Towards high performance cancer staging from histology images

Dissertation Defense

by

ASHWIN RAJU

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at The University of Texas at Arlington

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2022

To Mom and Dad.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Junzhou Huang for giving me this opportunity, supporting and believing in me. Dr Jiawen yao for hiss constant ssupport and motivation. I would also like to thank my colleagues and labmates at SMILE Lab for making my tim at UTA fun and productive. Moreover, I would like to thank my Ph.D supervising committee members, Dr. Jia Rao, Dr. Jean Gao, Dr. Dajing ZHu for their support and valuable insights during my Ph.D.

Last but not least, I would like to thank my parents Sathya and Raju and my wife Pooja for the continuous support and encouragement throughout my Ph.D. It would not have been possible without them.

12-August-2022

ABSTRACT

Towards high performance cancer staging from histology images

Ashwin Raju, Ph.D.

The University of Texas at Arlington, 2022

Supervising Professor: Junzhou Huang

Digital Pathology (DP) has been recently used in replacement to traditional microscopy samples as it easy to navigate and can be analysed, processed and saved. With the invention of Digital pathology, there has been exponential increase of automated process to make the life of Doctors easier. One such automated process is Artificial Intelligence (AI) where the AI is used as an assistant to Humans and to make the analysis and guide the experts. With the advent of AI and in particular Deep Learning, research has been divided and focused to solve multiple problems in Digital Pathology. One such important application is to analyse the Whole Slide images (WSIs) of patients and predict Cancer stages for the patient. This is crucial because the WSIs becomes too many which requires Expert knowledge and time consuming job. This make a perfect application where Deep learning can be used.

In this dissertation, we address the problem of identifying WSIs of patients and predict the cancer stages. We further identify several important observations to improve the performance of WSIs. We extract the granular details of the WSIs and capture the spatial relationship of granular features. We use these Graph of granualr features to further classify the cancer stages of patients.

We conduct several experiments to prove our work experimentally and make conclusion that the proposed work can be used to predict cancer stages.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

1.1 A Gentle Introduction to Digital pathology and Whole Slide Image

Digital pathology (DP) has been a recent invention in medical domain to process the human tissues in a software. The need for a digitized pathology is make to process, analyse and store the observations of the tissues observed by the Human experts. Since the digital pathology has been software based analysis structure, there has been many ways to automate the software. The need for automation is to guide the expert to make their work easier. Along with the Software to handle digital pathology, research has been started in parallel to automate the digital pathology. Some of the automation work involves segmentation of nuclei cells, segmentation of tissues, analysis of cancer stages, analysis of survival rate of the patients.

The tissue that is stored in digital pathology is Whole Slide Images. Whole slide images are the way to process the tissue with different magnification. The different magnification levels are 5x, 10x, 20x and 40x. The need for different magnification levels is to allow the experts to zoom into more granular information which is necessary for solving downstream task.

1.2 Challenges and Motivation

In this section we will look into different challenges and need for automated software for handling Whole slide images.

Firstly, the magnifications are huge which makes it difficult to incorporate the whole slide image into to machine learning model. Research has been conducted to overcome the issue of handling large magnifications into the machine learning model. One such research area is to divide the whole slide image into different patches and perform automated analysis on the divided patches. Later rearrange the patches into Whole slide image to perform downstream task. There are still challenged in doing these ways. One such challenge is that when the whole slide image are divided into patches, the patches lose spatial relationship across the patches. This makes the research difficult to improve the performance of the task.

With this issue, we motivate the task to incorporate spatial relationship across patches and use granular information to improve the task of cancer staging.

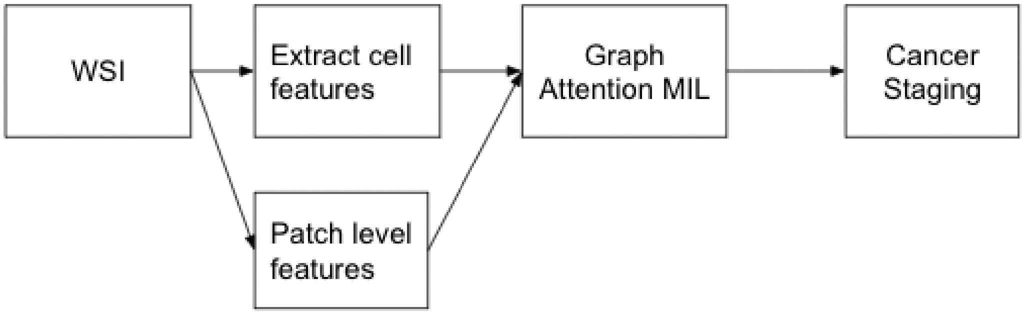


Figure 1.1: Research Overview

1.3 Dissertation Structure

The rest of this dissertation is outlined as follows. Chapter 2 makes an in depth understanding of Graph Attention Multiple instance learning. The need for Multiple Instance learning to improve the performance of cancer staging.

Chapter 3 discusses and evaluates a task by using shape as a constraint to have an interactive segmentation to segment nuclei cells in Whole Slide Images. Chapter 4 discusses and evaluates the task of using sub type cell features and to perform cancer staging based on the subtype cell features. Finally, Chapter 5 concludes this dissertation with future research directions and highlights the takeaways of this research.

CHAPTER 2

Graph Attention Multi-instance Learning for Accurate Colorectal Cancer Staging

2.1 Introduction

Colorectal Cancer (CRC) is one of the most common cancer diagnosed in humans. Outcomes vary significantly among patients with different tumor status. Accurate staging of colorectal cancer for personalized treatment is thus highly desired. Whole slide pathological images (WSIs) serves as the gold standard for Tumour Node Metastasis (TNM) staging. However, TNM staging for colorectal cancer relies on labor-intensive manual discriminative patch labeling, which is not suitable and scalable for large-scale WSIs TNM staging. Though various methods have been proposed to select key image patches to perform staging, they are unable to consider the structure of tissue types in biopsy samples which is a key evidence for determining tumor status. In this paper, we propose a Graph Attention Multi-instance Learning (Graph Attention MIL) with texture features, which encodes a spatial structure between patches and jointly predicts the TNM staging. We evaluated our proposed method on a large cohort of colorectal cancer dataset. The proposed framework improves the performance over the existing state-of-the-art methods indicating the future research towards graph based learning for TNM staging.

In this paper, we address the problem by considering the spatial relationship of tumor with other tissue partitions by introducing a novel Graph Attention Multi-instance learning network where multiple graphs, with each graph having nodes representing different tissues acts as an instance. The multiple instances for a whole slide

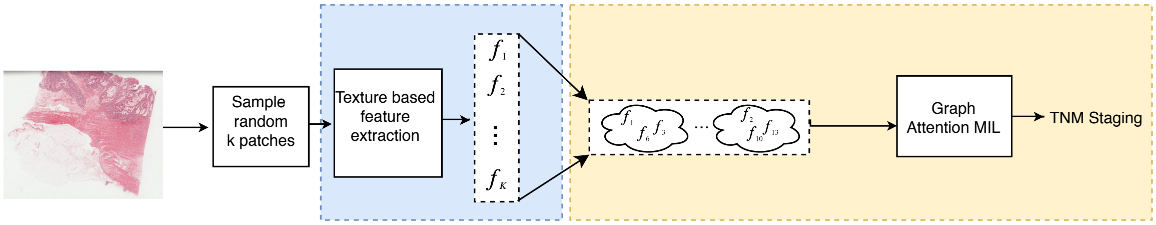


Figure 2.1: **Overview of our pipeline.** Given a WSI, we randomly sample k patches and extract texture features for each patch. The texture features are grouped into multiple graphs and each graph has features from all clusters. Here the cloud represents a graph. The Graph Attention Multi-instance learning is used to predict the tumor stage. (Best viewed in color).

pathological image (WSI) form a bag which aids to predict the tumor stage. We represent the node in the graph as the texture feature of an image patch. Different from the previous state-of-the-art methods, our proposed framework considers the spatial relationship between different texture features. To summarize our motivations, we introduce a texture feature extraction method to encode texture for an image patch and cluster similar texture features together. We then introduce a novel Graph Attention Multi-instance learning network to predict the tumor stage by considering a bag of multiple graphs with spatial relationship between tissues invoked in each graph. Extensive experiments verify the effectiveness of our proposed framework on a large cohort of CRC.

2.2 Methodology

We denote the WSI dataset $\mathcal{D}_\ell = \{X_i, Y_i\}$, with X_i denoting the WSI and $Y_i \in \{0, 1, 2, 3\}$ indicating different tumor stages. Given the rich information of a WSI, generally in range $(10^6 \times 10^6)$, we first extract random M tissue patches at 20X (0.5 microns per pixel) objective magnification with image size fixed to $224 \times 224 \times 3$. We then create a set of bags $B = \{B_1, B_2, ..., B_n\}$, where B_n contains randomly sampled M tissue patches for X_n. The training objective is to correctly classify the tumor stage with the given B_n from X_n. The overview of our proposed two step pipeline is outlined in Fig **??**. The first step is a texture extractor, which projects input raw image patches into a low-dimensional space. Motivated from [3, 4], we propose a texture based clustering auto encoder to cluster similar features into same cluster. Different from [5], where a pre-trained VGG model from ImageNet is used to extract the features, we propose a texture auto encoder to extract the textures from each patch as proposed by [4] and force the textures to be invariant to different augmentation of the same patch. The reason for using texture based encoding is because texture features encode domain specific information from the image patch. In step 2, the extracted texture features for B_i is used for our Graph Attention Multi-instance learning network to classify image level label for a WSI X_i. We explain each step in more details in the further sections.

