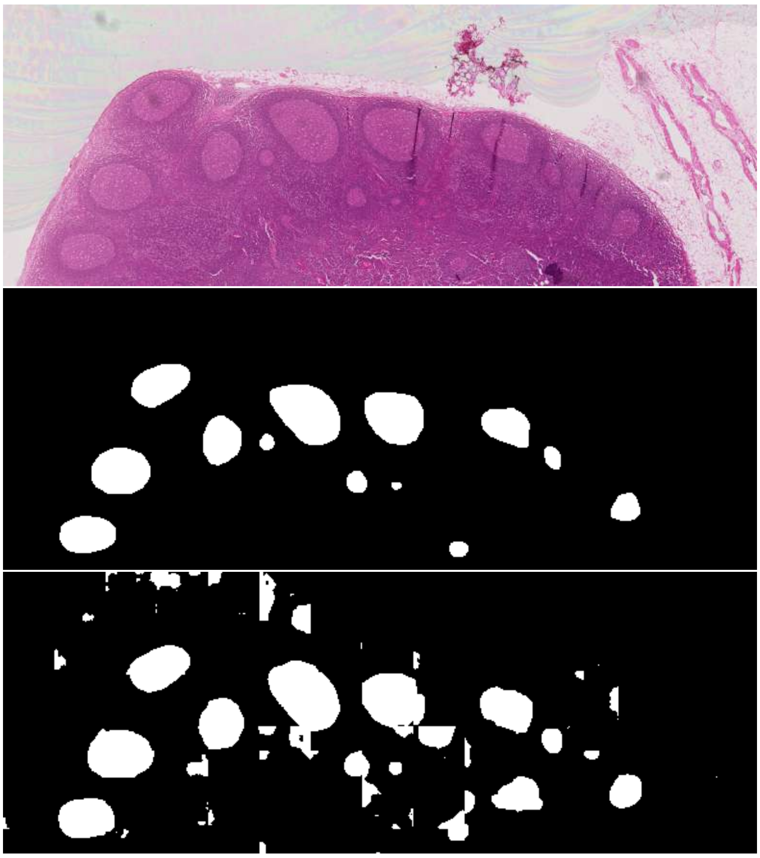


Figure 4.5: Germinal Centre prediction on whole region of interest at 2.5x resolution. Uppermost (Region of Interest from WSI), Middle (Ground Truth), Bottom (Prediction of the network)

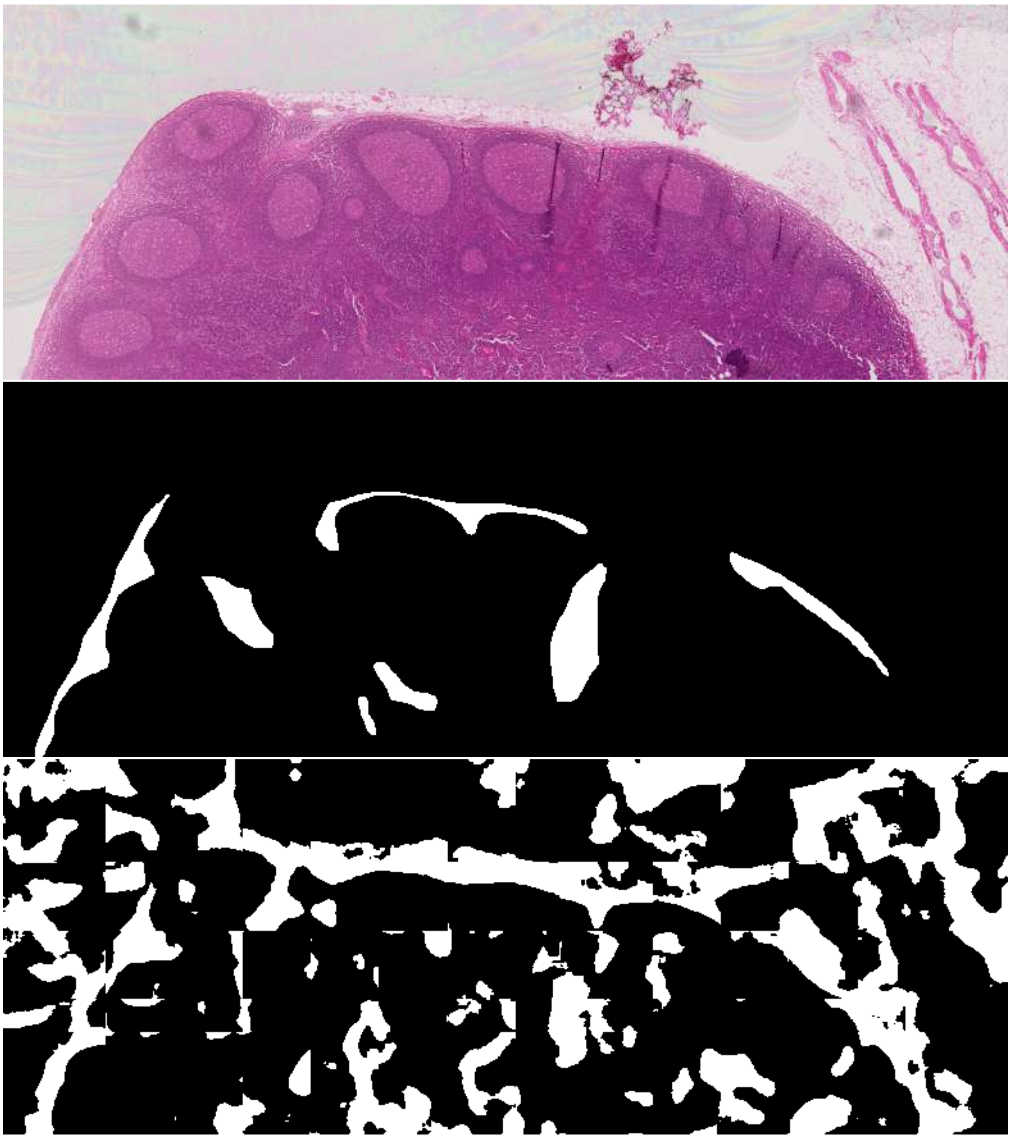


Figure 4.6: Sinus prediction on whole region of interest at 2.5x resolution. Uppermost (Region of Interest from WSI), Middle (Ground Truth), Bottom (Prediction of the network)

4.2.2 Segmentation at 5x resolution with UNet

Germinal Center regions are the most important regions from metastasis correlation point of view. As the results obtained after segmentation on 4X resolution were not quite convincing and good enough to be used for further analysis of cancerous features, some more experiments for segmentations were tried on 5x resolution. Similar to the training method of 2.5x, segmentation experiments were conducted at 5x resolution with patch size of 512x512 this time. Following are some results obtained from one vs all as well as multi class segmentation experiments:

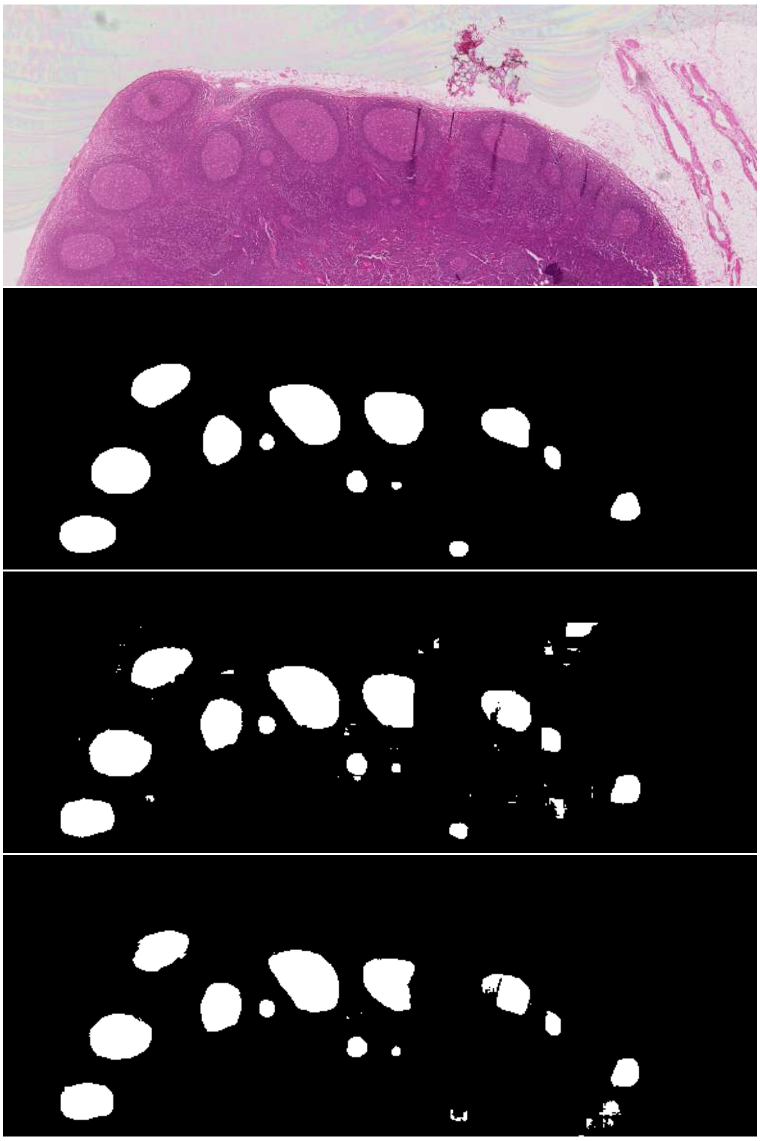


Figure 4.7: Germinal Centre prediction on whole region of interest.
Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network), bottom (prediction of the network when whole ROI image is fed at once to the fully convolutional network for prediction.)

After comparing the results in Fig. 4.5 and Fig. 4.7, we can say that segmentation seeems to be working better at 5x resolution as compared to the 2.5x resolution. Below figures contain some more results for other classes obtained at 5x resolution.

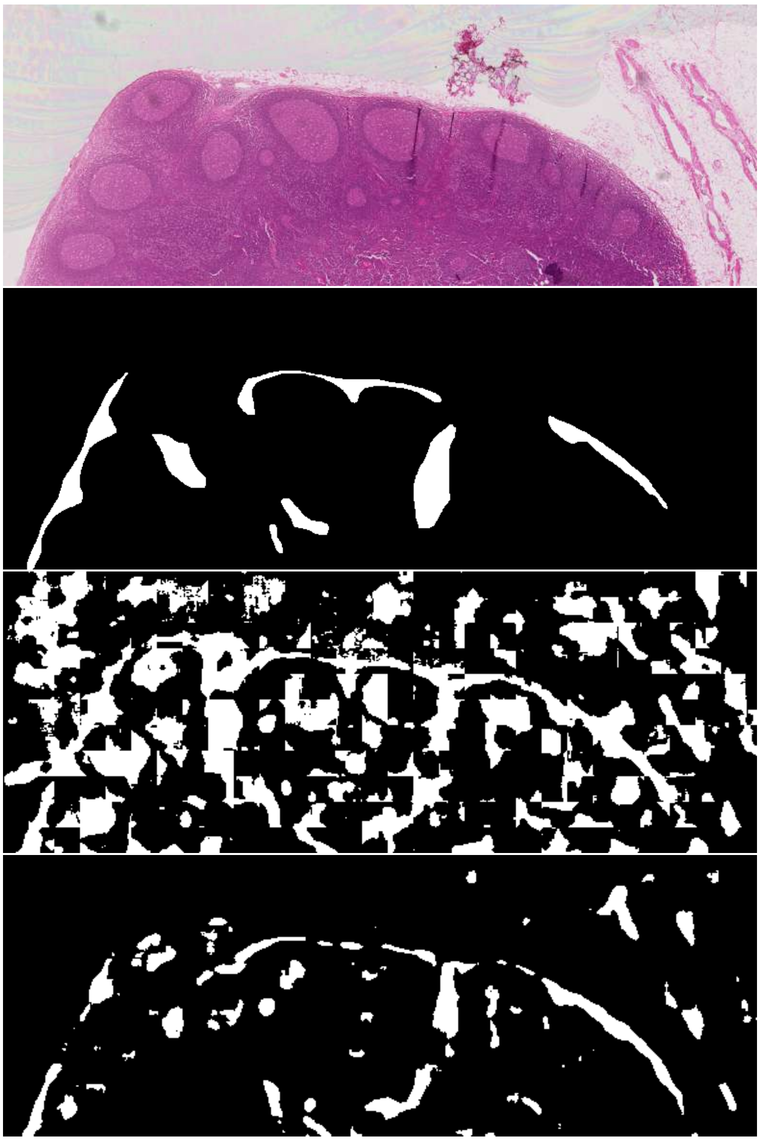


Figure 4.8: Sinus prediction on whole region of interest
Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network), bottom (prediction of the network when whole ROI image is fed at once to the fully convolutional network for prediction.)

As can be seen from the above figure, passing a whole ROI image through a fully convolutional part of trained network and then getting the predictions has a lot more advantage and better overall results. But as the whole slide images are quite large, it is not always possible to get a prediciton by feeding the whole image to the CNN given the hardware constraint. For example predicting the above results required a RAM of more than 100GB. In our lab, the highest RAM computer has 64GB of RAM and a lot of swap space was used which makes the process extremely slow.

Figure 4.2.2 contains the results after combining the predictions of all the classes for the example WSI region we have been using so far.

4.2.3 Multiclass segmentation with UNet on 5x resolution

The results so far include the one-vs-all class segmentation and their combined results. A multiclass segmentation approach was also tried in several experiments. Although after analyzing the results of various multiclass as well as one-vs-all class segmentation experiment, it was observed that one vs all classification in general has produced better results for this problem. This section includes the results of one such multi class segmentation experiment, the network took around 40 hours to train on a 12GB Nvidia GPU. Following are the results on the example image we have been using so far.

4.2.4 Summary of results on 5x resolution

Fig. 4.11 contains the summary of results obtained with various experiments on 8X resolution. The results are measured as Intersection of Unions score. FCP is the notation for the results obtained when the input to the network is the whole ROI image. Hence FCP results are those when prediction is obtained by passing the whole image at once through the trained network without dividing it into patches. The networks were tested on a total of 4 whole slide images numbers 1 to 4 in the table. As can be seen from the table, the best IOU score is obtained for adipose region, which is 0.567 (Average of all the whole slide images in test set). The worst is for sinus. When the prediction is obtained by feeding the whole image to CNN, a significant improvement in the IOU score is obtained, as can be seen for all the classes. Combined IOU in the table means the overall IOU for all the classes after a combined result was obtained by putting all the results onto a single mask (Combination was obtained by putting the results in the order adipose, sinus, follicle, GC). When a combined result is obtained then regions of overlap are taken by class whose result is put after other classes.