Texture feature extraction. The goal here is to learn a feature representation for a given image patch. We focus on texture based feature representation which encodes orderless visual patterns of an image. Furthermore, we want the texture features for similar image patches to be close to each other and for dissimilar image patches far away from each other. To achieve this we use two main components, 1) Texture encoding network. 2) Cluster embedding network. Texture encoding network [4], uses a novel *learnable residual encoding layer*, which learns an inherent dictionary and domain specific information for a given image patch. The cluster embedding network, clusters similar textures by using a Siamese network to train a binary classification task, where similar textures are assigned the same class and dissimilar patches are assigned a different class. The Siamese network shown in Fig. 2.2 under Step I depicts how texture features are clustered and when trained till convergence the texture features are clustered like the Embedding space shown in

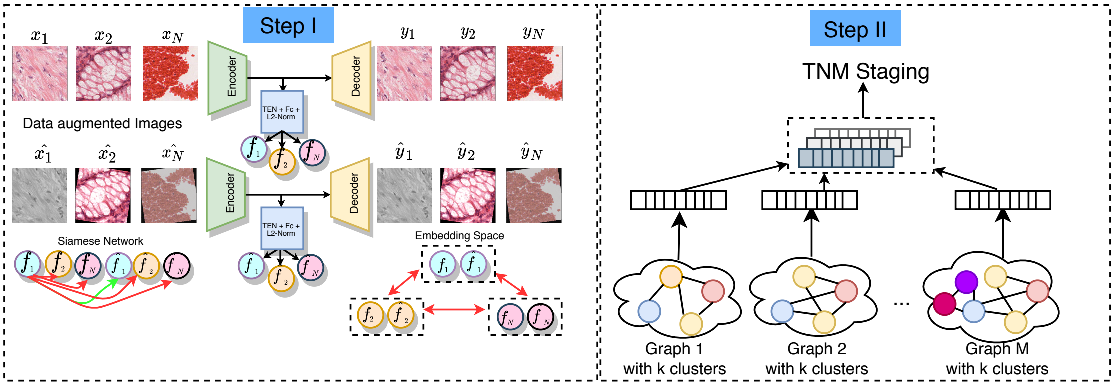


Figure 2.2: Step I of our architecture performs clustering of texture features. The siamese network in step I brings the patch P_i and data augmented patch $\mathcal{T}(P_i)$ to be same and the patch P_i and patch P_j to be different in the embedding space. Step II of our architecture contains many small graphs, where each graph contains random subset of patches from all C clusters. The feature pooled from each graph are represented as instance from the graph to perform attention multi-instance learning to predict the tumor Stage.

Fig. 2.2 under step I. We denote the image $P_i \in \mathbb{R}^{H \times W \times 3}$ as a randomly sampled patch from the WSI and use ResNet50 [6] as an encoder to extract feature map $E_i \in \mathbb{R}^{\frac{H}{R} \times \frac{W}{R} \times C}$, where R indicates the downsampling factor and C is the number of features. We remap the feature map E_i back to the image resolution using the decoder proposed by [7]. We follow the deep texture encoder network [4] to encode the visual descriptors $E_i = \{e_1, .., e_N\}$, where N is the number of spatial locations in the feature map, to a fixed length representation $F = \{f_1, .., f_K\}$, where K indicates the number of texture features.

The textures of same image under different data augmentation should be close to each other and should be spread-out for different images. Inspired from [3], we apply data augmentation $\mathcal{T}(.)$ to slightly modify the image patch. For an instance i, where the original image P_i and data augmented image $\mathcal{T}(P_i)$, denoted by $\hat{P}_i$ with their corresponding $\mathcal{L}_2$ normalized texture features f_i and $\hat{f}_i$ should be classified into

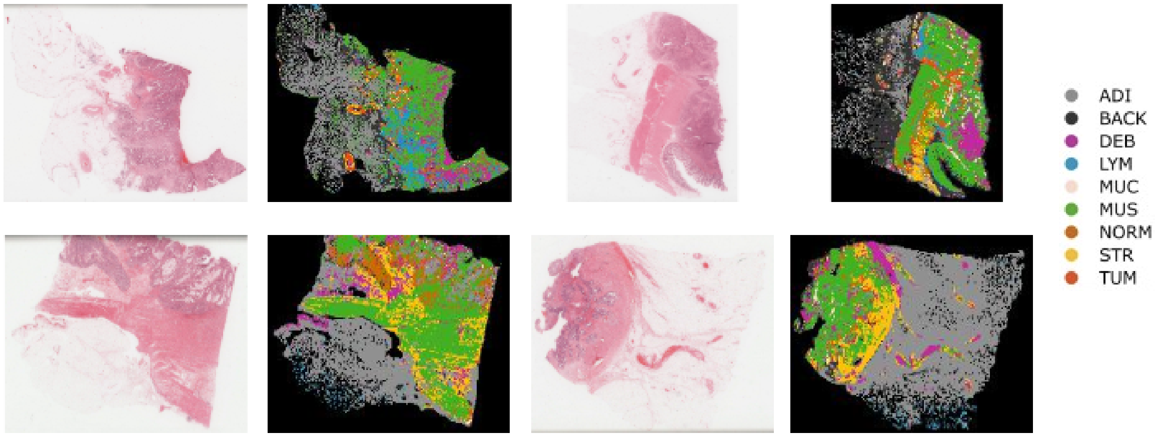


Figure 2.3: **Clustering overview.** Each pixel in the colormap represents a texture feature and representing different tissue types provided by [1].

instance i, and for other instances, $j \neq i$ shouldn't be classified into instance i. We train a Siamese network as proposed by [3] to achieve this objective.

Following the texture feature extraction, we assign cluster labels for each texture feature. Assigning the labels to each texture feature for a WSI aids to stratify the random patches so that each graph in our Graph Attention Multi-instance learning network can have different distribution of tissue types. In order to assign a label for similar image patches, we take use of tissue wise annotated CRC dataset [1] which we call as reference dataset and randomly sample 100 patches from each tissue from their training dataset. The reference dataset is used only to assign a label for image patches. For a given l_2 normalized texture feature f_i and $F \in \mathbb{R}^{M \times D}$, where M denotes all the l_2 normalized texture features from reference dataset [1] and D indicates the dimension of feature, we assign the cluster label by applying weighted kNN proposed by [8] with k set to 1. Fig. 2.3 shows the result of our texture features that has been assigned to different cluster labels. Our clustering approach based on texture features is dataset invariant and can be visually seen in Fig. 2.3.

To optimize step I, we use mean squared error to minimize the distance between predicted pixel and ground truth pixel. To optimize texture feature embedding in step I, we use the loss function mentioned by [3]. The overall loss function is defined as:

$$L_{step1} = L_{mse} + \lambda_1 L_{texture} \tag{2.1}$$

where L_{mse} is applied for each image in the batch, $L_{texture}$ as defined by [3] is applied for texture features in the batch.

Graph Attention Multi-Instance Learning. Once the patches have been clustered into different tissue types as shown in Fig. 2.3, the next step is to use the texture features F from the patches randomly sampled from the WSI to predict the tumor stage. The goal is to learn the relationship between the features from each WSI jointly to predict the tumor stage. Graph Convolutional Network (GCN) provides a good direction to consider information exchange between nodes and the spatial structure of medical imaging data [9, 10, 11]. In our work, we incorporate spatial information of different tissue features in the form of graph learning. We decompose each WSI into multiple graphs and each graph has features from all the cluster labels with approximately equal number of patches from each cluster. The multiple graphs are treated as multiple instances in a bag and the bag for each WSI is used to predict the tumor stage using attention multi-instance learning (MIL) as depicted in Fig. 2.2 (step II). We use Adaptive GraphSage proposed by [10] to create a graph of nodes where nodes present texture features extracted from the image patches. The reason for using Adaptive GraphSage over other graph networks is because the ability to learn the embedding feature between nodes more effectively as mentioned in [10].

To construct the adjacency matrix for our graph we follow [10]. Formally, the adjacency matrix can be written as:

$$A_{ij} = \begin{cases} 1 \quad \text{if } j \in KNN(i) \text{ and } D(i,j) < d \\ 0 \quad Otherwise \end{cases} \tag{2.2}$$

where $D(.)$ is the euclidean distance, d is the manually selected threshold, KNN is the K nearest neighbour to the patch i. Based on the empirical analysis, here we set the threshold for d as 0.4. We follow the architecture proposed by [10], where the nodes are represented as texture features and the adjacency matrix defined from Eq(2.2) to create multiple graphs with shared weights. We use k graphs, where k is set 7 based on the experiments and each graph contains randomly sampled features from C clusters from a WSI. Here we set C as 9 based on the reference dataset [1].

To extract texture feature from a graph $\mathcal{G}_i$, we use the concatenation of max operation and mean operation on the node embeddings [12]. The feature pooling (FP) operation can be defined as:

$$FP = \frac{1}{N}\sum_{i=1}^{N} x_i \| \max_{i=1}^{N} x_i \tag{2.3}$$

where N is the number of nodes and x_i is the output embedding feature for each node in the graph network. From multiple graphs, we extract features $\{FP_1, FP_2, ..., FP_k\}$, where FP_k represents feature from the graph $\mathcal{G}_k$. We consider the set of features as an instances in a bag and train an attention MIL [13] to predict the tumor stage. The learnable attention weights in the MIL gives more importance to instances in the bag which are responsible for predicting the tumor stage. The output from the attention MIL yields a feature vector which is connected to a linear classifier to predict the tumor stages. To optimize Graph Attention Multi-instance learning, we minimize the loss function as follows:

$$L_{step2} = \frac{1}{M} \sum_{i=1}^{M} H\left(p_i, q_i\right) \tag{2.4}$$

where M is the number of WSIs and $H(.)$ is the weighted cross entropy loss between ground truth p_i and prediction q_i.

2.3 Experiments

Dataset description and Baselines. The data we used, Molecular and Cellular Oncology (MCO) study [14, 15], is a collection of imaging, specimen, clinical and genetic data from over 1,500 Australian individuals who underwent curative resection for colorectal cancer from 1994 to 2010. To evaluate our models, we split the total dataset containing $1,345$ WSIs with 115, 202, 698 and 330 stage I to stage IV cancer WSIs into 70%, 10%, 20% for training, validation and testing, respectively. Each WSI has been annotated with a image level label representing the tumor stage by the expert pathologist. We extracted 1000 random patches at 20X magnification from each WSI, covering approximately 82% of tissue area. In total, $1,345,000$ patches (more than 1 million) patches were extracted from the dataset. We build several baselines for comparisons. To evaluate results from directly using down-sampled WSIs, we treat the WSI as a reduced image resolution with size (2048×2048) and then train a image classification model. We also compare with other state-of-the-art models for WSIs classification and survival prediction task and accordingly modify them for colorectal cancer staging. The comparison method includes Tellez et al. [16], Gupta et al. [1]

Table 2.1: Performance comparison of the proposed method and other existing related methods using mean accuracy and mean F1 scores are tabulated.

Model	**Accuracy**	**F1**
Baseline	53.6	50.9
Tellez et al. [16]	63.2	62.8
Yao et al. [5]	66.8	65.2
Gupta et al. [1]	71.5	60.8
Yao et al. [5] with our step I	74.5	72.5
Cell graph [10] with 200 patches	69.7	67.2
Cell graph [10] with 1000 patches	79.8	77.4
Proposed	**81.1**	**79.8**

and Yao et al. [5]. We also perform ablation study on our proposed architecture in step II with graph CNN proposed by [10].

Results and discussions. For training step I, we used Adam optimizer with learning rate set to 0.003 and reduced the learning rate for every 20 epochs. We used random horizontal and vertical flipping, random crop, random rotation and Gaussian noise as transformation operation $\mathcal{T}(.)$ in step I. For training step II, we used Adam optimizer with learning rate set to 0.0003. Table 2.1 shows the performance of our proposed method with other comparison methods. The baseline method trained on reduced resolution with (2048×2048) achieves the 53.6% which demonstrates down-sampled WSIs is not useful for staging. The reason is due to the reduced resolution loses much information and many details from the original WSI. Tellez et al. with contrastive network, uses a time consuming encoding step to predict the image level label whereas our proposed method extracts random patches from the WSI to predict the image level label. Gupta et al. [1] uses a labelled patch level label to train a patch level classifier and uses a network similar to step II proposed by Tellez et al. to predict image level label. The step II of Gupta et al. is as time consuming as Tellez et al. Yao et al. [5] used kmeans clustering on 1D features extracted from pre-trained VGG model whereas our proposed step I trains to bring similar patches closer and dissimilar patches far from each other.

From the Table 2.1, we can see the model [5] with using k-means to cluster features performs 0.668 on our test dataset whereas when replaced k-means with our clustering method the performs increases by 9%. The reason for the improvement in the performance is using texture based feature extraction and Siamese network to make similar features close to each other and dissimilar features far from each other. However, both the models does not perform better than our proposed method, the reason is DeepMIL [5] does not consider relationship between node features to predict the image level label. To predict tumor stage, the spatial relationship between features is important. We also compared with step II architecture with a single graph proposed by [10], to show that our proposed multi graph Attention MIL can perform better than a single Graph network. The single graph with 1000 patches from the features extracted from the step I of our pipeline achieves 0.7987 but is computationally slower as the time complexity for graph CNN is $O(n^3)$ [17], where n is the number of nodes in the graph. We also compared with 200 patches using a single graph, however it yields a lower accuracy than the single graph with 1000 patches. We have showed the confusion matrix of our proposed method and the cell graph with 1000 patches in the supplementary material.

Our proposed method uses multiple graphs with each graph having randomly sampled features from the 1000 patches for each WSI. Each graph also has features from every tissue types represented by [1]. The time complexity of each graph is less than a single graph and multiple graphs are considered to be instances in a bag to predict the tumor stage. For fair comparison, we use the same Adaptive

GraphSage architecture as [10]. From the table 2.1, our proposed method achieves best performance when compared to other compared methods.

2.4 Conclusion

We proposed a novel Graph Attention Multi-instance learning framework to learn the spatial relationship between image patches. Here we also demonstrate that the Graph network using textures result in a better performance when evaluated with other previous state-of-the-art methods. Future research will focus on how our proposed method can also be used on different task such as predicting Overall survival rate of an patient.

CHAPTER 3

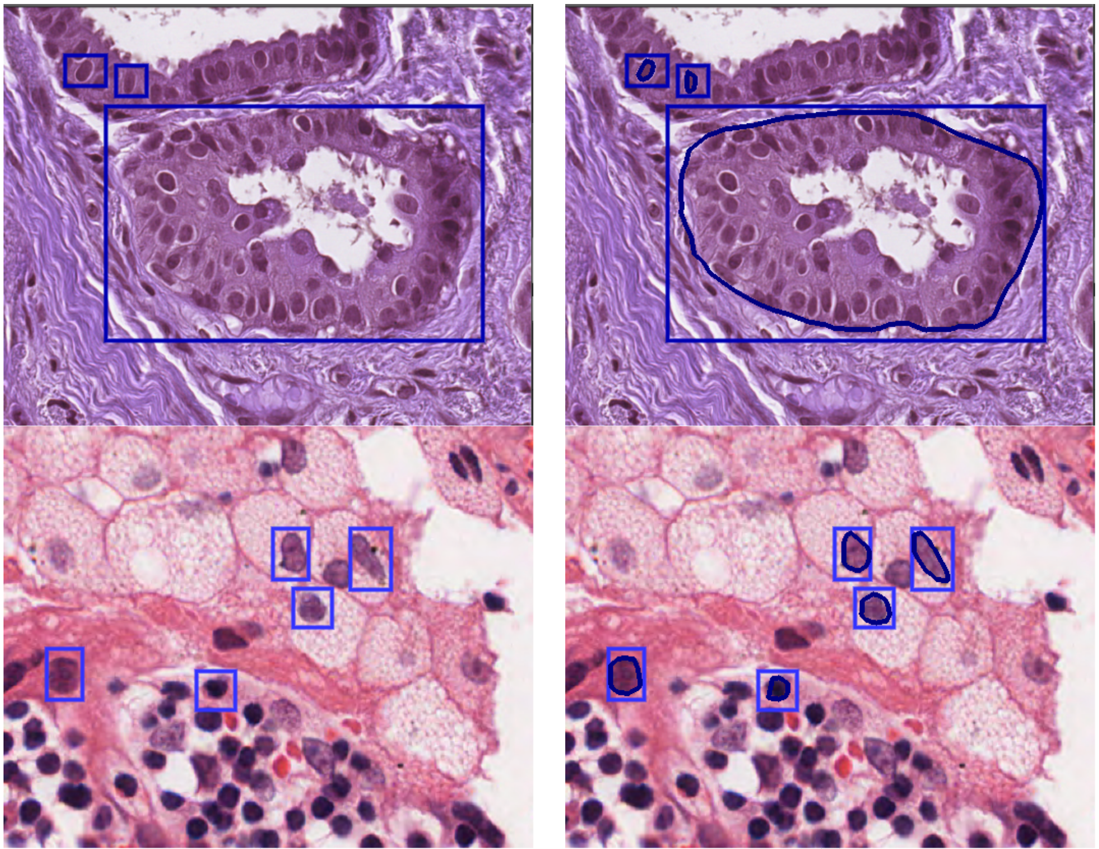
Interactive Segmentation of Microscopy Images through Shape Prior

3.1 Introduction

Assessment of Haematoxylin and Eosin (H&E) stained histology slides rely on the segmentation of nuclei cells or glands from microscopy images. Manual assessment including segmentation suffers from low throughput and is prone to intra and inter-observer variability. To overcome the difficulties of manual assessment, Automated analysis of nuclei cells or tissue types from digital pathology (DP), where whole slide images WSIs are acquired using scanning devices have become a recommended approach.

Deep learning (DL) approaches have become state-of-the-art methods in nearly all computer vision tasks recently. In medical imaging, DL approaches are used to tackle a wide range of problems. However, there are considerable requirements for DL approaches to be successful in the medical domain. Firstly, DL methods are computationally expensive and secondly, they are data-hungry. To solve the computation resource constraint, several methods have been proposed to use fewer memory resources without sacrificing performance. On the other hand, to handle the data requirement constraint, several methods have been proposed to synthetically expand the dataset [18]. However, research shows that synthetically expanding the dataset or performing any data augmentation on the given dataset does not generalize the DL method. If DL methods are trained from one particular domain then they over-fit on that domain and do not generalize on all domains. This becomes a bottleneck when DL methods are to be deployed to production where a wide range of datasets are to be tested based on the trained model. Hence, to have a generalized model which could perform significantly well on all different domains, one needs to consider training on different datasets.

Collecting annotations for several domain datasets is labor-intensive and involves expert knowledge. In particular, for collecting segmentation labels, one has to classify each pixel into one of the respective class labels. To be more specific, annotating one nucleus takes 5s, a visual field containing 100 nuclei takes 17 minutes which is a considerably tedious process [1]. Therefore, gathering a large dataset for training different domains is a tedious task. To solve labor-intensive problem, research has been conducted to use semi-supervised approaches or weakly supervised approaches [19, 20]. Semi-supervised approaches use partial annotation available to segment dataset from unlabelled datasets. In this way, the accumulation of new dataset to the training datasets is reduced which tries to solve the problem of domain specification. However, the performance of semi-supervised and weakly supervised approaches are not on par with supervised approaches [21]. The efficient and better strategy is to use semi-supervised approaches in an interactive way. Semi-supervised approaches with user interactions are recommended due to the following. Firstly, User has the control over the region of interest for which the segmentation model needs to segment the nuclei cells or glands. The segmentation model uses a trained model where the region of interest is provided by the user. Secondly, the user has the control to alter the segmentation result if needed to make sure the segmentation prediction is close to the desired results. Semi-supervision using user interactions, therefore, reduces the time taken to annotate the H&E images as well as produce high quality segmentation ground truth. Research has focused on how to introduce minimum user interactions and get high quality annotation labels. Even with the recent advancement in interactive semi-supervised segmentation, the approaches uses labelled dataset to train the



(a) (b)

Figure 3.1: Figures in column (a) shows the region of interest provided by the user. Figures in column (b) shows the segmentation masks predicted by our segmentation model.

models. To do this the annotations have to be provided by the experts to train the DL methods, which is indeed a time consuming process. In this paper, we focus on using minimum user interaction and weak supervision without burdening the annotators to label any new segmentation masks for training the interactive segmentation model. We do this by invoking shape prior to the architecture and allowing the segmentation model to predict a segmentation mask that is similar to the shapes from the pre-collected dataset. Overall, our main contribution in this paper is as follows:

- We propose a unified interactive segmentation by using shape priors to segment the region of interest.

- Our extensive experiments shows that by using our proposed user interactive segmentation model, the annotation time is reduced significantly and is able to produce high quality annotation results.
- We perform an extensive ablation study to show the need for a simple bounding box user interaction as a preferred way rather than other existing approaches.