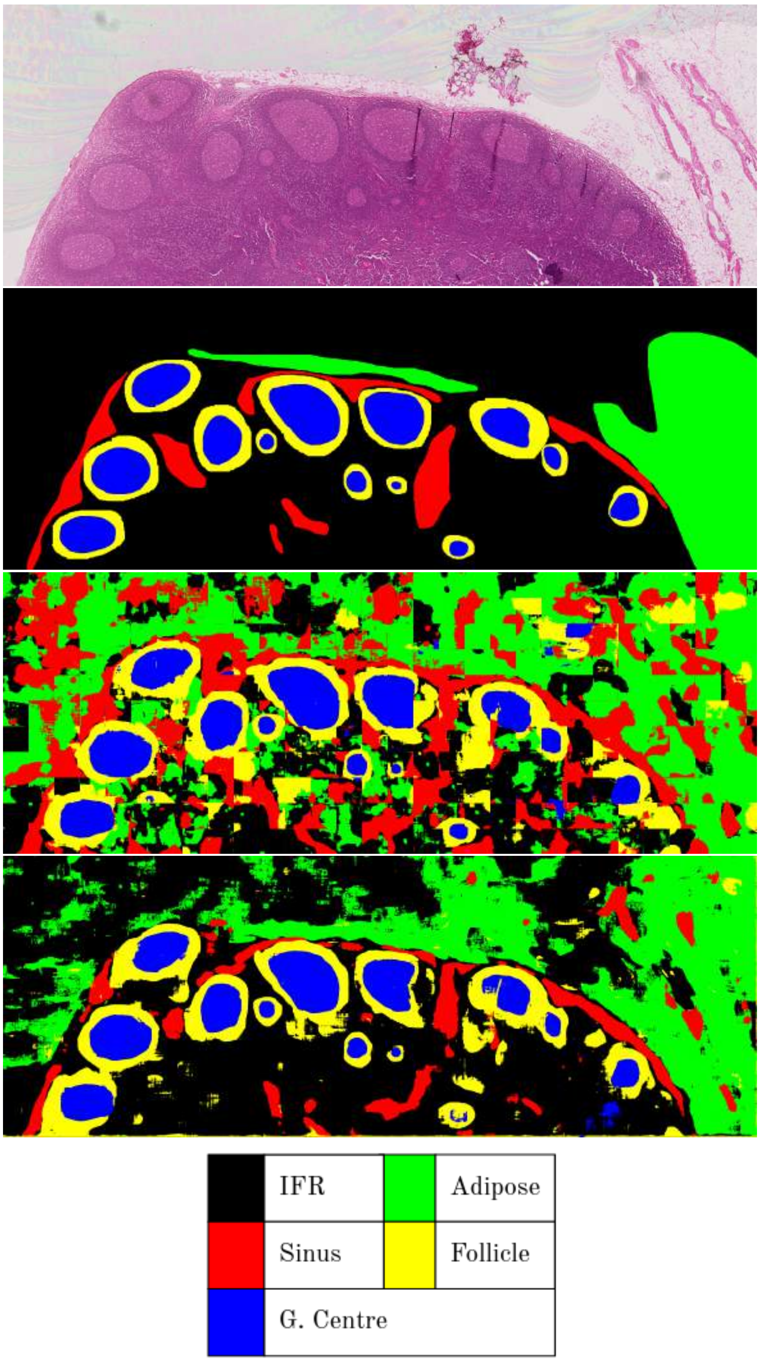


Figure 4.9: Prediction on whole region of interest.
Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network), bottom (prediction of the network when whole ROI image is fed at once to the fully convolutional network for prediction.)

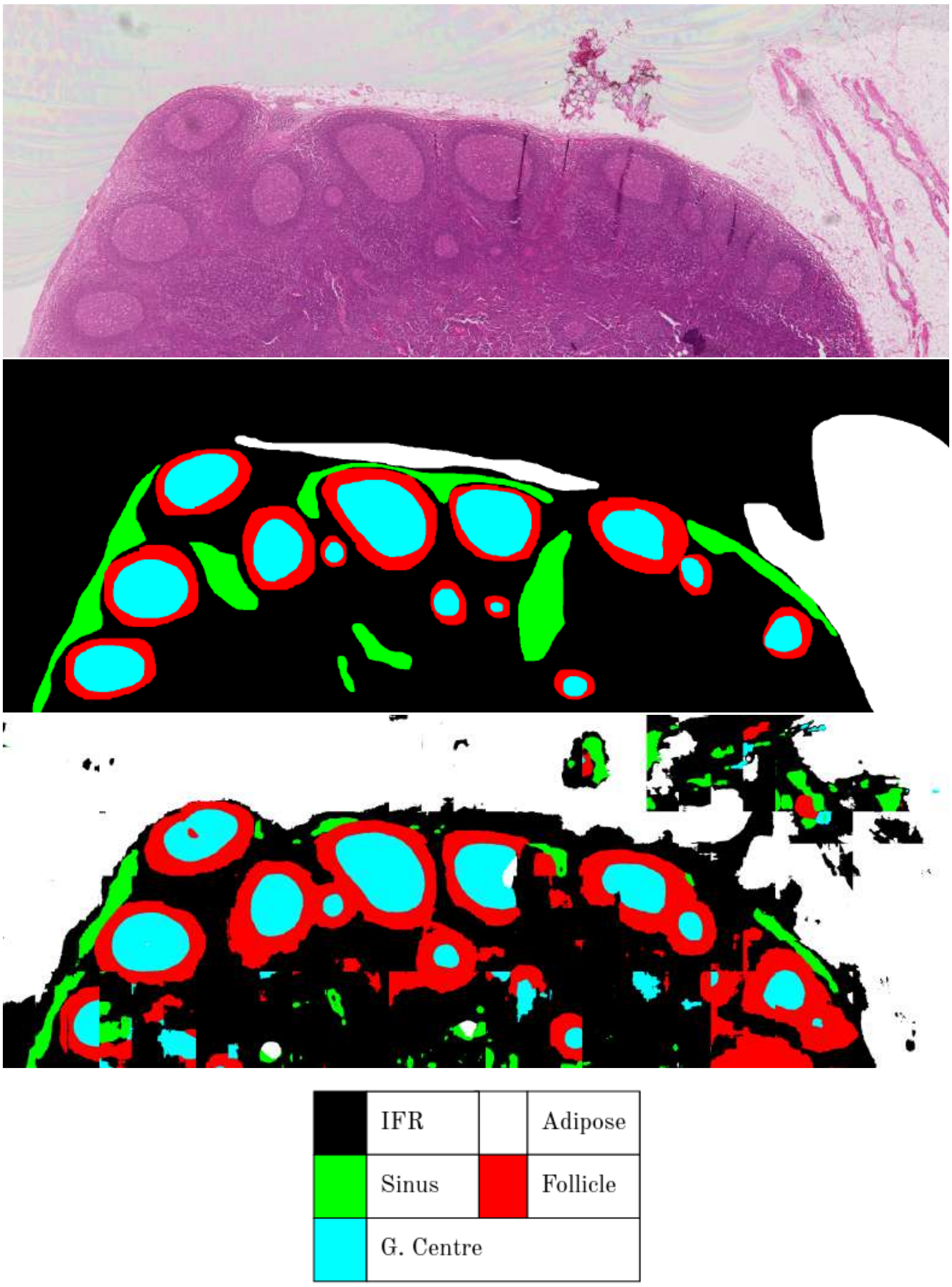
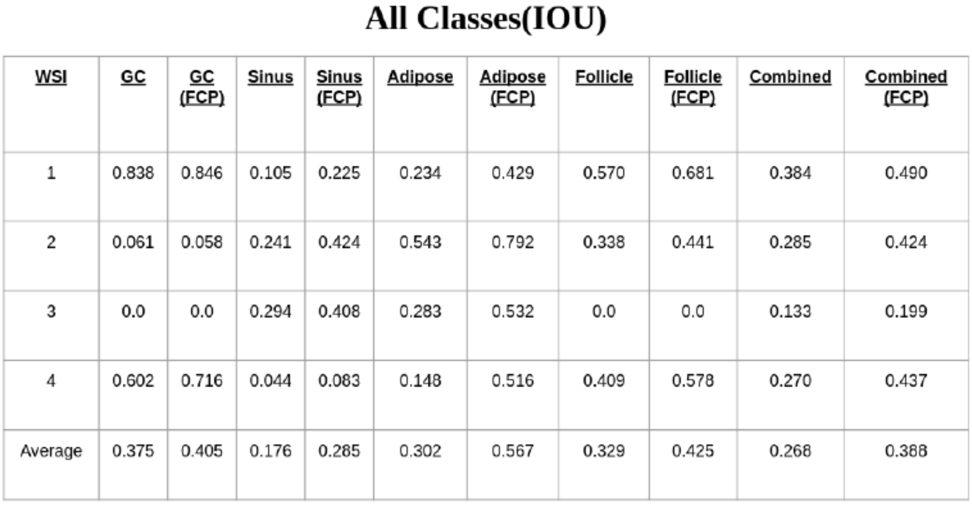


Figure 4.10: Prediction on whole region of interest
Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network)



All Classes(IOU)

WSI	GC	GC (FCP)	Sinus	Sinus (FCP)	Adipose	Adipose (FCP)	Follicle	Follicle (FCP)	Combined	Combined (FCP)
1	0.838	0.846	0.105	0.225	0.234	0.429	0.570	0.681	0.384	0.490
2	0.061	0.058	0.241	0.424	0.543	0.792	0.338	0.441	0.285	0.424
3	0.0	0.0	0.294	0.408	0.283	0.532	0.0	0.0	0.133	0.199
4	0.602	0.716	0.044	0.083	0.148	0.516	0.409	0.578	0.270	0.437
Average	0.375	0.405	0.176	0.285	0.302	0.567	0.329	0.425	0.268	0.388

Figure 4.11: Summary of results on 5x resolution

4.2.5 Segmentation with Unet at 10x for Germinal Centers

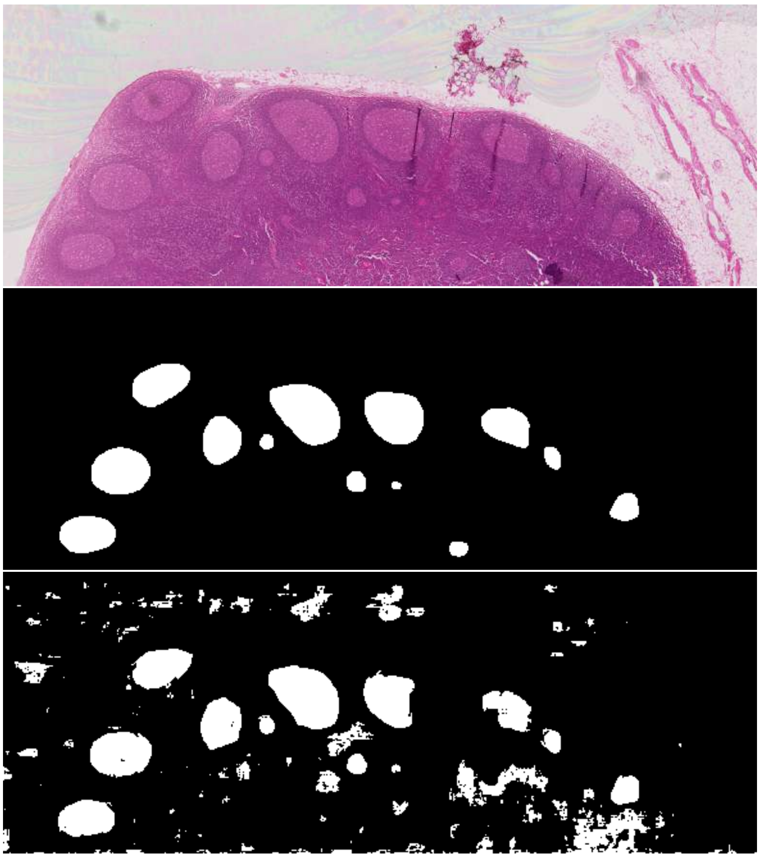


Figure 4.12: Prediction on whole region of interest using UNet model 10x resolution, Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network)

4.3 Experiments-II: Multiresolution Model with Crossentropy Loss

Multi resolution architecture which is explained in section 3.3 was trained using patches of dimensions 512x512 for 5x resolution and patches of dimensions 1024x1024 for 10x resolution. Although the network is being called a multi-resolution network, the input is processed at multiple resolutions by downsampling it inside the network. So if the resolution of input is 10x, the first layer of network processes resolutions of 10x, 5x and 2.5x and if the resolution of input patch is 5x then the resolutions at which network's first layer acts on are 5x, 2.5x and 1.25x. As the Multi-resolution model performs many operations inside the network, it took more time to train than the plain UNet models. For training on 5x resolution, it took 40-45 hours on average to train the network and 70-80 hours for training on 10x resolution patches. Following figures contain the results of multi-resolution network for the example WSI that has been used so far.

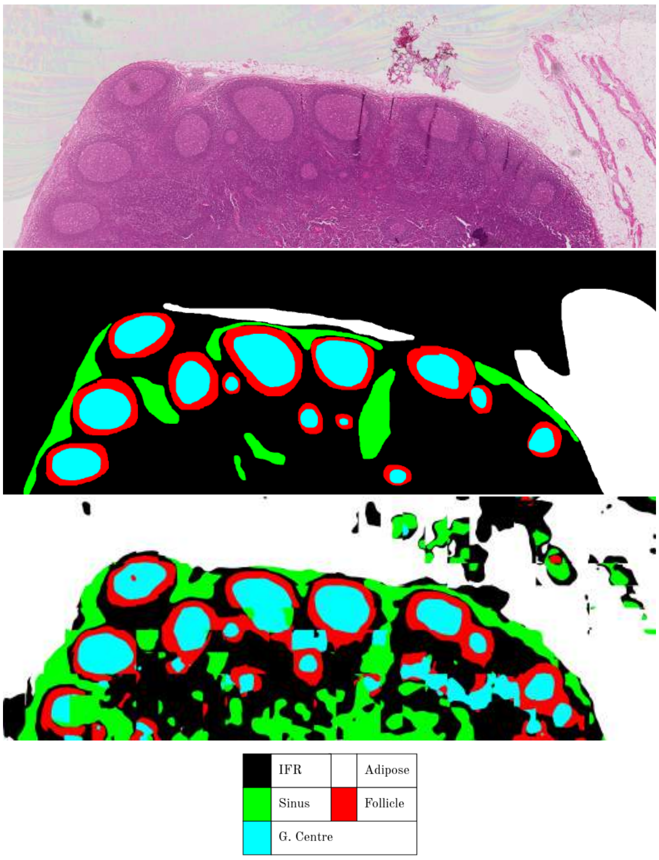


Figure 4.13: Multiclass segmentation of whole region of interest
Multi-resolution model with inputs at 5x resolution, Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Prediction of the network)

Germinal Centre segmentation with multiresolution architecture on 10x resolution

Patches of 10x resolution, 1024x1024 dimensions were taken from the whole slide images to train the architecture described in section 4.3. It took around 100 hours to train this network on a 12GB Nvidia GPU for a dataset of 39 Whole slide images. It was observed that, of all the experiments for segmentation of Germinal Centre region, the best results were obtained in with this architecture and resolution setting.