3.2 Related Works

3.2.1 Nuclei cell and gland segmentation

Several automated methods have been proposed to segment nuclei cells and glands from H&E slides. Before the era of DL methods, [22] used threshold to obtain certain markers and the energy landscape as input for the watershed to extract nuclei cells. Similarly, [23] uses graph based learning to segment a gland structure within the region of interest. However, these methods do not generalize on different H&E slides and particularly when H&E images are noisy. After DL methods became popular in computer vision, several automated methods were proposed to segment nuclei cells and glands from various modalities [24, 25, 26]. Unlike other medical image segmentation, segmenting nuclei cells or glands is a hard problem due to the following reasons. Firstly, nuclei cells or glands are of different shapes and sizes and some are hard enough which makes the segmentation model to miss. Secondly, the nuclei cells or glands overlap with each other making the segmentation model to focus on how to separate the overlapping objects from each other. [26, 27] uses contour based learning as an added cue to separate overlapping cells in H&E. These methods use fully convolutional networks (FCN) based architecture which maps pixel to pixel. Research has also been focused on treating nuclei cells or glands as instance. Cell-RCNN [28] uses Mask-rcnn [29] based framework to detect each nuclei and segment the cropped nuclei. Cell-RCNN can be directly be imported to predict glands in microscopy images. Our proposed method uses instance segmentation framework following Cell-RCNN [28] feature backbone. Our architecture uses bounding box as an user input and crop the features within the bounding box. These cropped features are used to segment the object thus making the architecture similar to Mask-rcnn.

3.2.2 Interactive segmentation

A wide range of interactive segmentation approaches has been proposed using machine learning in the medical imaging domain [30]. In some methods [31, 32] interactive segmentation is formulated as energy minimization on a graph defined over objects. [33] uses interactive segmentation with active learning to segment 3D images. With the advancement of DL, research has been focused on using DL to interactively segment objects in medical imaging. In order to reduce the annotation time of users, several methods have been proposed to use extreme points [34, 35], scribbles [36, 37], boundary points [38], single point [39] to segment the objects from the image. In extreme points, Users are required to input 4 extreme points in the case of 2D or 6 extreme points in the case of 3D as an additional input to segment the required object. In scribbles, the user has to point to the area that needs to be focused in the form of a scribble. All these approaches need the supervised signal to guide the interactive segmentation. Therefore, there is a requirement for collecting supervised datasets for these tasks. In our proposed approach, we use a simple bounding box as a user input but we do not use any pixel wise supervision. We rather make the segmentation model output the shape that resembles the shape of the dataset collected.

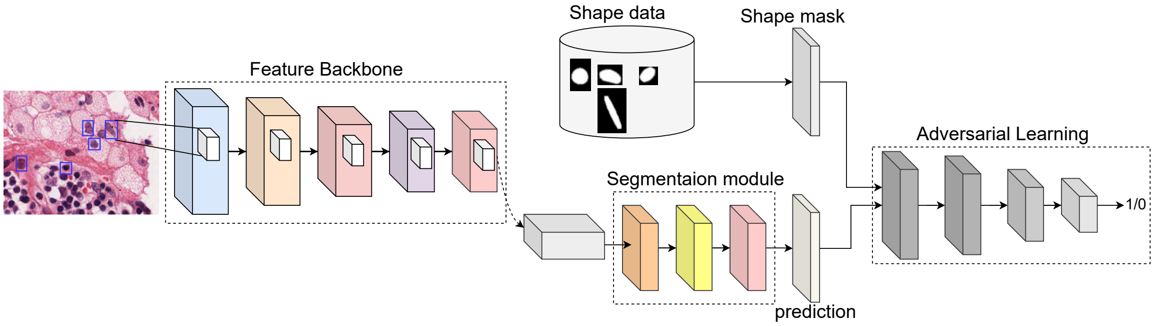


Figure 3.2: **Overview of our proposed model.** Latent feature map for a given bounding box is extracted using a feature backbone. The segmentation module predicts the segmentation for the given feature map. The adversarial shape loss is computed with the predicted segmentation mask and with the shape dataset.

3.2.3 Adversarial Domain Adaptation

Adversarial domain adaptation (ADA) for segmentation has received a recent attention in medical imaging [40]. The main idea of domain adaptation is to align the predictions or feature space of the segmentation module to that of the shape dataset. Here the shape dataset generally refers to a database of ground truth annotation masks collected from various sources. In [41], semi-supervised adversarial learning network is performed with a deep atlas prior and the network is trained to map the predictions to the annotation dataset. Similarly, [42] uses predictions from the target dataset (un-labeled) to map to the predictions of the source dataset (labeled). Different research have been used to adapt the output from segmentation to the source or defined dataset. [43] uses entropy to regularize the errors between source and target domain. The advantage of using adversarial domain adaptation is that the domain of input need not be from the same dataset. Since the domain adaptation tries to map the predictions, the input can be captured from different domains. In our proposed method, we use ADA as a signal to make the predictions of the segmentation module to be the same as that of the shape dataset.

3.3 Methods

In this section, we explain the proposed framework for interactive segmentation. Figure 3.2 summarizes the architecture.

3.3.1 Segmentation input

The input to our interactive segmentation framework is an image X^i from dataset $D = \{X^i\}; i = 1, 2, ..., K$, where K is the total number of images in the training dataset. For each input we randomly choose bounding boxes $B^i; i = 1, 2, ..., M$, where there can be M bounding boxes for each input X^i. The goal of the segmentation model is to input the image X^i and random bounding boxes from the set B^i and output a segmentation for each bounding box in B^i. We also have a large set of ground truth masks $G = \{Y^j\}, j = 1, 2, ..., N$, where N is a fairly large ground truth masks extracted from several datasets to induce shape priors to segmentation model.

3.3.2 Feature Backbone

The goal for the feature backbone is to have a $f(X^i, B^i)$, where f is a feature backbone that outputs a latent feature map of size $B \times C \times H \times W$, where B is the batch size, C is the number of channels, H and W are height and weight of feature map. The number of feature maps is based on the number of bounding boxes fed to the framework provided by the user. In our experiment, we fix height and width as 14 respectively and the number of channels as 256. For Feature backbone, we use a resnet101 with feature pyramid networks (FPN)[28] and initialize the pre-trained weights for feature backbone trained from [28]. We then freeze the layers of feature backbone i.e, the gradients are not passed to the feature backbone while training the interactive segmentation, making the feature backbone as a feature representation network. The features corresponding to each bounding box are cropped and aligned using RoIAlign [29] to resample to an unified height and width. In figure 3.2, under feature backbone section shows that each bounding box provided by the user in cropped using RoIAlign.

3.3.3 Segmentation Backbone

The goal of the segmentation backbone in our interactive segmentation is to input the feature maps cropped and aligned using RoIAlign for the given bounding box in an image. The output of the segmentation model is segmentation probability prediction with size $B \times 1 \times H \times W$, where B is the number of bounding boxes provided by the user for an image, H and W are height and width of the segmentation prediction. We use H and W as 28 similar to [28]. Unlike feature backbone, we do not initialize the weights from pre-trained model and do not freeze the segmentation architecture.

3.3.4 Adversarial Domain Adaptation

Once the segmentation produces a segmentation output, the goal of adversarial learning is to make sure that the segmentation output adapts the shape of annotation labels from the shape dataset. We collect the shape dataset from various nuclei cells and glands from different sources. The shape data is resampled using RoIAlign to the size same as the segmentation output size. The ADA component in our segmentation model adapts the distribution of the shape dataset and incurs loss when the predictions of the segmentation module looks different from the shape dataset. More formally, let $d(.)$ be defined as an FCN discriminator, then the discriminator loss can be expressed as

$$\mathcal{L}_d = \frac{1}{N_\ell} \sum_{\mathcal{D}_\ell} \ell_{ce}(d(\hat{Y}_s), \mathbf{1}) + \frac{1}{N_u} \sum_{\mathcal{D}_u} \ell_{ce}(d(\hat{Y}_t, \mathbf{0})), \tag{3.1}$$

$$\mathcal{L}_{adv} = \frac{1}{N_u} \sum_{\mathcal{D}_u} \ell_{ce}(d(\hat{Y}_t, \mathbf{1})), \tag{3.2}$$

Where D_ℓ and D_u represent the data from a different domain and the shape dataset acquired from various sources respectively, ℓ_{ce} represents a pixel-wise cross entropy. The discriminator loss $\mathcal{L}_d$ distinguishes between the segmentation output from shape dataset and from the segmentation model. The adversarial loss $\mathcal{L}_{adv}$ fools the segmentation model to classify the segmentation output to be the same as the shape dataset.

3.4 Implementation details

In this section, we look into the implementation details and time analysis for our proposed model.

We use detectron2 [44] to implement the proposed framework. Some of the important hyper-parameters which needs to be carefully tuned while training the model are the learning rate of adversarial (λ_l) and discriminator (λ_d). We set the λ_l as 0.001 and λ_d as 0.0001 based on the experiments and stability. We use SGD optimizer to train adversarial learning and a batch size of 32. We set the height and width of RoIAlign output to be 14 and the output height and width of segmentation output to be 28. For discriminator network, we pass the segmentation predictions and ground truth mask. We allow the discriminator to predict if the output comes from the target or source respectively. Table 3.1 shows the architectural structure for the discriminator.

We analyze the time taken during the inference stage for our proposed model. The average time taken for a single region of interest provided by the user to get the segmentation result is 0.2s, which is 30 times better than the time taken to annotate each pixel manually. The average time taken to annotate an image slide of size 256×256 with an average of 200 nuclei cells is around 10s whereas it takes an average of 7 minutes to annotate through pixel wise manually. All evaluations are done using RTX 1×1080 Ti GPU.

Table 3.1: Discriminator architecture: Here Conv refers to convolution + batch normalization + relu, FC refers to dense layer.

Model	Input size	Output size
Conv1	$1 \times 28 \times 28$	$32 \times 14 \times 14$
Conv2	$32 \times 14 \times 14$	$64 \times 7 \times 7$
Conv3	$64 \times 7 \times 7$	$128 \times 7 \times 7$
AvgPool	$128 \times 7 \times 7$	$128 \times 1 \times 1$
FC	128	1

3.5 Experiments

In this section, we will look into datasets that are used in our experiments and the state-of-the-art methods that are used to compare with our proposed model.

3.5.1 Datasets

To evaluate the proposed method, we need a shape dataset which contains nuclei cells and gland segmentation masks. We also need dataset which is different from shape dataset and which does not contain segmentation masks. To do this, we explain the datasets that we use in the following:

Shape dataset: To extract the nuclei cells and glands, we use the nuclei cells segmentation datasets [2], extracted at 40× magnitude. The aggregated dataset contains $495,179$ nuclei cells. For gland dataset, we use Gland Segmentation Challenge Contest in MICCAI 2015 (also named as Warwick-QU dataset) [45]. Totally our shape dataset contains nuclei cells as well as gland masks. To experiment on the shape dataset, we split the images into 70/10/20 with training, validation and testing.

CRC dataset: CRC [46], which contains 139 images taken from WSIs representing colorectal cancer. The images are divided into normal, low grade and high grade based on the degree of gland category. To conduct the experiment, we separate the dataset into 70/10/20 with training, validation and testing images respectively. For each images, we allow the annotator to provide bounding boxes for random nuclei cells and glands. For validation and testing, we allow the users to segment random sample of regions to evaluate the performance of the proposed model.

BACH dataset: BACH [47], which contains 400 slides extracted from WSIs representing breast cancer. The images are divided into normal, benign, in-situ and invasive cancer types. Similar to CRC dataset, we split the dataset into 70/10/20 with training, validation and testing images respectively.

3.5.2 Comparison Methods

To compare with other state-of-the-art segmentation methods, we consider both fully supervised methods as well as interactive segmentation methods. Even though our proposed approach is focused on interactive segmentation, we compare it with few state-of-the-art fully supervised methods. For fully supervised methods, we consider the following methods.

DCAN [26] is a popular segmentation model for medical image segmentation similar to UNet [24] architecture. DCAN uses contour as an aid to separate the overlap between glands. The output of the DCAN model is a pixel level class which is same as that of the input size.

Cell-RCNNV3 [28] is a panoptic based instance segmentation model. It uses a two-stage pipeline to propose the proposals and segment the object.

To compare with interactive segmentation methods, we compare with the following.

DexTR [34] is an interactive segmentation method which uses extreme points as the additional input to guide the segmentation. The segmentation backbone is the deeplab-v3 which is used as similar to architecture proposed in DexTR.

DeepCut [38] uses bounding box as an user input which acts as a region of interest to crop the region and perform segmentation for the region of interest.

Scribble [36] uses scribble as an user input which allows the segmentation model as an additional guidance to segment the area of scope.

Nu-Click [39] uses a single click as the user input to allow the segmentation model to focus on the area of interest provided by the user.

IFCN [48] uses multiple user clicks as the user input to allow the segmentation model to segment the object.

The state-of-the-art comparison methods uses fully supervised dataset to train the segmentation model whereas, our proposed method does not rely on any supervised dataset, rather it relies on the shape dataset with pre-defined shapes that can be extracted from various domains.

Table 3.2: Quantitative results on Nuclei cell dataset. For HD, lower the better.

Model	DSC	HD
DCAN	84.1 ± 8.6	9.1 ± 5.4
Cell-RCNNV3	**86.3 ± 7.9**	**8.7 ± 5.7**
DexTR	80.3 ± 5.6	11.2 ± 7.2
DeepCut	79.5 ± 5.1	10.4 ± 6.5
Scribble	78.4 ± 6.3	12.3 ± 5.6
Nu-Click	83.7 ± 7.5	8.3 ± 5.1
IFCN	84.9 ± 7.1	8.4 ± 4.8
Ours	85.3 ± 9.2	8.3 ± 4.6

Table 3.3: Quantitative results on CRC dataset. For HD, lower the better.

Model	DSC	HD
DCAN	83.3 ± 7.6	8.9 ± 3.4
Cell-RCNNV3	85.2 ± 6.4	8.3 ± 3.9
DexTR	80.3 ± 8.9	12.2 ± 4.2
DeepCut	79.5 ± 9.5	11.3 ± 4.1
Scribble	79.4 ± 8.9	12.3 ± 3.9
Nu-Click	83.9 ± 6.7	8.5 ± 3.4
IFCN	84.5 ± 5.9	8.4 ± 4.1
Ours	**86.1 ± 5.4**	**8.1 ± 4.3**

Table 3.4: Quantitative results on BACH dataset. For HD, lower the better.

Model	DSC	HD
DCAN	79.2 ± 5.6	10.3 ± 5.6
Cell-RCNNV3	80.3 ± 6.3	9.5 ± 4.9
DexTR	77.3 ± 6.3	10.1 ± 6.1
DeepCut	78.1 ± 5.4	9.4 ± 5.3
Scribble	76.5 ± 7.5	9.9 ± 5.6
Nu-Click	79.8 ± 7.4	8.9 ± 5.1
IFCN	78.3 ± 6.9	8.8 ± 6.4
Ours	**81.2 ± 5.6**	**8.6 ± 6.1**

3.5.3 Observations

We can see from the table 3.2, that our proposed model outperforms the competitor interactive methods in both dice similarity coefficient (DSC) and Hausdorff Distance (HD) in the shape dataset. This makes us understand the need for shape prior to the segmentation model. The supervised method (Cell-RCNNV3) performs better than interactive segmentation on the sshape dataset. The reason is that, the supervised segmentation model is trained and tested on the same domain [2] and supervised methods are designed to perform better when trained and tested on the same domain. However, our interactive segmentation performs on par with the Cell-RCNNV3. We also report the scores evaluated on the CRC and BACH dataset in tables 3.3 and 3.4 respectively. As CRC and BACH images are not present in the shape dataset, we treat this particular problem as domain generalization. In CRC dataset, our proposed model outperforms both supervised segmentation model as well as other interactive segmentation models. Similarly, our proposed model also outperforms in BACH dataset in both supervised as well as interactive segmentation models.

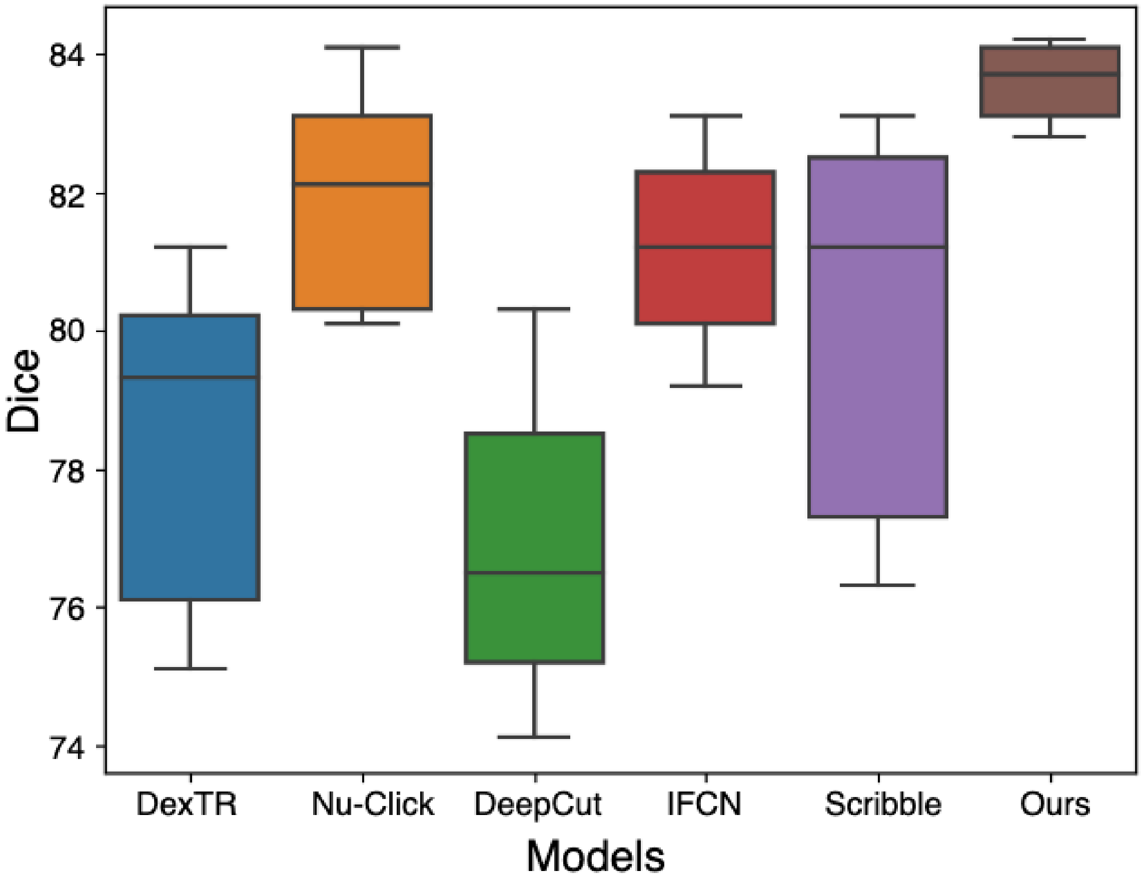


Figure 3.3: **Intraobserver variability.**

3.6 Ablation study

In this section, we will look into each component and its need for our proposed method.

3.6.1 Need for bounding box interaction

We analyze the need for bounding box interaction as a user input to facilitate the annotation process. Figure 3.5 shows the average time taken to annotate the slides. The average time is taken across the test dataset from CRC dataset with the approximately 100 nuclei cells in each slide. We find that the DSC of our proposed

Table 3.5: Our proposed method without the shape constraint and treating as a fully supervised method outperforms when trained and tested on the same domain. However, it fails to outperform when tested on the images from different domain.

Model	Domain A	Domain B
Ours w/o ADA	84.4 ± 3.1	80.6 ± 7.4
Ours	83.5 ± 4.1	82.1 ± 3.8

method is 86.1 which is significantly better than the state-of-the-art interactive segmentation methods. The reason for relatively less time to annotate is because we use 4 points to compute the segmentation mask. We can also find that Nu-Click takes less time to annotate as it needs 1 point to annotate the nuclei cell, however, the DSC of Nu-Click is not better than our proposed method.

3.6.2 Number of bounding boxes per input

We analyze the impact of the number of bounding boxes that are needed to improve the performance scores in the test dataset. We do this by considering the annotated segmentation masks performed by the user interaction as a part of the training dataset and train a segmentation model. We see that, as the number of annotation bounding boxes increases the DSC of segmentation model increases in the test dataset.