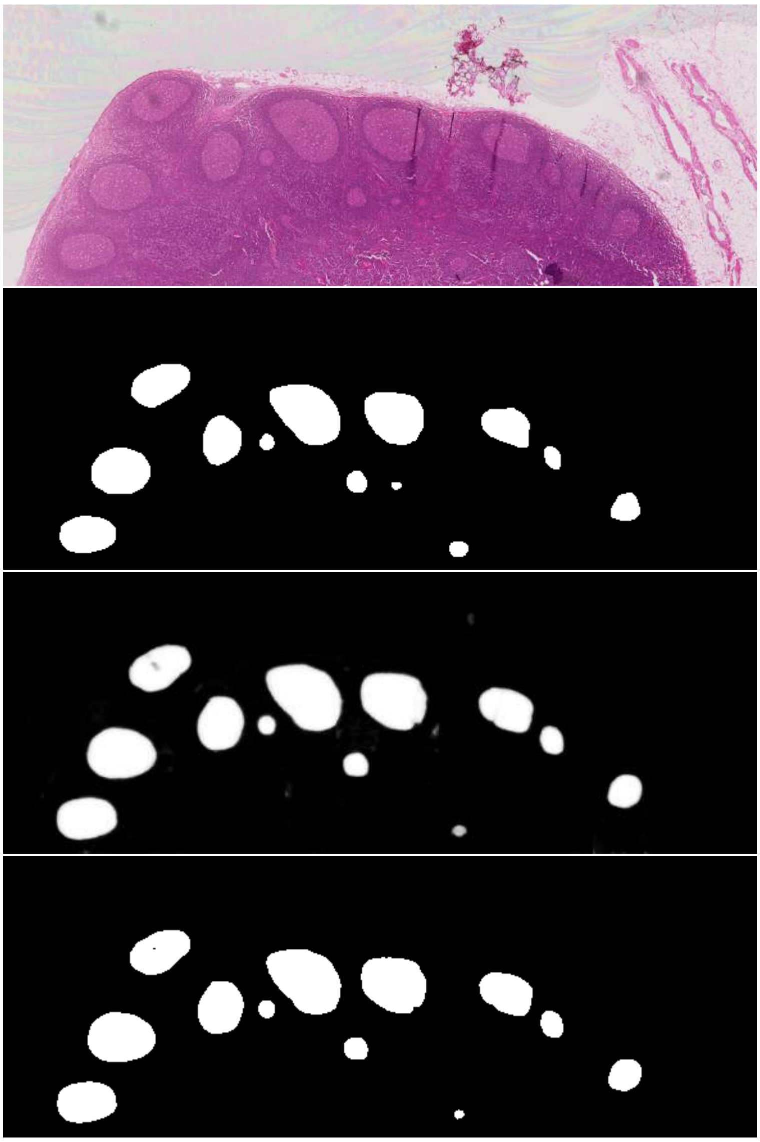


Figure 4.14: Prediction on whole region of interest for Germinal Centre Uppermost (Region of Interest from WSI), 2nd from top (Ground Truth), 3rd from top (Grayscale prediction of the network), bottom (prediction of the network by thresholding grayscale image using best average threshold on results.)

Using the grayscale output of the network, a best threshold was calculated by varying the threshold and comparing the thresholded results with ground truth. Fig. 4.15 shows a plot of threshold vs the IOU for Germinal Center predictions.

The abrupt jump in the IOU in Fig. 4.15 is because one of the four WSIs used for

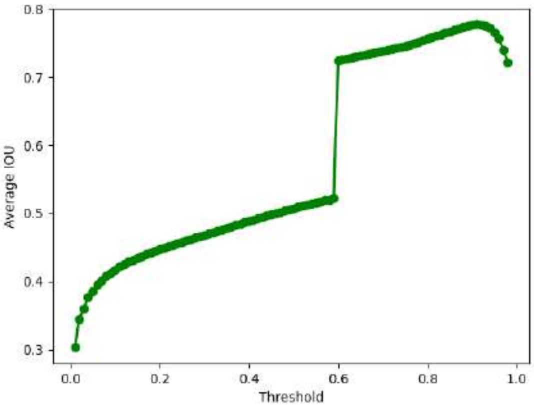


Figure 4.15: Threshold vs IOU

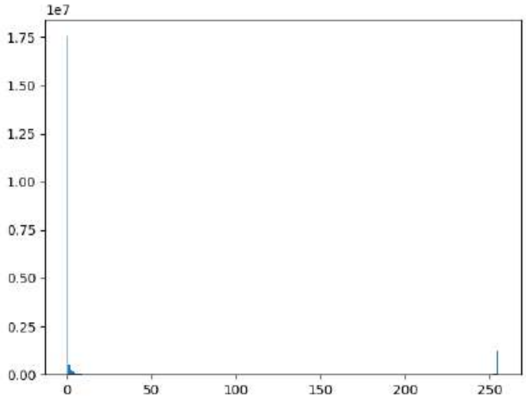
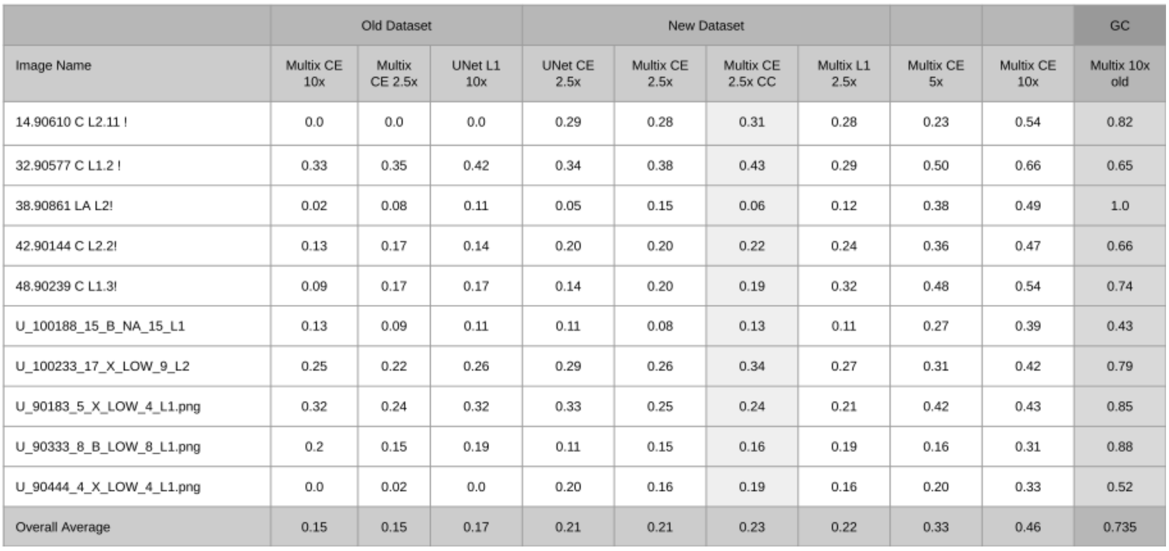


Figure 4.16: Histogram of grayscale values for prediction of image used in the example result

testing doesn't have any Germinal Center and using threshold greater than round 0.6 makes its individual IOU to jump from 0 to 1, but the peak if obtained at around 0.7, which, as is expected to be general best IOU has been used to obtained the final results.

Fig. 4.16 shows a histogram of grayscale values for the prediction of the network for image in Fig. 4.14. We can see the most values of pixel intensity are clustered either around 0 or 1 which means that network is quite confident in its predictions.

4.3.1 Summary of Results and Comparison of Models



	Old Dataset			New Dataset						GC
Image Name	Multix CE 10x	Multix CE 2.5x	UNet L1 10x	UNet CE 2.5x	Multix CE 2.5x	Multix CE 2.5x CC	Multix L1 2.5x	Multix CE 5x	Multix CE 10x	Multix 10x old
14.90610 C L2.11 !	0.0	0.0	0.0	0.29	0.28	0.31	0.28	0.23	0.54	0.82
32.90577 C L1.2 !	0.33	0.35	0.42	0.34	0.38	0.43	0.29	0.50	0.66	0.65
38.90861 LA L2!	0.02	0.08	0.11	0.05	0.15	0.06	0.12	0.38	0.49	1.0
42.90144 C L2.2!	0.13	0.17	0.14	0.20	0.20	0.22	0.24	0.36	0.47	0.66
48.90239 C L1.3!	0.09	0.17	0.17	0.14	0.20	0.19	0.32	0.48	0.54	0.74
U_100188_15_B_NA_15_L1	0.13	0.09	0.11	0.11	0.08	0.13	0.11	0.27	0.39	0.43
U_100233_17_X_LOW_9_L2	0.25	0.22	0.26	0.29	0.26	0.34	0.27	0.31	0.42	0.79
U_90183_5_X_LOW_4_L1.png	0.32	0.24	0.32	0.33	0.25	0.24	0.21	0.42	0.43	0.85
U_90333_8_B_LOW_8_L1.png	0.2	0.15	0.19	0.11	0.15	0.16	0.19	0.16	0.31	0.88
U_90444_4_X_LOW_4_L1.png	0.0	0.02	0.0	0.20	0.16	0.19	0.16	0.20	0.33	0.52
Overall Average	0.15	0.15	0.17	0.21	0.21	0.23	0.22	0.33	0.46	0.735

Figure 4.17: Summary of results obtained with experiments in section 4.2 and 4.3.

Figure 4.3.1 shows the results obtained with different loss functions and UNet and Multiresolution models. It can be concluded from the table of results in figure 4.3.1 that multiresolution model always gives a better performance for segmentation than a UNet. Also, crossentropy loss seems to be a better choice than L1 loss.

4.3.2 Sinus Segmentation with Multi-resolution Model with 10x input and Random Rotations

In this subsection, results obtained by using random rotations data augmentation with multi-resolution model, along with center cropped loss calculation strategy are presented. The random rotations data augmentation was found to improve the results considerably. The green color represents the true positives or the correct predictions, red color represents false positives and blue regions indicate false negatives.

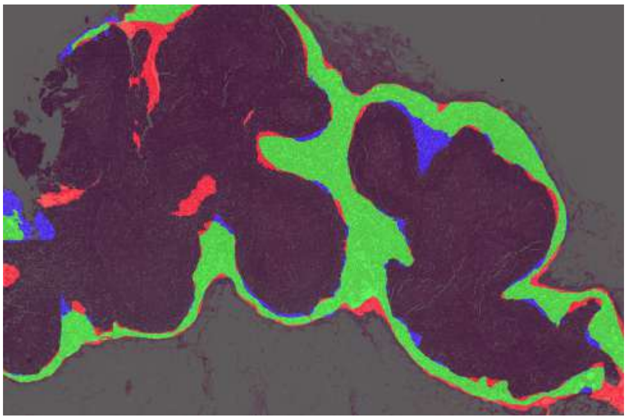


Figure 4.18: Example-1, Segmentation with multiresolution model at 10x resolution and random rotations data augmentations
Green: True Positives, Red: False Positives, Blue:False Negatives

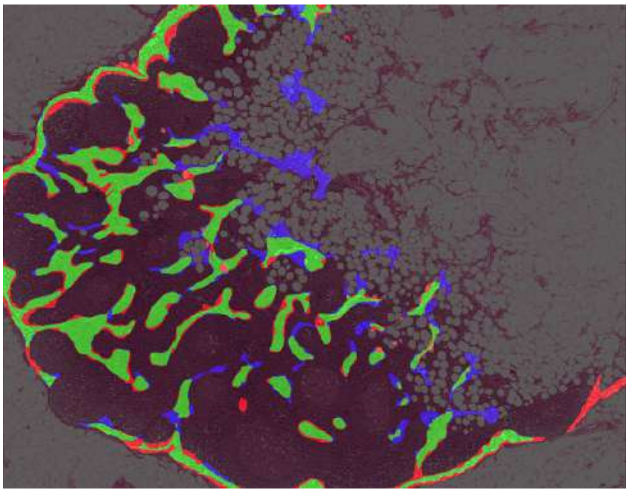


Figure 4.19: Example-2, Segmentation with multiresolution model at 10x resolution and random rotations data augmentations
Green: True Positives, Red: False Positives, Blue:False Negatives

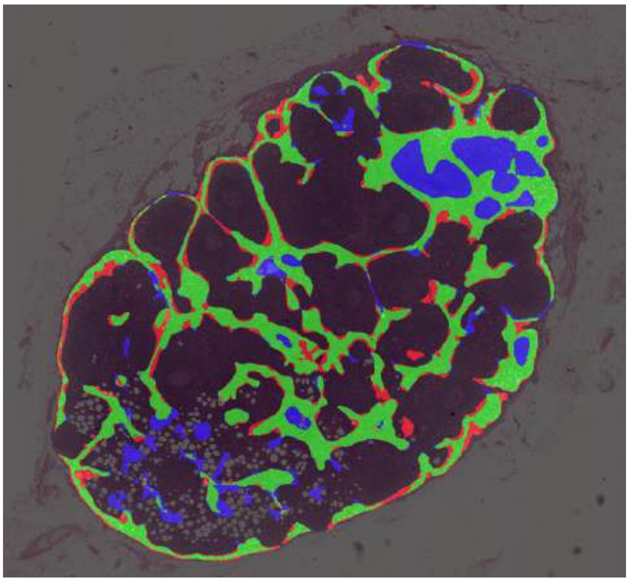


Figure 4.20: Example-3, Segmentation with multiresolution model at 10x resolution and random rotations data augmentations
Green: True Positives, Red: False Positives, Blue:False Negatives

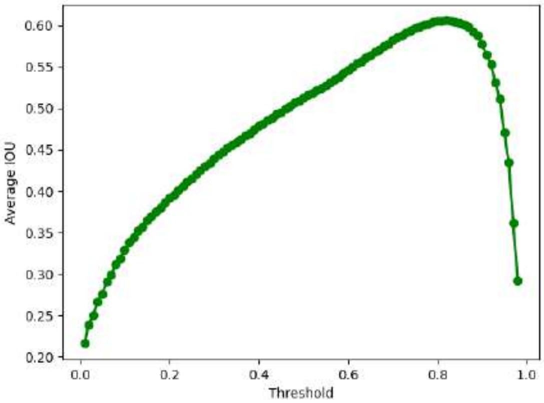


Figure 4.21: Threshold-vs-IOU curve for multi-resolution model predictions

4.4 Experiments-III: Multiresolution Model with Fuzzy Boundaries and Bootstrapping

4.4.1 Fuzzy Boundaries Training

This section includes the results obtained by training a multiresolution model with fuzzy boundaries technique. In this technique, while computing the loss for the prediction of an example, an additional mask is computed by blurring the edges in the segmentation mask. This blurred mask is then subtracted from the original mask and the absolute value at each location is taken. This mask is then normalized and subtracted from a mask of 1's.

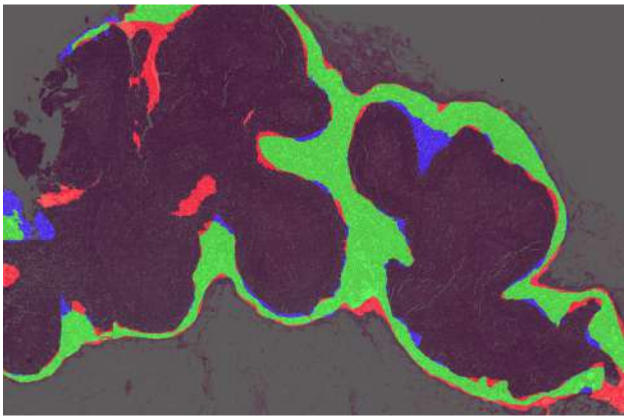


Figure 4.22: Example-1, Segmentation with multiresolution model at 10x resolution and fuzzy boundaries training
Green: True Positives, Red: False Positives, Blue:False Negatives

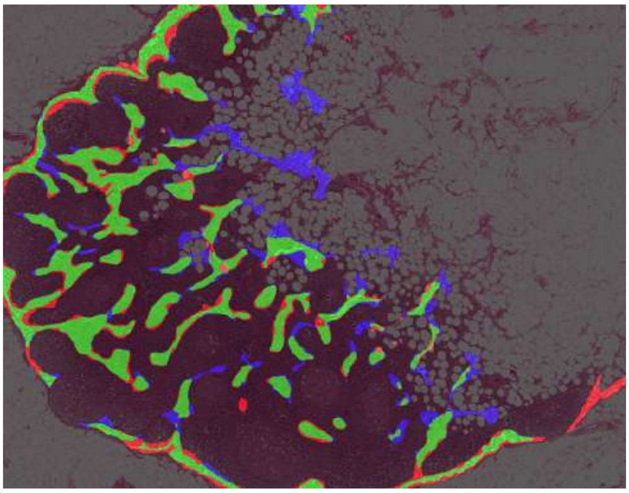


Figure 4.23: Example-2, Segmentation with multiresolution model at 10x resolution and fuzzy boundaries trainig
Green: True Positives, Red: False Positives, Blue:False Negatives

This way we finally obtain a mask indicating the confidences of 1's at the regions away from any kind of boundaries and smaller values closer to and on the boundaries depending on the gradient of each location. This mask is then multiplied with the crossentropy loss values calculated for respective locations. Finally the overall loss is averaged. This way, the values closer to boundaries contribute less in the loss calculation and the values away from boundaries contribute more and technically we let the model decide for the locations closer to boundaries and don't force it to learn the labels marked in annotations.