3.6.3 Intraobserver variability

We analyze the intraobserver variability across different interactive segmentation models. To do this, we allow 5 users to annotate the H&E slides. We compare the acDSC results of 5 users to understand the generalizability of our proposed model. Figure 3.3 shows the box and whisker plot of the intraobserver variability. We can see that the variation among 5 users for our proposed interactive segmentation is significantly lesser than the competitor methods. The reason for the less variance is because we make the segmentation prediction to resemble the shape of the pre-defined shape dataset. However, the user interactions such as scribbles need expert knowledge on the boundary as the nuclei cells or glands overlap with each other.

3.6.4 Need for shape constraint

We analyze the need for an adversarial learning mechanism to learn the segmentation model through shape loss rather than simple supervised learning. To do this, we test our proposed model by replacing the adversarial learning component. We replace the adversarial learning component with fully supervised learning. We see that the supervised learning component do increase the DSC of the test dataset taken from the same domain but when tested on a different domain, the fully supervised learning fails to perform better. On the other hand, our proposed model with adversarial learning component generalizes across various domains. Table 3.5 shows the DSC evaluation on two datasets. When adversarial learning is not used in our segmentation model, we train and test on dataset taken from Domain A as well as train in Domain A and test in Domain B.

3.7 Conclusion

In this work, we show an interactive segmentation architecture which is easy to use and takes comparatively lesser time to use when compared to state-of-the-art interactive segmentation methods. We also show that our proposed method outperforms the competitor methods in both DSC and HD metrics. We do this by allowing the model to learn the shape of the nuclei cells rather than a simple pixel-wise segmentation. With the shape learning mechanism, our model is able to generalize with different image domains. We perform several ablation studies to further prove the importance of each component in our segmentation model. In the future, we aim to use the proposed interactive segmentation model on volumetric datasets.

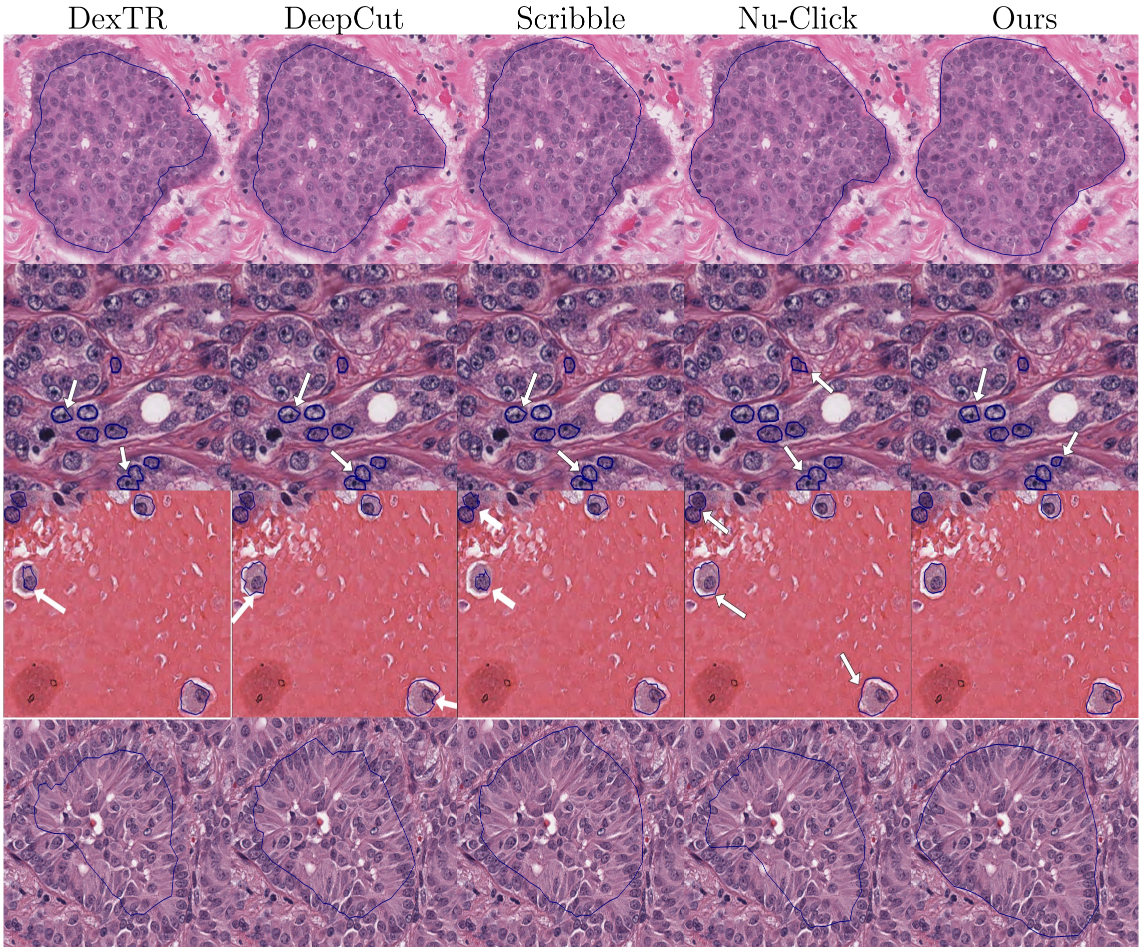


Figure 3.4: Qualitative comparison of our proposed model. The first two rows are taken from CRC dataset and second to the fourth row are taken from BACH dataset. We show that our proposed model is able to segment the region of interest provided by the user. For simplicity, we only allow the user to segment some of the nuclei cells or glands randomly in an image.

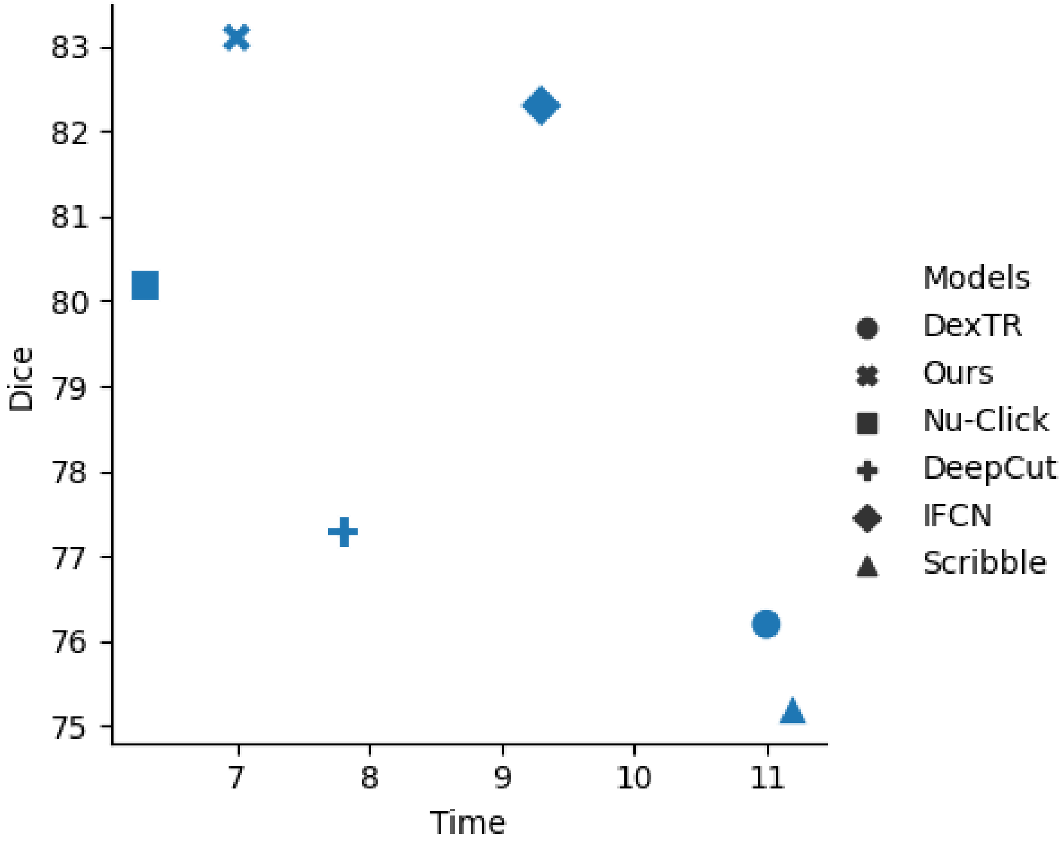


Figure 3.5: **Average inference time comparison.** Time is denoted in minutes

CHAPTER 4

Towards accurate histology image classification through subtype granular information

4.1 Introduction

Histopathological analysis is a crucial procedure in the early diagnosis of diseases such as cancer. Pathologists typically analyze the histological properties in the digitized tissue samples called WSIs to search for cancerous regions. However, these histology images are significantly larger as each image may contain thousands of cells which can be time-consuming and subjective [10]. Computer Aided Diagnosis (CAD) systems can be used to alleviate the procedure by providing objective analysis to the pathologists.

With the advent of deep learning, Convolutional Neural Networks (CNNs) have been widely used in medical imaging to achieve state-of-the-art-results in various tasks. In pathology, CNNs are widely used to produce promising results in nuclei cell segmentation [49, 28], cancer region segmentation [50, 51], survival analysis [52, 11], cancer staging [53, 54, 55]. Cancer staging or tumor classification requires high-resolution cellular level information to capture the micro-environment. In general, WSIs are processed in a multi-resolution format with the highest resolution reaching up to $10^6 \times 10^6$ pixels at 0.25 mm containing millions of cells. To cope up with the large size of WSI, the automated methods sample image patches from WSI to perform any downstream tasks. However, these methods come with multiple drawbacks. First, sampling patches from WSIs lose spatial relationships with each other. Due to the lack of contextual information between tissues which is the key essential for cancer grading/ tumor classification, the underlying tissue architecture is missed. Second, there is a direct correlation between patch size and context information. The Larger the patch size, the more context information is provided to the automated methods. However, the large patch size leads these automated methods difficult to capture the granular information and cell-level information in an image. Large patch size further adds computational burden to CNNs. To overcome the loss of spatial information, methods [53, 55, 56] use context aggregation mechanism to encode neighboring tissue information in WSI. However, these methods require encoding each image patch size into a latent vector which loses cell-level interaction information.

Different types of cells and their interactions play an important role in initiation, development and deducting therapeutic response for tumor. In [57], Histopathology analyses show that variable amount of infiltrating immune cells are found in and around the tumor region. Similarly, Macrophages, mast cells are found surrounding the tumor cells and lymphocytes are not randomly distributed but located in specific areas which indicates a specific cancer type. With the importance of cell level interaction to predict the cancer staging/tumor type, the research focus is how to make automated methods efficiently use cellular features into their frameworks. In [10, 58, 59], different cell-graphs are proposed to encode the cell-level interaction to predict tumor grades. These methods demonstrated the performance improvement by encoding cell-level interactions. Motivated by the evidence of different cell types in and around the tumor region to predict tumor grades [57, 60] and the benefits of cell-graph modeling, we propose a novel framework to embed cell subtype features in a cell-graph to predict different tumor types/tumor grades. We propose a discriminative latent feature extractor to represent different nuclei cells, by this way we eradicate the use of handcrafted features for nuclei cells. We further build a framework to predict the tumor types based on the learnt cell features. We present numerous experimental results to show the benefit of using subtype nuclei cell features to represent as nodes in cell-graph and the advantage of the proposed for improving the performance to predict the tumor types/tumor grades.

Overall, our main contribution in this paper is as follows:

- We propose a novel transformer based instance segmentation framework to classify different types of nuclei cells more accurately.
- We propose a Graph based Attention Multiple Instance learning framework for histology images which builds a cell-graph based on nuclei cell features to effectively represent the histology images to perform the downstream task.
- Our extensive experiments shows the need of using the subtype cell-interaction for cancer staging/tumor classification and Graph Attention Multiple Instance Learning framework to improve the performance for tumor classification.

4.2 Related work

4.2.1 Nuclear instance segmentation and classification

Several automated methods have been proposed to segment and classify nuclei cells from H&E stained images. Before the era of deep learning, popular methods were based on thresholding and performing morphological operations to segment the nuclei cells [61, 62]. These methods fail to generalize or accurately segment the nucleus. Recently, Deep neural networks have shown promising results in segmenting and classifying nuclei cells in a more complex background [49, 63, 64]. However, automatically segmenting nuclei cells at the instance level still remain challenging due to the following reasons. Firstly, nuclei cells tend to be vastly occluded which makes the automated methods difficult to segment at the instance level leading to poor morphological measurements. Secondly, due to the subtle morphological difference between several types of nuclei cells, the automated methods have difficulty in classifying different types of nuclei cells. To address the first issue, hover-net [49] proposed an architecture to measure the horizontal and vertical distances of nuclei to their center of mass. These distances are used to segment occluded nuclei cells. BPR-Net [65] proposed an architecture to segment and refine its boundary in parallel. These proposed methods are computationally expensive and still suffer from overlapping issue. To address the second issue, on top of instance segmentation [49] performs pixel-level classification to identify different types of nuclei. Cell-RCNNV3 [63] uses Mask-RCNN inspired classification head and panoptic segmentation to segment and classify nuclei cells. However, due to the poor discriminative features, these models produce low precision. To overcome the aforementioned issues, our proposed segmentation model is light-weight and produces higher performance when compared to the state-of-the-art methods in terms of both segmentation and classification.

4.2.2 Tumor subtyping

Various automated methods have been proposed to categorize histology images into several types based on the appearance of tumors in the WSIs. Due to the expensive cost of labelling the in WSI, methods have been developed to use weak labels such as image-level class label. In general, methods rely on patch-wise feature extraction and then aggregating the features to perform the downstream task. In [66], is used on the features extracted by CNN to classify different tissue types. In [55], Context aware feature aggregation module is used to aggregate the features extracted from CNN. These proposed methods rely on compressing the image patch into a latent vector, which loses granular information related to the patch. Recently, methods have been proposed to include granular information to predict the tumor subtypes. CGC-Net [10], uses morphological features of nuclei cells instead of patch-

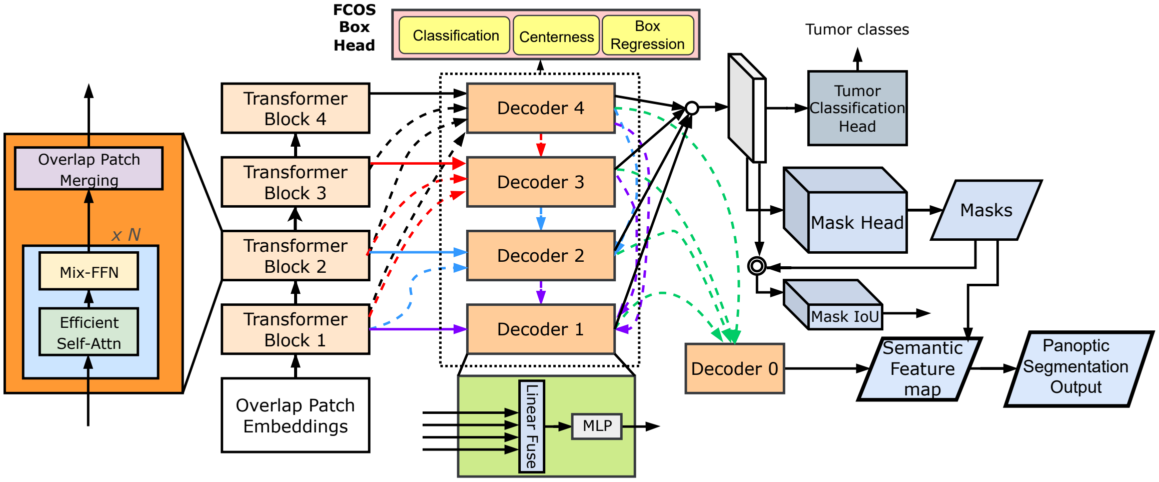


Figure 4.1: Overview of our proposed segmentation framework (TranSeg). The image is first split into patch embeddings and then passed to several transformer blocks. The output of decoder blocks are used for FCOS loss. The proposals are computed and extracted from decoder outputs which are used to output segmentation mask.

wise features to predict the tumor grades given a WSI. Further, [59] proposed a statistical network analysis method to describe the complex network structure of tissue micro-environment for image classification. HACT-Net [58], improved over CGC-Net by combining hierarchically the granular nuclei cell features and patch-wise features to predict different types of cancer from breast tissues.

Our proposed method differs from these methods in two ways. First, the proposed method uses subtype nuclei cell CNN features rather than morphological/handcrafted features, thereby capturing a deep latent vector which empirically shows a better performance than previous methods. Second, it uses multiple graph representations to describe a WSI and use MIL to predict the tumor types.

4.3 Proposed Method

In this section, we explain the proposed framework for segmentation and tumor grade classification respectively. Figure 4.1 and figure 4.4 summarizes the segmentation and classification architecture respectively.

4.3.1 Segmentation Input

The input to our segmentation framework is an image X from dataset $D = \{X^i, Y^i\}; i = 1, 2, ..., K$, where K is the total number of images in the training dataset. Each Y^i contains labels from $1, 2, ..., C$, where C is the total number of classes. The goal of the segmentation model is to input the image X^i and output a segmentation output with the same image size as X^i where each pixel is labeled into one of the C classes.

4.3.2 Segmentation Encoder

Segmentation Encoder used in our framework consists of a hierarchical transformer to generate coarse-grained to fine-grained features. Given an image size of $H \times W \times 3$, we divide the image size into patches of 4×4. We use the overlapping patches into a hierarchical transformer to output multi-resolution feature maps of stride $\{1/4, 1/8, 1/16, 1/32\}$. We use Encoder which is inspired from SegFormer [67]. We use MIT-B3 as the encoder due to its complexity and fast inference. Each hierarchical transformer consists of N blocks of Efficient Self-Attention module and Mix-Feed Forward Network. The output features are then merged using Overlap Patch Merging to produce features with the same size as the non-overlapping process. The Efficient Self-Attention layer reduces the multi-head self-attention process time complexity $O(N^2)$ to $O(\frac{N^2}{R})$, where R is a hyper-parameter set across different stages from stage-1 to stage-4. The multi-head attention with heads Q, K, V having the dimensions $N \times C$, where $N = H \times W$ is the length of the sequence with C features is written as:

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d_{head}}})V, \tag{4.1}$$

The efficient self-attention used in [67] reduces the sequence process as follows:

$$\begin{aligned} \hat{K} &= Reshape(\frac{N}{R}, C \cdot R)(K) \\ K &= Linear(C \cdot R, C)(\hat{K}), \end{aligned} \tag{4.2}$$

Where K is the sequence to be reduced and $Reshape$ function reshapes K to $(\frac{N}{R}, C{\cdot}R)$. The result K has the dimensions $\frac{N}{R} \times C$. The Mix-FFN is specifically designed for segmentation which alleviates the problems faced by in [67]. The Mix-FFN uses a 3×3 convolution to provide positional information for Transformers and is written as follows:

$$x_{out} = MLP(GELU(Conv_{3\times3}(MLP(x_{in})))) + x_{in}, \tag{4.3}$$

where x_{in} is the output from self-attention module, MLP is a linear function, $Conv_{3\times3}$ is the 3×3 convolution and $GELU$ is the activation function [68].

4.3.3 Light-weight Feature Pyramid Networks

The goal of Feature Pyramid Networks is to use the multi-resolution feature maps generated from the segmentation encoder to have semantics from low to high levels. For this, we build a pyramid hierarchy network where each pyramid has features from all multiple levels. The dotted lines in Figure 4.1 shows the multi-level feature aggregation at each pyramid level. The feature pyramids which we name from Decoder1-Decoder4, at any point of time takes features from multiple resolutions to output the resolution of the same resolution. To do this, we scale the feature maps to the corresponding feature scale and then concatenate the feature maps from multiple resolutions. We then use a to merge the concatenated features to output a feature map of the desired size.