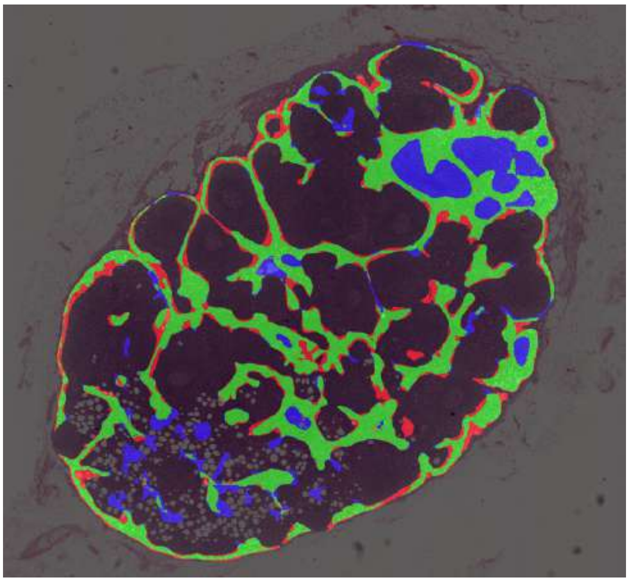


Figure 4.24: Example-3, Segmentation with multiresolution model at 10x resolution and fuzzy boundaries training
Green: True Positives, Red: False Positives, Blue:False Negatives

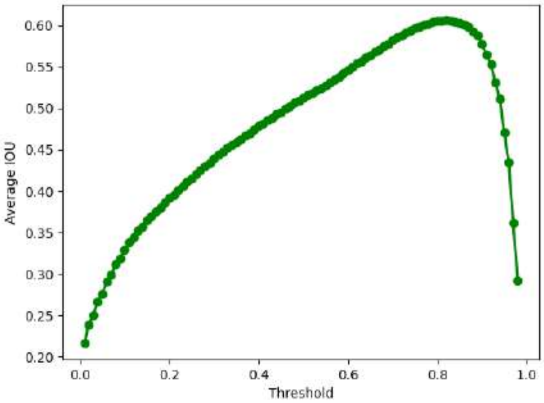


Figure 4.25: Threshold vs IOU curve for multi-resolution model with fuzzy boundaries training

4.4.2 Bootstrapping

The annotation procedure is quite laborious and the provided annotations are almost always inevitably noisy especially with a large amount of incorrectly marked pixels for regions like sinus and follicle where even pathologist may disagree on the correct label for a region of pixels. This way, while training, we inevitable force the network to learn some wrong labels for some of the regions and that makes it to produce poor results at the test time. One way to deal with this problem is to train the network for a few epochs where it gets a good idea of regions which can clearly be said to belong to a class or not and then also give a weightage to the prediction of the network in loss calculation.

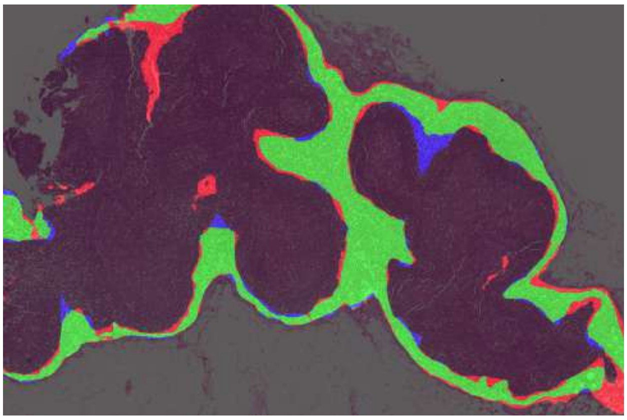


Figure 4.26: Example-1, Segmentation with multiresolution model at 10x resolution and bootstrapping strategy
Green: True Positives, Red: False Positives, Blue:False Negatives

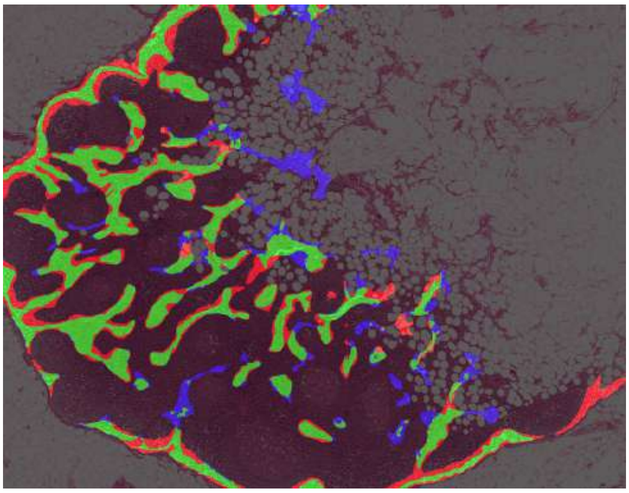


Figure 4.27: Example-2, Segmentation with multiresolution model at 10x resolution and bootstrapping strategy
Green: True Positives, Red: False Positives, Blue:False Negatives

This way, in addition to the ground truth mask used for segmentation, the prediction of network itself is also used to compute the loss by deciding some weightage for it. The consideration of the networks prediction begins after training it for a number of epochs initially which is a hyperparameter. The results presented in this section were obtained by using such a strategy and seem usually better than the results obtained with the strategies in previous sections, although the average IOU remained the same as the fuzzy boundaries method. The IOU score seems to be limited by the noisy annotations provided by pathologists in test images.

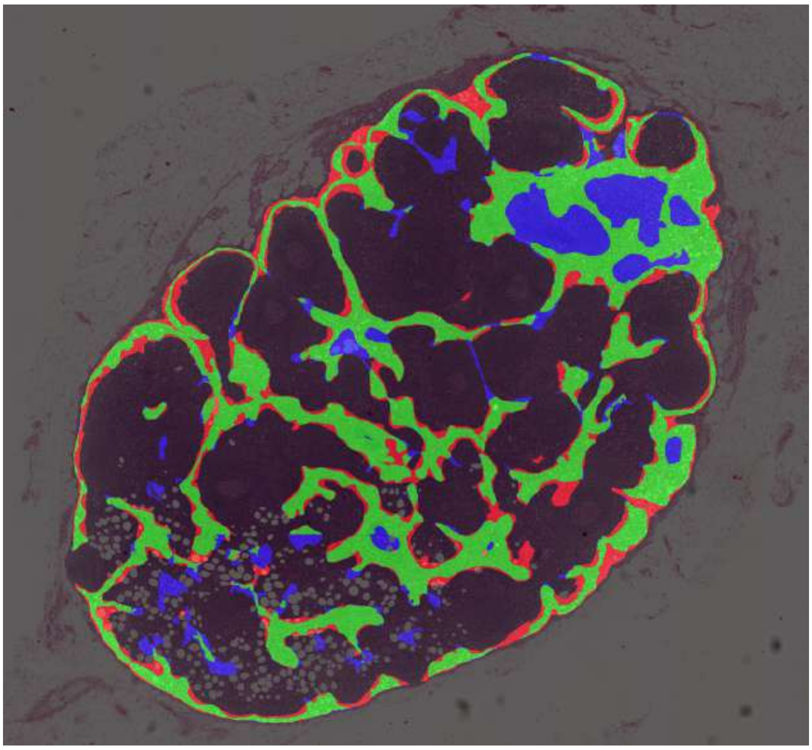


Figure 4.28: Example-3, Segmentation with multiresolution model at 10x resolution and bootstrapping strategy
Green: True Positives, Red: False Positives, Blue:False Negatives

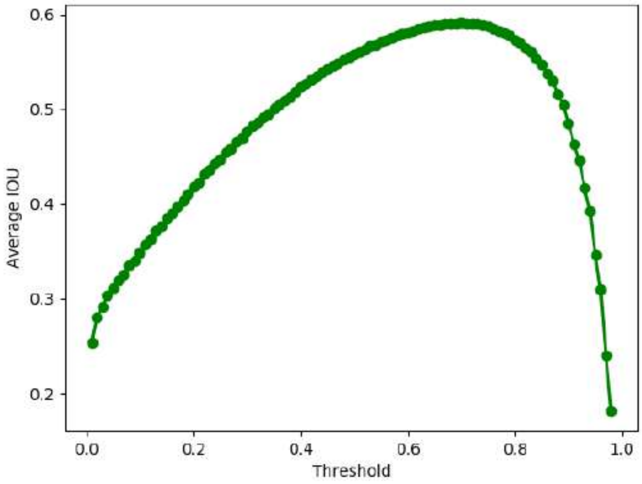
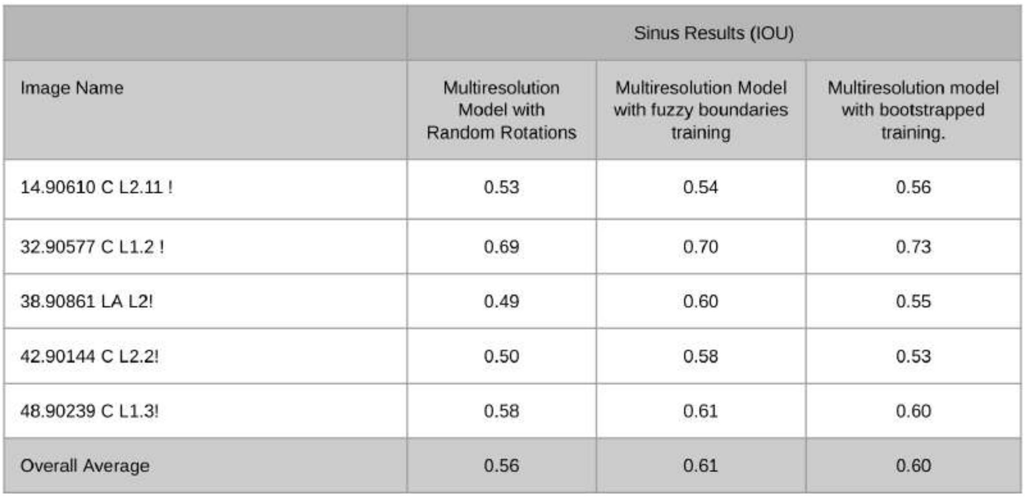


Figure 4.29: Threshold vs IOU curve of results obtained with bootstrapping experiments.

4.4.3 Summary of results of Experiments-III



	Sinus Results (IOU)		
Image Name	Multiresolution Model with Random Rotations	Multiresolution Model with fuzzy boundaries training	Multiresolution model with bootstrapped training.
14.90610 C L2.11 !	0.53	0.54	0.56
32.90577 C L1.2 !	0.69	0.70	0.73
38.90861 LA L2!	0.49	0.60	0.55
42.90144 C L2.2!	0.50	0.58	0.53
48.90239 C L1.3!	0.58	0.61	0.60
Overall Average	0.56	0.61	0.60

Figure 4.30: Summary of results obtained with experiments in section 4.4.

4.5 Experiments-IV: Summary and Comaprison of Results

Guy's hospital dataset, as discussed in chapter 2, contains two sub datasets. Out of 77 images, 29 images are more carefully marked as compared to the rest 48 images. For this reason, the more carefully annotated 29 images shoudl be given more importance in training as compared to the less carefully marked 48 images. So, during training of experiments discussed in this chapter, if a patch came from the less noisy dataset (more carefully marked 29 whole-slide images), its loss is multiplied with a higher weight as compared to the loss of a patch which comes from the noisy (less carefully annotated dataset of 48 whole-slide images). Most of the experiment where not mentioned, the weight ration of noisy:less-noisy dataset is 3:2.

Wherever not mentioned, for bootstrapping experiments, the bootstrapped training was done after first fully training the network with all other augmentations and loss calculation strategies and after the best model was saved according to the validation loss, bootstrapped fine tuning was done.

For fuzzy boundaries based training, kernels of different sizes were used for the noisy and less noisy dataset. The size of these kernels is mentioned in all of the experiments in the fuzzy boundaries column. A large size kernel is used for the noisy dataset and a comparatively smaller size gaussian kernel is used for the less noisy dataset.

4.6 Sinus Segmentation

4.6.1 2.5X Resolution Experiments for Sinus Segmentation

The results of experiments conducted to study the effect of various data augmentations are given in table 4.1. By looking at experiments no. 3-6, we can conclude that a color jitter of 10% gives the best average precision for the Guy's hospital dataset. Additionally, it can be observed that a combination of random horizontal and veritcal flip along with color jitter performs better that any of these augmentations alone. Finally it can be concluded by comparing the results of exps 14,9 and 4, that the normalization along with the 3 augmentation (random horizontal and vertical flip, random rotation and color-jitter) works best. So this combination is referred to as optimum augmentations in the experiments in rest of the chapter.

UNet Experiments for Sinus-2.5X Segmentation

Exp. No.	Normalization	Horizontal and Vertical flip	Color-Jitter	Random Rotations	Test IOU	IOU Best Threshold	Average Precision
1	✖	✖	✖	✖	0.429	0.437	0.504
2	✔	✖	✖	✖	0.405	0.460	0.520
3	✖	✖	✔(5%)	✖	0.440	0.477	0.521
4	✖	✖	✔(10%)	✖	0.435	0.479	0.524
5	✖	✖	✔(15%)	✖	0.429	0.469	0.519
6	✖	✖	✔(20%)	✖	0.393	0.478	0.484
8	✖	✔	✔(5%)	✖	0.454	0.486	0.544
9	✖	✔	✔(10%)	✖	0.459	0.507	0.553
10	✖	✔	✔(15%)	✖	0.450	0.489	0.543
11	✖	✔	✔(25%)	✖	0.451	0.491	0.531
12	✖	✔	✔(5%)	✔	0.454	0.512	0.537
13	✖	✔	✔(10%)	✔	0.476	0.485	0.558
14	✔	✔	✔(10%)	✔	0.479	0.484	0.567

Table 4.1: Experiments for Data Augmentation Validation. UNet, KCL-Sinus-2.5x, Patchsize:504

Results of the experiments conducted for studying the effect of weakly supervised training techniques (fuzzy boundaries and bootstrapping) can be found in table 4.2. Comparing the results of experiments 2,3, 5 and 6, it can be stated that fuzzy boundaries based training improves results significantly and at 5X resolution, set of (31:11) kernels works best. Results of experiments no. 8-12 show that bootstrapping helps impove results and a beta value of 0.5 is optimum.

Table 4.3 includes results of experiments conducted for comparison of the multi-resolution model with a UNet. From this set of experiments it seems that the multi-resolution model doesn't perform as good as a UNet in most of the experiments. But observing the results of training on other resolutions such as 5X and 10X, it can be observed that as the starting resolution is increased, the multi-resolution network performs better than UNet. The reason for that could be that the multi-resolution model downsamples the input three times in each block, if the starting resolution itself is much smaller then most of the weights of multi-resolution model are dedicated at recognizing patters at a very low resolution, wherease if the starting resolution is large enough, then the multi-resolution network large enough resolution for all the three downsampled versions of the input.

So, from the results of experiments conducted on 2.5X resolution for sinus, it can be concluded that the set of data augmentations (random horizontal and vertical flips, color-jitter and random rotations) along with normalization gives significant improvement over not using any of these augmentations. Also it can be concluded that at 2.5X resolution a plain UNet works better or equivalent in some experiments as compared to a multi-resolution network architecture.

UNet Experiments for Sinus-2.5X Segmentation

Exp. No.	Optimum Augmentations	Center Crop Loss	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✖	✖	✖	✖	0.429	0.437	0.504
2	✔	✖	✖	✖	0.476	0.485	0.558
3	✔	✖	✔(51:21)	✖	0.484	0.494	0.563
4	✔	✖	✔(21:9)	✖	0.484	0.517	0.577
5	✔	✔	✔(21:9)	✖	0.464	0.503	0.556
6	✔	✖	✔(31:11)	✖	0.491	0.509	0.571
7 (TC)	✔	✖	✔(31:11)	✖	0.467	0.509	0.556
8	✔	✖	✔(31:11)	✔($\beta : 0.2$)	0.500	0.510	0.576
9	✔	✖	✔(31:11)	✔($\beta : 0.3$)	0.500	0.511	0.576
10	✔	✖	✔(31:11)	✔($\beta : 0.5$)	0.501	0.515	0.581
11	✔	✖	✔(31:11)	✔($\beta : 0.8$)	0.499	0.512	0.575
12 (RIS)	✔	✖	✔(31:11)	✔($\beta : 0.8$)	0.471	0.520	0.559

Table 4.2: UNet, KCL-Sinus-2.5x, Patchsize:504

RIS - Random initialization started (starting from 1st epoch itself).
TC- Transposed convolutions in UNet instead of binlinear interpoltion.
Fuzzy Boundaries (alpha:beta): alpha is the size of gaussian kernel for the more noisy dataset, and beta is the size of gaussian kernel for less noisy dataset.
Bootstrapping ($beta : x$): x=Value of beta for bootstrapping

MultiX Experiments for Sinus-2.5X Segmentation

Exp. No.	Optimum Augmentations	Normalization	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✔	✖	✖	✖	0.458	0.475	0.544
2	✔	✖	✔(31:11)	✖	0.452	0.478	0.554
3 (H6)	✔	✔	✔(21:11)	✖	0.446	0.484	0.551
4 (H6)	✔	✔	✔(31:11)	✖	0.459	0.473	0.561
6 (H4)	✔	✔	✔(31:11)	✖	0.466	0.495	0.575
7 (H4)	✔	✖	✔(31:11)	✖	0.453	0.478	0.556
9 (H4)	✔	✔	✔(31:11)	✔($\beta : 0.5$)	0.473	0.486	0.581

Table 4.3: MultiX, KCL-Sinus-2.5x, Patchsize:504

H6- Multi-resolution network with 624 kernels in the bottleneck block (27 million trainable paramers).
H4- Multi-resolution network with 420 kernels in the bottleneck block (23 Million trainable parameters).
Fuzzy Boundaries (alpha:beta): alpha is the size of gaussian kernel for the more noisy dataset, and beta is the size of gaussian kernel for less noisy dataset.
Bootstrapping ($beta : x$): x=Value of beta for bootstrapping

4.6.2 5X Experiments for Sinus Segmentation

UNet Experiments for Sinus-5X Segmentation

Exp. No.	Optimum Augmentations	Random Rotations	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✔	✖	✖	✖	0.463	0.519	0.654
2	✔(-norm)	✔	✖	✖	0.470	0.517	0.662
3	✔	✔	✖	✖	0.462	0.516	0.669
4	✔	✔	✔(61:21)	✖	0.473	0.509	0.672
5	✔	✔	✔(61:21)	✔($\beta : 0.5$)	0.475	0.508	0.676

Table 4.4: UNet, KCL-Sinus-5X, Patchsize:1008

Optimum Augmentations: Color Jitter + Random Horizontal and Vertical Flips + Normalization
-norm = Without Normalization

Multi-resolution Experiments for Sinus-5X Segmentation

Exp No.	HV Flips and Color Jitter	Random Rotations	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✔(+norm)	✖	✖	✖	0.463	0.517	0.661
2	✔	✖	✖	✖	0.453	0.509	0.656
3 (H5)	✔	✖	✔(61:21)	✖	0.504	0.501	0.684
4 (H4)	✔	✔	✔(61:21)	✖	0.481	0.504	0.692
5 (C64)	✔(+norm)	✔	✔(61:21)	✖	0.444	0.496	0.659
6	✔	✔	✔(61:21)	✔(RIS[1])	0.484	0.489	0.664
7	✔(+norm)	✔	✔(61:21)	✖	0.478	0.491	0.674
8 (H5)	✔	✔	✔(61:21)	✔($\beta : 0.5$)	0.473	0.518	0.691

Table 4.5: MultiX, KCL-Sinus-5X, Patchsize:1008

H5 - Multi-resolution model with 540 kernels in the bottleneck block
H4 - Multi-resolution model with 420 kernels in the bottleneck block
C64 - Center cropped loss calculation with margin of 64 pixels
Fuzzy Boundaries (alpha:beta)- alpha=size of gaussian kernel for noisy dataset, beta-size of gaussian kernel for less noisy dataset.

From results in table 4.4, it can be again observed that fuzzy boundaries and bootstrapping improve the results significantly at 5X resolution also. The best combination of gaussian kernels for noisy and less noisy dataset was found to be 61:21. Comparing the result of table 4.4 and table 4.5, it can be concluded that the multi-resolution model works better than the UNet model at 5X resolution for segmentation of sinus region. Again it can be observed that normalization of dataset results in improvement. Also, it can be observed that for multi-resolution model also at 5X resolution, fuzzy boundaries and bootstrapped training results in improvement of overall average precision. One more observation that can be made here is that starting bootstrapping right from the start gives inferior results as compared to starting bootstrapping after training fully and then bootstrapping.

[1] RIS:Start from beginning with first epoch

4.6.3 10X Resolution Experiments for Sinus Segmentation

Results of training experiments conducted for segmentation of sinus region using 10x resolution are shown in tables 4.6 and 4.7. From table 4.6, it can again be observed that fuzzy boundaries and boot-strapped training improves the segmentation results (Average IOU) significantly.

From table 4.7, it can observed that the best ratio of loss weights for noisy and less noisy datasets is 3:2. Also it can be observed that the best size of gaussian kernels for fuzzy boundaries training for noisy and less noisy datasets is 81:31. We can also conclude that as the resolution is increased, the margin with which the multi-resolution model outperforms the UNet model also increases. Overall it can be observed that with increase in resolution, the performance of multi-resolutin model increase wherease the performance of UNet seems to be saturating as the input resolution is increased.

UNet Experiments for Sinus-10X Segmentation

Experiment No.	Optimum Augmentations	Fuzzy Boundaries	Boot-strapping	Test IOU	IOU Best Threshold	Average Precision
1	✔	✖	✖	0.446	0.480	0.644
2	✔	✔	✖	0.457	0.491	0.653
3	✔	✔	✔($\beta : 0.5$)	0.451	0.496	0.661

Table 4.6: UNet, KCL-Sinus-10x, Patchsize:1024

MultiX Experiments for Sinus 10X Segmentation

Exp No.	Optimum Augmentations	Center Crop Loss	Fuzzy Boundaries	Boot-strapping	Test IOU	IOU Best Threshold	Average Precision
1 N(2:1.2)	✔	✔(64)	✔(71:11)	✔($\beta : 0.5$)	0.440	0.481	0.673
2 N(2:1.2)	✔	✖	✖	✖	0.456	0.463	0.657
3 N(1:1)	✔	✖	✔(31:11)	✖	0.457	0.494	0.669
4 N(1:1)	✔	✔(64)	✔(31:11)	✖	0.439	0.442	0.651
6 N(1:1)	✔	✔(64)	✔(81:31)	✖	0.466	0.453	0.667
7 N(3:2)	✔	✖	✔(81:31)	✖	0.482	0.496	0.696
8 N(3:2)	✔	✖	✔(81:31)	✔($\beta : 0.5$)	0.486	0.501	0.702

Table 4.7: UNet, KCL-Sinus-10x, Patchsize:1024

4.7 Germinal Center Segmentation

Results of experiments conducted for segmentation of germinal center regions are present in tables 4.8 and 4.9. It can be seen that the best threshold IOU and Average precision for segmentation of germinal center regions is much higher as compared to the sinus region. Here also, it can be observed that the multi-resolution model outperforms the UNet model with a significant margin of average precision. The observation that fuzzy boundaries and bootstrapped training helps improve segmentation results can again be reinforced from the results of tables 4.8 and 4.9.

UNet Experiments for GC 10X Segmentation

Exp No.	Optimum Augmentations	Normalization	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✔	✔	✖	✖	0.498	0.706	0.795
2	✔	✔	✔	✔	0.512	0.712	0.812

Table 4.8: UNet, KCL-GC-10x, Patchsize:1024

Multix Experiments for GC 10X Segmentation

Exp No.	Optimum Augmentations	Center Crop Loss	Fuzzy Boundaries	Bootstrapping	Test IOU	IOU Best Threshold	Average Precision
1	✔	✖	✖	✖	0.545	0.765	0.845
1	✔	✔	✖	✖	0.552	0.771	0.853
1	✔	✖	✔	✖	0.558	0.766	0.860
1	✔	✖	✔	✔($\beta : 0.5$)	0.560	0.760	0.865

Table 4.9: Multi-Resolution Network, KCL-Sinus-10x, Patchsize:1024

4.8 Other Experiments

In addition to the experiments discussed in the above sections, some experiments were also conducted with the multi-resolution network given in [3] which is trained by creating a dataset explicitly at 3 different resolutions and training 5 neural networks with 3 being UNets, which all are trained simultaneously in an end-to-end fashion. Experiments with this network required lots of GPU memory (Atleast 3 GPUs with 12 GB RAM each) and it was really difficult to continue witholding that much resources by devoiding others of it, such experiments were concluded with obtained results not better than a UNet. In addition to that, I also tried the segmentation with another multiresolution network, the architecture of which was hypothesised along with the multi-resolution architecture which is extensively used experiments in this work. In that architecture, we explicitly downsample the input patch as many times a there are downsampling blocks in the UNet and each block, in addition to the output tensor of its previous block, is also fed with the downsampled image tensor. At the decoder side, the segmentation mask is similarly downsampled and prediction at each upsampling block is used to compute a separate loss for each resolution, by giving a weight to each resolution. A higher weight is given to the lower resolution predictions as we want the overall shape of the segmented objects to be correct first and then focus on high resolution boundaries. The results obtained by such a multiresolution network also were not quite satisfactory and best of the results were only as good as a plain UNet. The experiments based on this multiresolution network were also concluded after these observations.

4.8.1 Center Cropped Loss Calculation

One more strategy of loss calculation, to reduce the effect of padding which is done in all of the convolutional layers of the CNN architectures used for segmentation, which we call the center crop loss calculation was also experimented with. The idea being that theoretically, after predicting the segmentation mask for a patch, if only some central area of the predicted mask is used for loss calculation, then the padded zeros during the forward pass wouldn't effect the training in any way. The reason for padding zeros not effecting is that the receptive field is finite and border zeros can only effect the pixels in predicted mask till only a certain margin. So, training with loss computed using this strategy should result in better segmentation virtually at a whole-slide level. But, practically it was observed that center cropped loss calculation resulted in inferior segmentation performance as compared to a plain loss computation strategy. The reason for this outcome couldn't be investigated fully.

4.8.2 Experiments on Camelyon Dataset for segmentation of Tumours

A number of experiments were also conducted for segmentation of tumour and non-tumour regions in H&E whole-slide images of lymph nodes in Camelyon16 dataset. Figures 4.31 and 4.32 show the test results of UNet for segmentation on 2.5X and 5X resolution respectively.

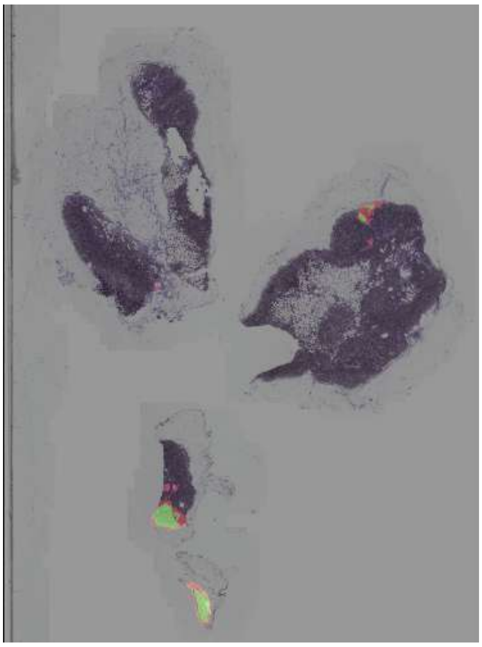


Figure 4.31: Tumour Segmentation in Camelyon Dataset Sample Result - 1.

Although the segmentation framework work effectively to a good extent for segmentation of tumour regions in Camelyon dataset images, the overall IOU score and average precision values are very less because more false positive predictions are observed as compared to segmentation of sinus and GC regions in Guy's hospital dataset. Experimental work for develpment for segmentation framework used so far for effective segmentation of tumours in Camelyon dataset is still ongoing.

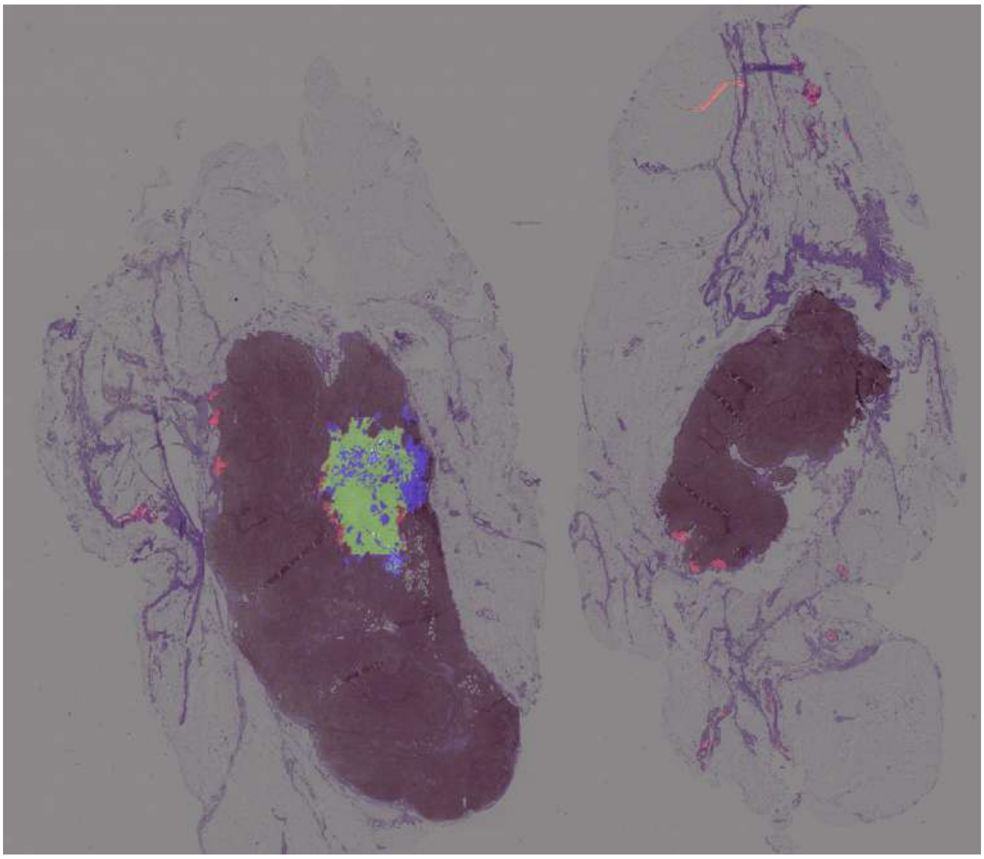


Figure 4.32: Tumour Segmentation in Camelyon Dataset Sample Result- 2.

One more apparent reason for more false positives in segmentation results could be possible absence of annotations for small tumour regions in the dataset itself as it is observed in most of the experiments that a large part of false positives belong to small regions which could actually be micro tumours.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Following is a list of conclusions that can be made based on the results obtained so far.

- A CNN capable of processing visual information from multiple resolutions of the whole slide images can be able to obtain better segmentation results than a single resolution network such a plain UNet.

- It can also be concluded that training the multi-resolution network on higher resolution is giving better results, as the the best results for Germinal Centre prediction are obtained at 10x resolution, 2nd best at 5x and worst so far at 2.5x resolution.

- As the input resolution is increased, the margin with which the multi-resolution network outperforms a UNet model also increases as can be observed from results included in section 4.5.

- Random rotations, horizontal & vertical flips and color jitter data augmentation are quite effective in improving the segmentation results of lymph node H&E images.

- Sinus regions are often incorrectly marked and are harder to predict than Germinal Centers, so require a more robust training mechanism.

- Feeding the entire region of interest into the fully convolutional segmentation network and then obtaining the prediction gives a remarkable improvement in results as compared to the patch prediction and stitching back approach. So whenever the hardware permits, it is better to predict by inputting whole image rather than predicting for patches and then stitching to avoid patching effect.

- Fuzzy boundaries based training gives significant improvement in average precision for segmentation. Bootstrapped training also adds up to this improvement as can be observed in results of most of the experiments.

5.2 Future Work

- As it was observed in most of the experiments that increasing the input patch size improves results, multi-GPU training with larger patch sizes and deeper multi-resolution model may be able to learn to identify larger and more complex patters (larger receptive field), and may improve overall results.

- Boundary correction techniques for noisy annotations, similar to the techniques used in [15], may help improve the results as boundaries for most of the regions in whole-slide images are imprecisely marked.

- Noisy segmentation approach as given in [16], which utilizes a small dataset of very clean annotations and a large corpus of regular, less carefully marked annotations, for learning the actual segmentation boundaries may prove to be quite effective if a small dataset of very precise annotations is created for a few lymph node whole-slide images.

- As the annotations for multiple classes are available in the Guy's hospital dataset, an approach which helps the network learning to segment one region by utilizing the annotations of another region, as can be found in [17], may help improve the segmentation results.

- The active boundary loss [18] or a similar approach, which utilizes KL-divergence based loss for automatically moving boundaries towards the most probable boundary pixels may prove to be quite effective for segmentation of regions in H&E images.

Chapter 6

Introduction: Graph Representation Learning with Keypoints in Images

6.1 Overview

This project aims to investigate if a good representation of images can be learnt using graph neural networks. For that purpose, the first step is to have a method to identify a good set of key locations in the image and then extract features from these locations and their neighbourhood. This way, these feature may be treated as features of nodes of a graph. As the graphs can be used to utilize the relationship between different objects which correspond to its nodes, it may be possible to learn a representaion vector for each image by processing with a graph neural network which is more suitable for classification and image compression purposes. After trying out a number of approaches such as concatenating the features at the output of graph neural network and then flatten before passing through layers of a classifier neural network, the procedure shown in image 6.1 was decide to be good training procedure for the experiments of this project. Multiple ways to detect the keypoints were experiment with and Chapter **??** includes their details and obtained results.

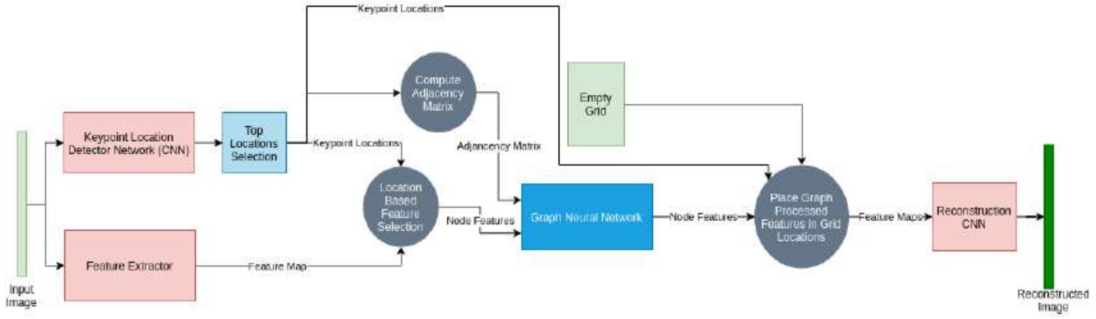


Figure 6.1: Procedure to learn reconstruction of images using keypoints and graph representations.

Figure 6.1 shows the procedure used for training in the experiments conducted as part of this project. Firstly, two parallel network take the image as input for feature calculation and keypoint location estimation. They keypoint location detector network may be implemented in a number of ways and depending on that the feature extractor network may also assume different archituecture based on the keypoint detector network. The keypoint detector network and and the key location calculation part was implemented in three different ways.

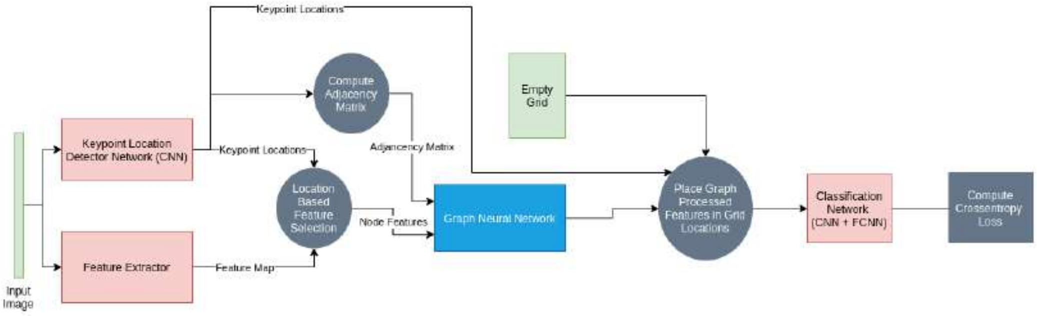

Keypoint Locations
Compute Adjacency Matrix
Empty Grid
Keypoint Location Detector Network (CNN)
Keypoint Locations
Adjancency Matrix
Location Based Feature Selection
Node Features
Graph Neural Network
Place Graph Processed Features in Grid Locations
Classification Network (CNN + FCNN)
Compute Crossentropy Loss
Feature Extractor
Feature Map
Input Image

Figure 6.2: Procedure to learn classification of images using keypoints and graph representations.

The experiments based on the different approaches used for keypoint location estimation and the results so obtained are presented in chapter **??**. After the keypoint location are detected by the combination of two leftmost networks shown in fig. 6.1, the next step is to present the features corresponding to these locations as nodes of a graph. So the feature at the output of feature extractor network are selected by selection based on N key locations (N is a hyperparameter) and become the node features for the graph neural network. An adjacency matrix is also calculated using the Euclidean distance between each pair of points. After the node features are processed by the graph neural network, they are put back into the locations (as predicted by keypoint detection network) of an empty tensor. This tensor is then further processed by a CNN to either reconstruct the image or predict the class label. The results obtained by using three different strategies for keypoint location detection are presented in the next chapter.

Chapter 7

CNN to Graph and Graph to CNN Interconnects

A Graph Neural Network (GNN) requires its inputs to be in the form of a graph. To use a GNN for inference on data which is not inherently structured as graph, a transformation of data structure may be required. More specifically, to process data such as images using a GNN, it may be required to transform the image or image feature maps (After some processing with a CNN) to a graph, to make it a valid GNN input. Similarly, to process the graph structured data using a CNN, a graph to a meaningful 3D tensor transformation may be required while preserving the possibility of gradient flow for backpropagation. In this work, two such interconnects i.e, a GNN to CNN and a CNN to GNN interconnects are presented.

In addition to the node features, graphs also have the connections information, whereas the CNN feature maps are represented as 3 dimensional tensors with the inherent spatial arrangement. An interconnect between a CNN and a GNN has to take care of transformation of the spatial arrangement to edge weights and vice versa in addition to assignment of features from nodes to spatial locations and vice versa.

In case of an interconnect where the entire 3D feature maps are required to be converted to a graph, every feature vector at a location (i, j) can become the feature vector of a node in the graph, i.e a tensor of size $N \times M \times L$ will produce a graph with $N \times M$ nodes with each node having a feature vector of length L. The inverse of distance between each pair of location (i, j) in the feature map or any other measure of connectivity can become the edge weight of connection between corresponding nodes. This way, for feature maps of size $N \times M \times L$, size of adjacency matrix would be $(NM) \times (NM)$. Similarly, an inverse transformation can be applied if location of each node as in the original 3D tensor of feature maps is tracked by rearranging the nodes features in the form of a 3D tensor using a suitable way for spatial assignment.

In case of an interconnect where only a handful of locations from the 3D tensor of feature maps are required for constructing a graph, for example in case processing features only from key-points in images, a slightly different approach than discussed in above paragraph is required. Two such interconnects for key-points based processing using a hybrid model of GNNs and CNNs are discussed in Sections 7.1 and 7.2.

7.1 CNN to Graph Interconnect

To transform the output feature maps of a convolutional layer, at any stage in CNN, into a graph, two pieces of information are required:

- Node features (The features from CNN output, which will become the features of Nodes of the graph).
- Connections information between the nodes.

In case of a key-points processing based CNN-to-graph interconnect, as shown in Fig. 7.1, the feature vectors belonging to the key-point locations become the features of nodes and the inverse of distances between every pair of these key-points becomes the strength of interconnection between the corresponding pair of nodes in the graph. This way if there are N key-points then the adjacency matrix will have dimensions of $N \times N$.

The implementation of this type of interconnect is available at:
`https://github.com/amitlohan/nn_interconnects`

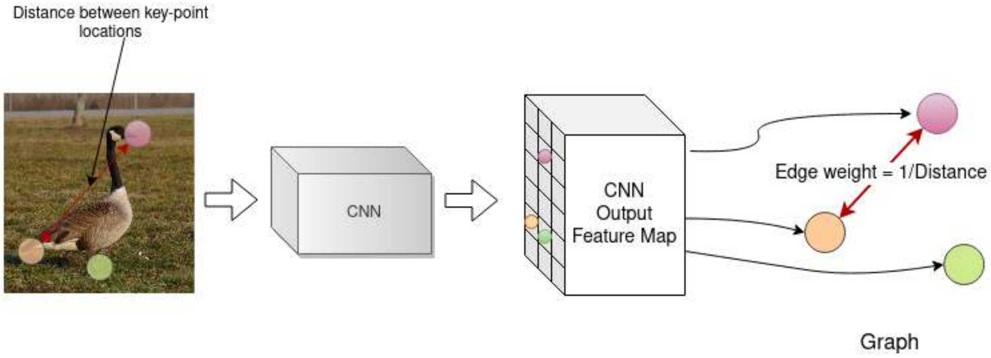


Figure 7.1: CNN to Graph Interconnect.

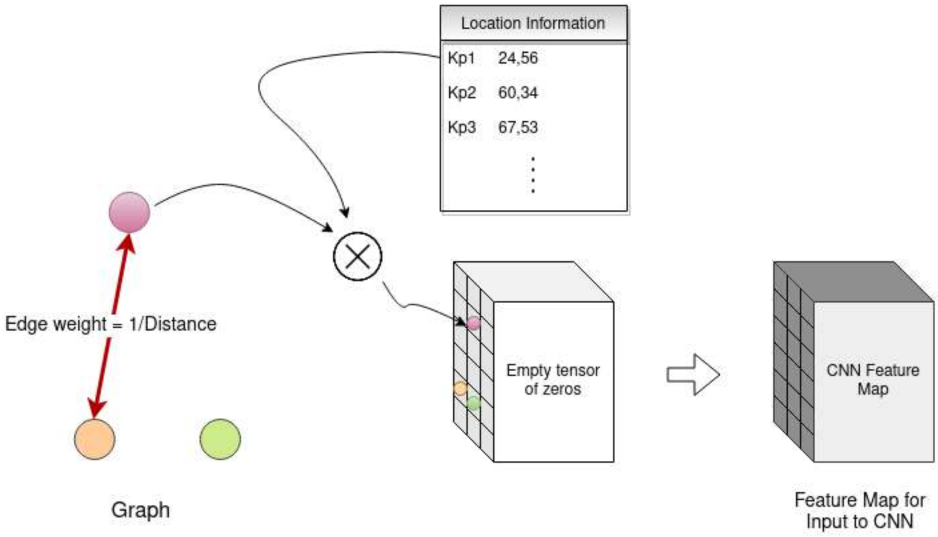


Figure 7.2: Graph to CNN Interconnect

7.2 Graph to CNN Interconnect

For transforming a graph into a tensor of CNN feature maps, we require node feature vectors and a way to arrange them spatially in a grid. To arrange the features correctly into a 2D grid, the original key-point locations are required. As shown in Fig. 7.2, first of all we take a 3D tensor of zeros with dimensions $N \times H \times W$, where N is the size of feature vectors of the nodes of graph, and W and H are respectively the width and height of the CNN feature maps to be obtained, then the features are assigned to the 2D locations as given by original locations of key-points which was used for conversion of feature maps to graph. This way a valid CNN features map is obtained.

The implementation of this type of interconnect is available at: `https://github.com/amitlohan/nn_interconnects`

Chapter 8

Key-point Detection Methods and Results

8.1 Attention Network as Location Selector

One simple way of detecting keypoint locations is to predict an attention score for each pixel location and then select top N points according to the predicted attention values. This way, using a classification or reconstruction loss at the output of post processing CNN, the attention network should learn to identify the most important points in the image from classification or reconstruction point of view. The results obtained by using such an approach are shown in fig. 8.2.

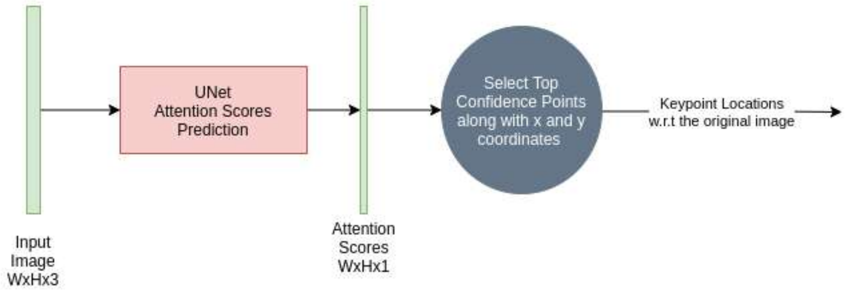


Figure 8.1: Procedure to identify keypoints with attention network based approach.

It can be seen in the images in fig. 8.2 that the detected keypoints using the attention network based method alway lie in bunches or groups. Two or more keypoints lying in the close proximity of each other are undesirable as the features corresponding to these points will be quite similar but will take different nodes in the graph network. So most of the features taken from such locations will be redundant and will wast lots of computation. This has been observed to be a serious problem with the attention network based keypoint detector network. To overcome this problem two more keypoint location detection strategies were developed and are discussed in section 8.2.

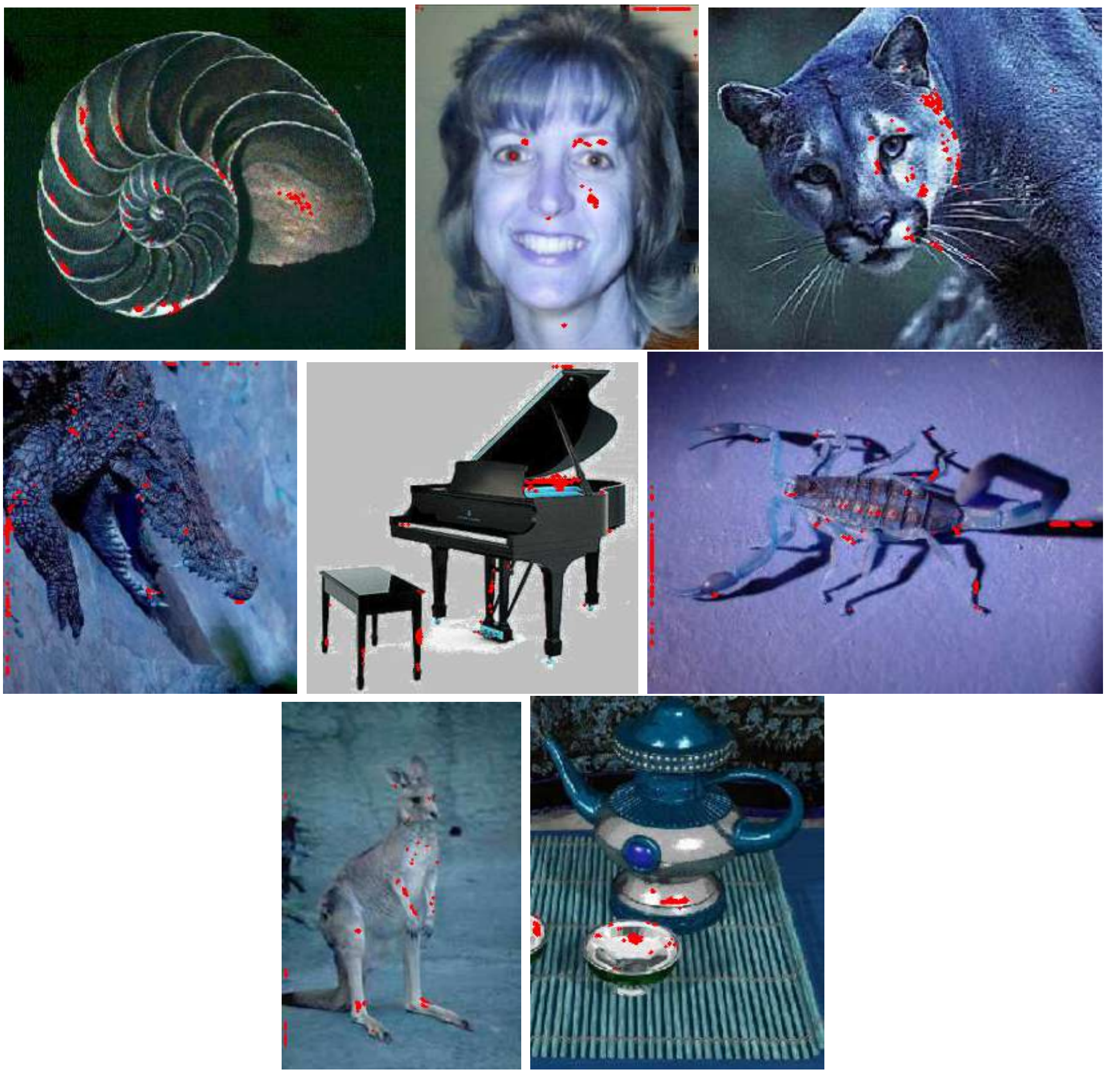


Figure 8.2: Detected key points (red) with attention network based approach

8.2 Keypoints Coordinates Prediction with a Neural Network

One possible way to overcome the problem posed by the attention network based method in section 8.1 is to have a network directly predict the coordinate values of keypoint locations and pre-train it with a dataset of images by computing ground truth at the training time. The ground truth locations for pretraining can be obtained by detecting key locations with SIFT or ORB keypoint detector algorithms. This way the keypoint coordinate prediction network will learn to identify the key locations to some extent before it is used in the main training loop.

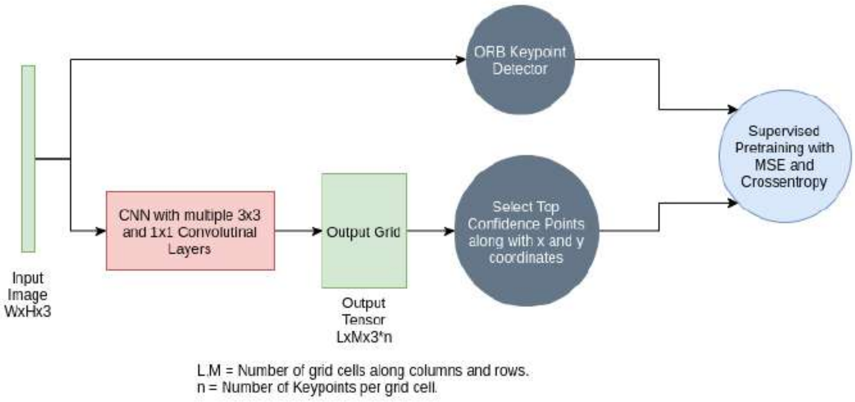

Input
Image
WxHx3
CNN with multiple 3x3 and 1x1 Convolutinal Layers
Output Grid
Output
Tensor
LxMx3*n
ORB Keypoint Detector
Select Top Confidence Points along with x and y coordinates
Supervised Pretraining with MSE and Crossentropy
L,M = Number of grid cells along columns and rows.
n = Number of Keypoints per grid cell.

Figure 8.3: Procedure to identify keypoints with coordinates prediction network based approach.

Figure 8.3 shows such a setup to predict coordinate locations. The output of predictor network should be a LxMx(3h) vector where L¡=W and M¡=H and L and M are number of cells along rows and columns. Each cell is responsible for detecting keypoints in a mxm square patch. The values of m in the experiments included in this section has been 8 and 16. Each cell can detect a maximum of h keypoints in the square patch it is responsible for. Each 3h numbers along the axis 2 of output tensor predict three things: x coordinate within the cell, y coordinate within the cell and confidence of prediction of each keypoint. In the training loop, confidences are multiplied with the selected feature from output of feature extractor network and in addition to providing confidence values for selection, they also act as attention values.

The results obtained using this approach are shown in figure 8.2. As can be seen, the network is able to track the keypoints on object quite well. The only problem that was observed with this approach was that after training the network for enough number of epochs, the keypoints are predicted very close to or on the boundaries of cells. The probable reason for that sounds to be the use of sigmoid to normalize the output in the range of 0 to 1. As the gradients at extreme points of sigmoid are quite low and very high around 0, the keypoints find it very unlikey around the center of the image. For classification task, this network was able to train quite quickly to 99.9% train accuracy but did not give better than 60% accuracy for 10 classes of imagenet. So far, I haven't been able to solve the problem of keypoints moving close to boundaries.

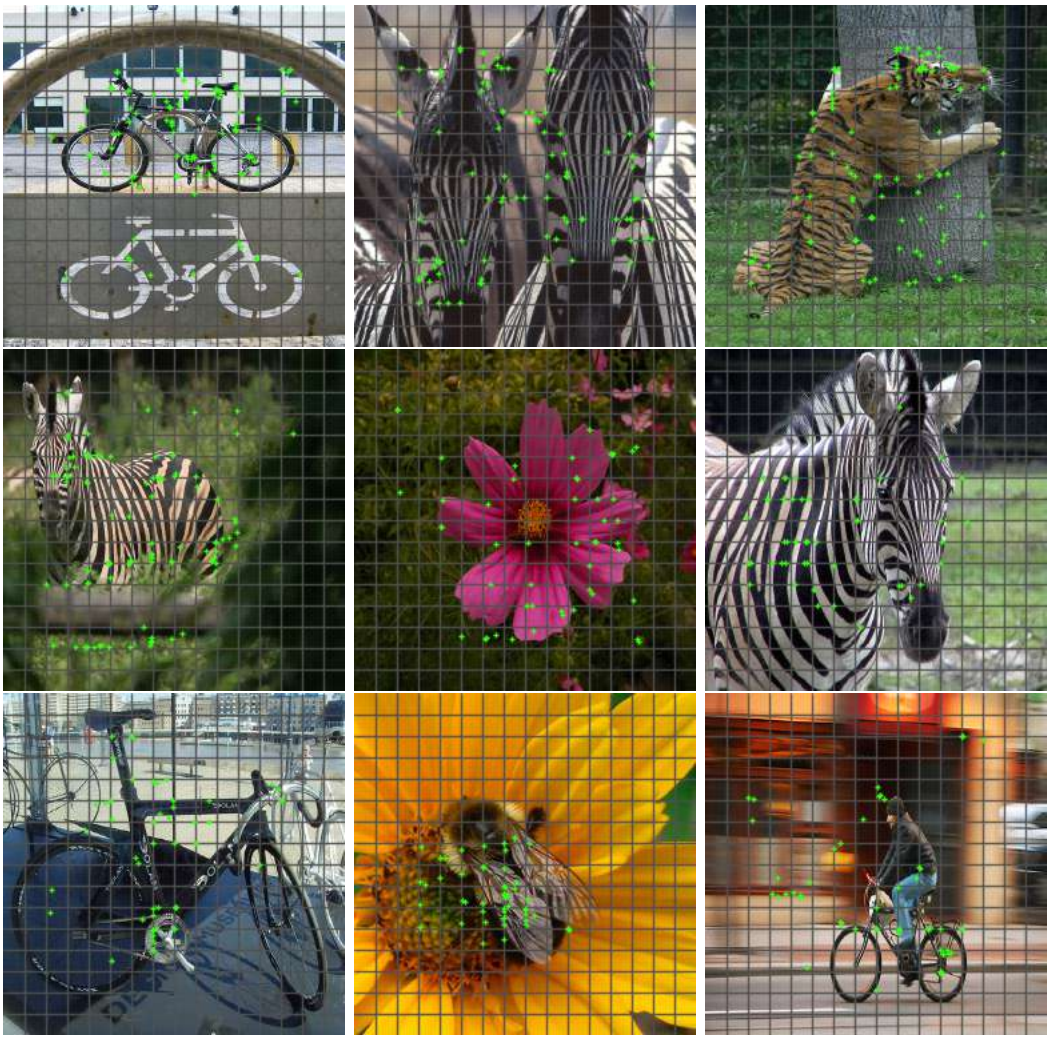


Figure 8.4: Keypoint locations identified by a coordinate prediction (regression) network based on the concept of [5].

8.3 Large Maxpool as Non-max Suppression

Another way that I worked out for non-max suppression in the experiment presented in the section 8.1 is to use the procedure shown in figure 8.5. A large maxpool of 8x8 or 16x16 is used to avoid the possibility of detecting points in close clusters and the locations of maxpooled in points are also saved (relative to the original predicted mask). Top N locations are then selected from the maxpooled feature map based on the attention scores and the respective locations are also selected from the set of locations previously saved. This way, we get a set of top keypoints with non-max suppression and spatial sparsity. The results obtained using the classification loss in this method are presented in figure 8.6.

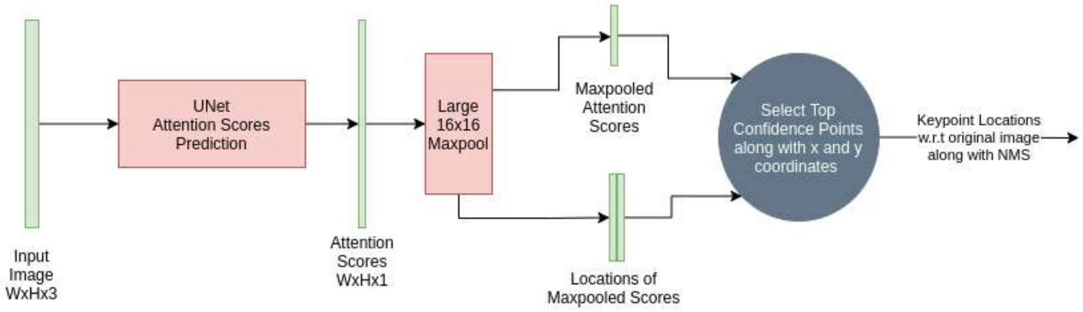


Figure 8.5: Procedure to detect keypoints using maxpool as non-max suppression.

In addition to the classification approach, training experiments were also conducted with the reconstruction objective with imagenet and a faces dataset. The reconstruction did not work well with the imagenet, the possible reason perhaps being high variablity of images. The results obtained with faces dataset are presented in figure 9.1. Average mse of 0.06 for images normalised in 0 to 1 was obtained for the reconstruction.

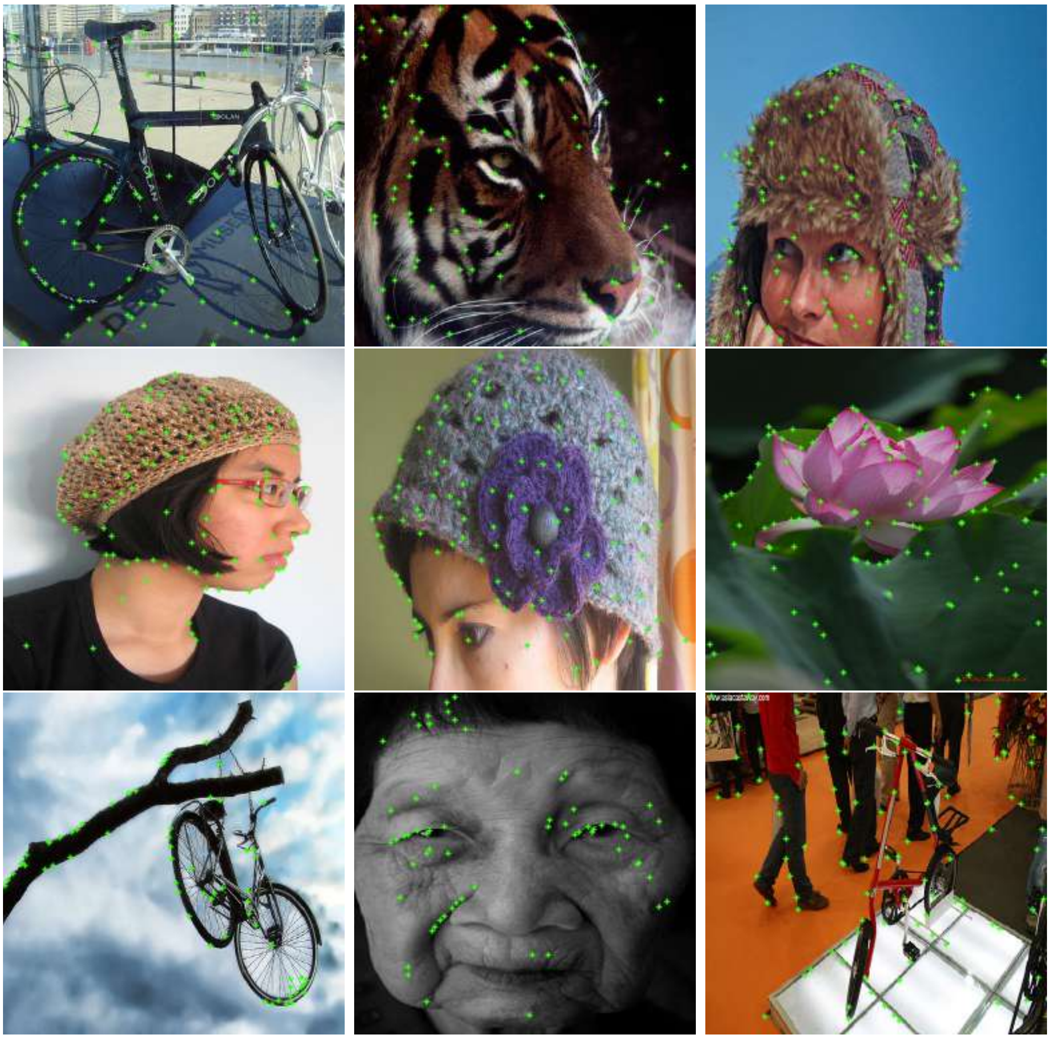


Figure 8.6: Keypoints detected by the model in section 8.3.

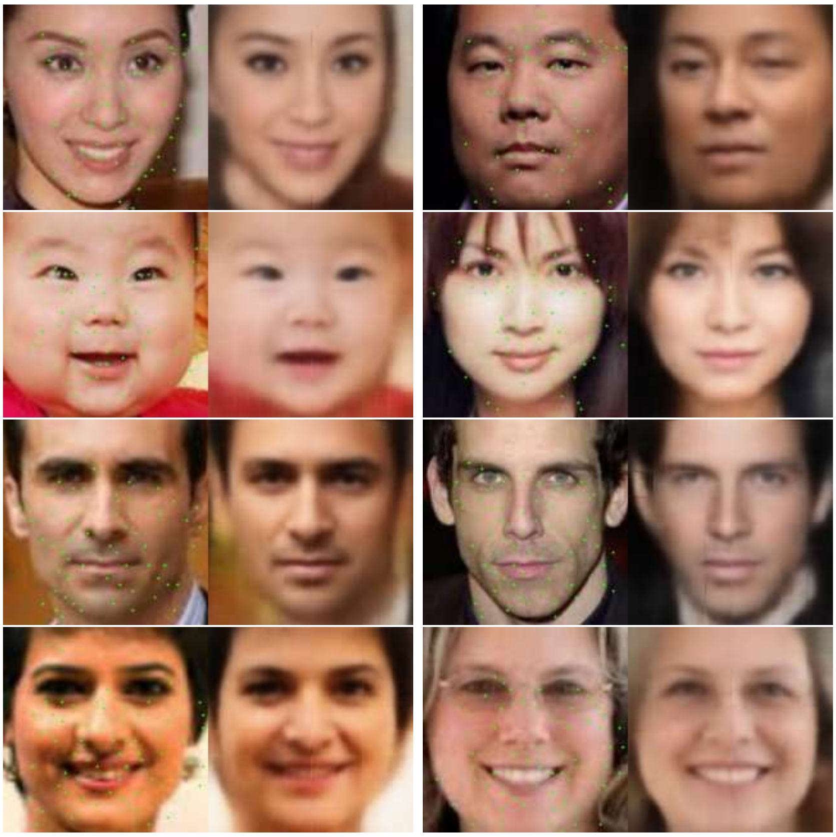


Figure 8.7: Image reconstruction results and keypoints detected by the model in section 8.3.

Chapter 9

Classification and Reconstruction Experiments and Results

The experiments and results discussed in chapter 8 were conducted with keypoint detection network and feature extractor network being two separate networks. Hence, the number of parameters were quite large and overfitting was observed in those experiments when classification and reconstruction experiments were conducted. To reduce the number of parameters, the two networks were combined, with a single network producing both the outputs (attention scores and features). A single extra layer in adding to the network output is used for attention scores. Remaining arrangment of networks and training flowgraph is same as shown in figures 6.1 and 6.2.

For all the experiments discussed in this chapter, a dataset of 10 classes taken from Imagenet was used. The dataset contains a total of 51916 images, with each class having more or less, same number of images.

In most of the experiments, a simple, fully connected graph neural network with adjacency matrix calculated from Euclidean distances between all the pairs of keypoints was used. In some of the experiments however, to iteratively reduce the number of nodes for obtaining a graph level embedding, a special graph neural network architecture called Diffpool, as discussed in [10] was used. In Diffpool, after each layer of neural network, number of nodes is reduced by a certain factor and after final layer only one node, whose embedding represents the whole input graph is obtained.

9.1 Classification Experiments

Exp. No.	GNN Architecture	No. of Graph Layers	No. of Key Points	No of Classes	Keypoint Score Activation	No. of trainable parameters	Test Accuracy
1	CNN Only		64	10	Linear	3.29M	51.60
2	GCN	4	64	10	Linear	3.31M	78.15
3	GCN	5	64	10	Linear	3.31M	74.35
4	Diffpool	5	64	10	Sigmoid	5.11M	98.1
5	GCN	5	64	10	Sigmoid	6.48M	98.4

Table 9.1: Imagenet Classification Experiments

Table 9.1, shows the results of classification experiments. It can be observed that when a CNN only architecture, with graph neural network being absent is used, a 51.60 % accuracy is obtained, which is much less as compared to the flow-graphs where a graph neural network is present in the loop. In experiment no. 2, when a graph neural network is inserted, in addition to the classification CNN, it can be seen that with the increase of only a small number of parameters, the accuracy increases quite significantly.

While ranking the importance of key-points, two types of activations can be used for calculating the final score, a linear activation and a sigmoid activation. In table 9.1 Experiments 3 and 5 show the difference of accuracies obtained when a sigmoid is used instead of a linear activation. It was observed that using the Diffpool architecture gives almost as good accuracy as the plain GCN network architecture. Additionally, diffpool architecture trains faster than the fully connected graph neural network architecture.

9.2 Reconstruction Experiments

For reconstruction, an upsampling convolutional network was used to reconstruct the image from tensor filled with node features. The loss function used for reconstruction was MSE and a modified MSE (termed as conf. MSE). For calculation of conf. MSE loss, the square error loss at each location is also multi with the attention score (output of keypoint detection network) or the confidence of that point/pixel being a key-point.

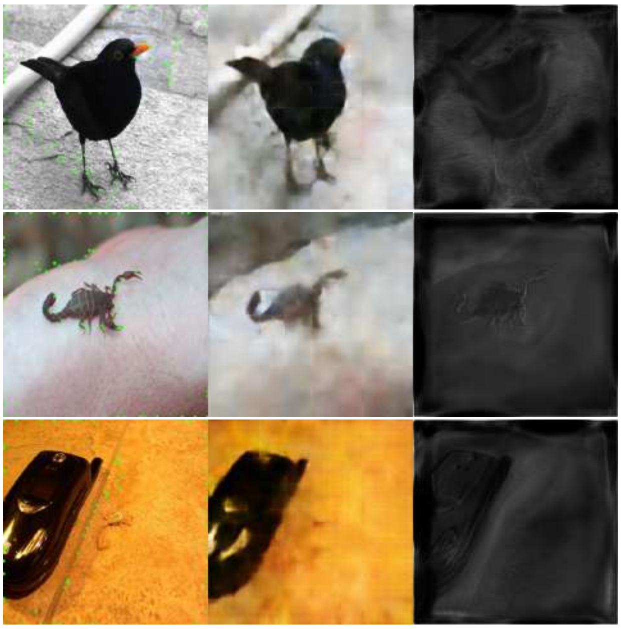


Figure 9.1: Image reconstruction results and keypoints detected for experiments of table 9.2.

This way, the potential background regions are not given more importance for reconstruction and only the object regions, where keypoints are meant to be detected are given more importance for training. Results of reconstruction experiments are given in table 9.2.

Exp. No.	Graph Architecture	No. of Graph Layers	No. of Key Points	Keypoint Score Activation	No. of trainable parameters	Loss Function	Loss Value
1	GCN	5	48	Linear	14.3M	MSE	0.01720
2	GCN	9	64	Linear	14.42	MSE	0.03128
3	GCN	5	64	Sigmoid	17.48	MSE	0.00928
4	Diffpool	5	64	Sigmoid	5.11M	Conf. MSE	0.11712

Table 9.2: Imagenet Reconstruction Experiments

9.3 Dual Task Experiments

As can be seen in figure 9.1, although the training with a reconstruction task successfully learns a representation from which the images can visually be reconstructed, the locations of key-points doesn't seem much meaningful. The key-points seem to be scattered around. Whereas it can be observed from figure 8.6 that with a classification task, the location of detected key-points is quite meaningful (on objects and on edges and corners).

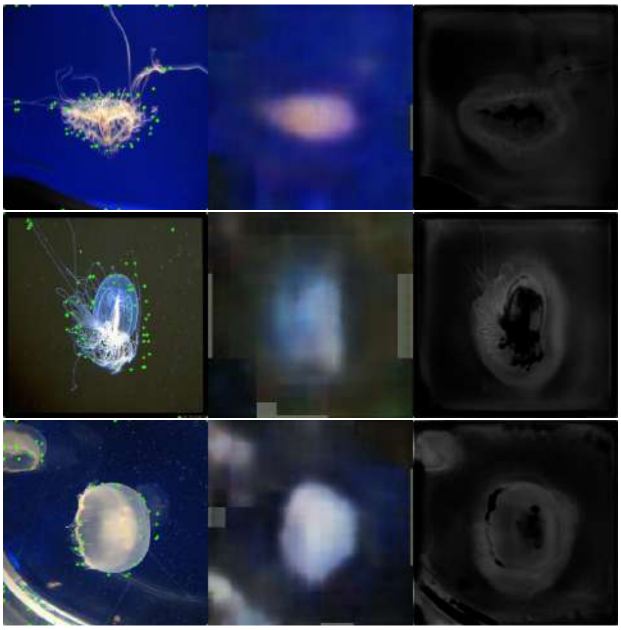


Figure 9.2: Dual task experiment results and keypoints detected for experiments of table. 9.3.

If the locations of detected key-points makes sense and lies on object parts, corners and edges, naturally, a better representation can be expected to be learnt. Keeping that in mind, a dual task, where the final loss is a combination of two losses: classification loss and reconstruction loss with reconstruction and classification networks in parallel, may be able to detect good key-point locations along with reconstruction. Results of such dual task experiments are given in table 9.3.

It can be observed from experiments in table 9.3, that with a dual task the key-points are getting detected at potential key-point locations but the reconstruction has suffered to certain extent. This kind of trade off between reconstruction and proper

Exp No.	Graph Architecture	No. of Graph Layers	No. of Key Points	No. of trainable parameters	Loss Weights CX:RX:ML	Grid Size	Reconstruction Loss	Accuracy
1	GCN	5	32	13.53	10:1:1	16	-	87.80
2	GCN	5	32	13.53	1:1:1	16	-	80.04

Table 9.3: Imagenet Dual Task Experiments

key-point location detection was observed in dual task experiments which is controlled by the weights chosen for classification and reconstruction loss.

Chapter 10

Conclusion and Future Work

It can be concluded from the experiments in section 8.1 that a vanilla attention based network is not a good solution to the problem of keypoints detection as it does not give a spatial sparsity of keypoints and introduces lots of redundance of data in the learnt representation, which is a contradiction with the very objective of this project. Observing the results in section 8.2, we can say that the coordinate prediction network is quite successful in locating the key-points on primary objects in the image and is able to effectively discard the points on background. The only problem with this approach seems to be its inability to track points in the central areas of grid cells, possibly because of the use of sigmoid at the output. As sigmoid seems to be the only available option of a activation function which is both differentiable as well as, is capable of mapping a real number to [0,1] range. A solution to this problem may help detect good points and might improve the classification and reconstruction results significantly.

The results of experiment in section 8.3 may be said to support the conclusion that a max-pool based non-max suppression is working quite effectively and key-points are being detected at appropriate locations. A network trained on features extracted from these locations should be able to give a good classification accuracy. Now the test accuracy obtained on 10 classes of Imagenet using the representations learnt with this approach was not more than 65% although the model was able to smoothly train to get close to 100% accuracy on the test dataset.

In the second stage of experiments in this project, after the keypoint-detection and feature extractor networks were combined, the overfitting reduced significantly as can be seen from results included in table 9.1. Also it can be observed that using a sigmoid activation instead of a linear activation to obtain final key-point scores given much better classification and reconstruction results as can be observed from tables 9.1 and 9.2.

Overall it can be concluded that the representation learning framework employed for reconstruction and classification tasks works well and is able to give 98.5% classification accuracy for a classification task and good enough representation for a reconstruction task for a dataset constructed from 10 classes of Imagenet. The method, however, failed to obtain good classification accuracy for 100 classes of Imagenet. Significant overfitting was observed for a dataset constructed with 100 classes as the model failed to obtain more than 35% accuracy for 100 class dataset. Further investigation may be done to find a solution to the problem of overfitting when the number of classes is increased from 10 to 100. Using a deeper model for keypoint detection, or using more number of key-points for a 100 classes dataset may result in improvement of overall classification accuracy as 48 or 64 key-points may not be able to capture enough distinguishable information for classification or reconstruction when training is done on a dataset of 100 classes.

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation* , CVPR 2015, https://arxiv.org/abs/1505.04597

[2] Grigoriadis et. al, *Histological scoring of immune and stromal features in breast and axillary lymph nodes is prognostic for distant metastasis in lymph node-positive breast cancers: Prognostic value of histological immune and stromal features*, The Journal of Pathology: Clinical Research, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5783956/.

[3] Hiroki Tokunaga, A. Y. R. B., Yuki Teramoto, 2019, *Adaptive weighting multi-field-of-view CNN for semantic segmentation in pathology*

[4] Yin Li, Abhinav Gupta, *Beyond Grids: Learning Graph Representations for Visual Recognition*, NeurIPS 2018, https://papers.nips.cc/paper/2018/hash/4efb80f630ccecb2d3b9b2087b0f9c89-Abstract.html

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, CVPR 2016, https://openaccess.thecvf.com/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[6] Shruti Jadon, *A survey of loss functions for semantic segmentation*, EESS, 2020, https://arxiv.org/abs/2006.14822

[7] Fisher Yu, Vladlen Koltun, *Multi-scale context aggregation by dilated convolutions, ICLR 2016, https://arxiv.org/abs/1511.07122*

[8] Long, J., Shelhamer, E., and Darrell, T., 2015, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition, https://arxiv.org/abs/1511.07122

[9] Khened, M., Kori, A., Rajkumar, H. et al., *A generalized deep learning framework for whole-slide image segmentation and analysis, Sci Rep 11, 11579 (2021), https://www.nature.com/articles/s41598-021-90444-8.pdf*

[10] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, Jure Leskovec, Hierarchical Graph Representation Learning with Differentiable Pooling, NIPS 2018, https://proceedings.neurips.cc/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7-Abstract.html

[11] Huang, X, He, H, Wei, P, Zhang, C, Zhang, J, Chen, J., *Tumor tissue segmentation for histopathological images, In Proceedings of the ACM Multimedia Asia 2019.*

[12] Campanella, G, Hanna, MG, Geneslaw, L, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images, Nature Med. 2019

[13] Halicek, M, Shahedi, M, Little, JV, et al., *Head and neck cancer detection in digitized whole-slide histology using convolutional neural networks, Sci Rep. 2019.*

[14] Wetteland R, Engan K, Eftestøl T, Kvikstad V, Janssen EAM., A Multiscale Approach for Whole-Slide Image Segmentation of five Tissue Classes in Urothelial Carcinoma Slides, Technology in Cancer Research & Treatment. January 2020.

[15] David Acuna, Amlan Kar, Sanja Fidler, *Devil is in the Edges: Learning Semantic Boundaries from Noisy Annotations, in CVPR 2019,* *https://openaccess.thecvf.com/content_CVPR_2019/papers/Acuna_Devil_Is_in_the_Edges_Learning_Semantic_Boundaries_From_Noisy_CVPR_2019_paper.pdf*

[16] Zahra Mirikharaji, Yiqi Yan, Ghassan Hamarneh, Learning to Segment Skin Lesions from Noisy Annotations, In MICCAI Workshop on Domain Adaptation and Representation Transfer, 2019.

[17] Olivier Petit, Nicolas Thome, Arnaud Charnoz, Alexandre Hostettler and Luc Soler, *Handling Missing Annotations for Semantic Segmentation with Deep ConvNets, MICCAI International Workshop on Deep Learning in Medical Image Analysis, 2018.*

[18] Chi Wang, Yunke Zhang, Miaomiao Cui, Jinlin Liu, Peiran Ren, Yin Yang, Xuansong Xie, XianSheng Hua, Hujun Bao, Weiwei Xu, Active Boundary Loss for Semantic Segmentation.