4.3.4 FCOS

To perform instance segmentation, our framework is required to propose bounding boxes which are used to predict segmentation output. Most object detectors such as Faster-RCNN [69], YOLO [70] use pre-defined anchor box which needs hyper-parameter tuning. To eradicate the need for hyper-parameter tuning, we use FCOS [71], which directly predicts bounding boxes and classification labels. In this way, we reduce the complexity in computing anchor boxes which reduces memory consumption and computation cost. Further, FCOS uses the "center-ness" branch to find the deviation of pixel to the center of its corresponding bounding box which improves the detection performance.

4.3.5 Occlusion Aware Instance segmentation

One of the important problems in nuclei cell segmentation is handling the overlapping nuclei cells. We handle the overlapping nuclei cells by decoupling overlapping objects in the same RoI into two distinct modules (occluder-occludee) where occludee is segmented under the guidance from shape and location of the occluder. The architecture of occluder-occludee is shown in figure 4.2(b) which is inspired from [72]. The input to the segmentation branch is a cropped RoI feature maps that are generated from Adaptive Feature Pooling [73]. The reason for using the Adaptive Feature Pooling network is to use features from multi-resolution feature pyramid levels. In FPN [74], feature pyramid level is selected based on the size of the proposal box. However, the size of nuclei cells makes the level selector to always select from the lowest level. By using Adaptive Feature Pooling network, we allow the network to use all levels for each proposal and fuse them for downstream task. The occluder branch from figure 4.2(b) uses layer followed by two FCN to detect the occluder's mask and contour. The occluder's features are used as guidance to detect the occludee's mask and contour. The occludee's feature branch is the same as occluder.

The module of occluder is designed similarly to [72], where we have one 3 × 3 convolutional layer followed by a and a FCN. The output of the FCN is passed to an up-sampling layer and a 1 × 1 convolutional layer to jointly output boundary and segmentation mask respectively. The occludee follows the similar architecture as occluder with addition to the output from FCN in occluder branch. The input to the occluder and occludee branch has feature map the size of 14×14 and the output has the size of 28×28.

We further improve the mask quality predicted from the occludee branch by using a mask quality branch. The segmentation output from occludee branch $1 \times 28 \times 28$ is concatenated with the RoIs of size $256 \times 14 \times 14$ by performing max pooling operation on the segmentation output. Mask quality branch contains 3 convolutional layers with kernel size 3×3 followed by 3 fully connected layers to predict the quality of mask, which is a float value in $(0, 1)$. We use the dice score between the predicted segmentation mask from occludee and ground truth segmentation mask as the ground truth mask quality score. The mask quality score is defined as:

$$S_{gt} = 2 * \frac{|M_p \cap M_t|}{|M_p| + |M_t|} \tag{4.4}$$

where M_p is the predicted segmentation mask and M_t is the ground truth mask. We use $l2$ loss between S_{gt} and predicted mask quality. The predicted mask quality is used during the inference stage to filter the masks.

4.3.6 Classification Head

In general, Classification head used in object detectors use based heads [69] where is performed on the . However, when the classes have a subtle difference between them and the power of becomes sub-optimal. The based module loses location, sizes and shape features due to the average operation performed on the . As shown in figure 4.2(a), we use cross-attention based classification head for subtype nuclei cell classification that implicitly learns the shape, location and inherent features representing the nuclei cells.

$$\begin{aligned} G_{q1} &= MultiHeadAttn(G_q, E, E) \\ G_{q2} &= FF(G_{q1}) \\ Logits &= GroupFC(G_{q2}) \end{aligned} \tag{4.5}$$

Where $MultiHeadAttn$ is the cross-attention used from [75] with G_q as the input group queries, E as the image embeddings, FF is the feed-forward layer and $GroupFC$ is the grouped fully connected layer proposed in [76]. The input to the classification head is the with feature map size $256 \times 7 \times 7$ and the output of the classification head is $N \times C$, where N is the number of proposals in an image and C is the number of classes.

4.3.7 Panoptic Instance segmentation

The instance segmentation framework produces segmentation masks from the boxes proposed by FCOS module. However, FCOS might generate or . To overcome this, we propose a feature fusion mechanism to incorporate semantic level features with instance level features.

We fuse the instances predicted with the semantic features following the algorithm **??**, where *resize* is a function to scale each instance mask to its corresponding predicted width and height, *normalize* is a function to normalize the output semantic feature map so that the sum of probabilities across classes equals 1.

4.3.8 Segmentation Loss function

As shown in figure 4.1, total loss function required to train the segmentation model is defined as:

$$\begin{aligned} L_{seg} = {} & L_{fcos-center} + \lambda_1 L_{fcos-iou} + L_{fcos-l1} \\ & + L_{fcos-cls} + \lambda_2 L_{mask-occluder} + \\ & + \lambda_3 L_{bo-occluder} \lambda_4 L_{mask-occludee} + \\ & + \lambda_5 L_{bo-occludee} + L_{mask-qua} \\ & + \lambda_6 L_{semseg} \end{aligned} \tag{4.6}$$

For object localization task, $L_{fcos-center}$ are the binary cross entropy loss between normalized ground truth centers and predicted centers, $L_{fcos-cls}$ are the binary cross entropy loss to detect if a nuclei is present in a pixel. $L_{fcos-iou}$ are the loss and $L_{fcos-l1}$ are $l1$ regression loss between ground truth bounding boxes and predicted bounding boxes respectively.

Similarly, for training segmentation task, $L_{mask-occluder}$ and $L_{bo-occluder}$ are the binary cross entropy loss for masks and contours of overlapping nuclei cells respectively. $L_{mask-occludee}$ and $L_{bo-occludee}$ are binary cross entropy loss for the current nuclei. $L_{mask-qua}$ is $l2$ regression loss between predicted mask quality and ground

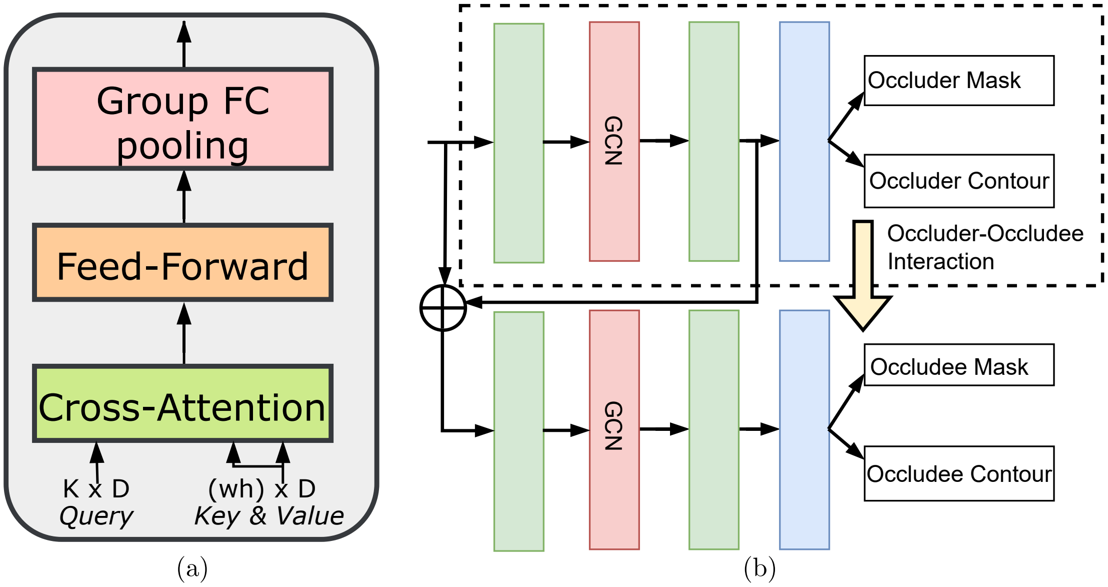


Figure 4.2: Figure (a) shows the classification head where they key and value are image embeddings and query are the non-learnable vectors. Figure (b) shows the occlusion-aware segmentation head where occluder and occludee branches interact to output occlusion free mask.

truth mask quality. Finally, L_{semseg} is the cross entropy loss between ground truth segmentation map and predicted segmentation. The weights of λ_1, λ_2, λ_3, λ_4, λ_5 and λ_6 are set to 1.25, 0.25, 0.5, 1.0, 0.5, 0.1 respectively and these hyper parameters were chosen based on heuristics.

During inference, we multiply the mask quality and bounding box score to filter the masks as final instances for an image.

4.3.9 Tumor grade classification

In this section, we discuss the tumor grade classification performed with the nuclei features extracted from TranSeg.

4.3.9.1 Cell-graph representation

We infer TranSeg on the tumor classification datasets and extract the subtype nuclei cell features along with their normalized spatial coordinates for each WSI. Each node contains 71 dimensions where 64 dimensions are extracted from TranSeg and 7 dimension represents a one-hot vector denoting each class. To create a graph topology, we utilize that spatially close cells have stronger interactions when compared to distant cells. We build the cell-graph by using the euclidean distance between nuclei centroids in the image space. Nuclei close to each other will have a distance close to 1 and distant nuclei will have a distance close to 0.

4.3.9.2 Cell-graph Architecture

For simplicity, we follow the graph architecture used in HACT-Net [58]. The graph architecture used in our framework uses PNA [77] operator within the framework of message passing neural network [78] obtaining the following GNN layer:

$$X_i^{t+1} = U\left(X_i^{(t)}, \bigoplus_{(j,i)\in E} M\left(X_i^{(t)}, E_{j\to i}, X_j^{(t)}\right)\right) \tag{4.7}$$

for a node X_i, the set of neighboring nodes X_j is concatenated and passed to a fully-connected layer M to produce a set of neighborhood-aware embeddings. Then multiple aggregators with degree scalars $\bigoplus$ is applied to a set of neighborhood embedding. The node feature X_i is then concatenated with the result and then passed to a fully-connected layer to update the node embedding. Details of $\bigoplus$ is borrowed from PNA [77]. The features from each PNA layers are aggregated and reduced to a latent vector using LSTM module [58].

4.3.9.3 Graph Attention Multiple Instance Learning

We create multiple graphs, where each graph contains nuclei cells extracted using [10] as the nodes. The method chooses subset of nuclei, where each nuclei has the farthest distance to the selected nuclei collection. Each graph generates a h_i, where $i = 1, 2, ..., K$ with K is the total number of graphs. We then merge h_i by following equation 4.8 followed by a fully-connected layer to predict $N \times C$ where N is the batch size and C is the number of classes. Figure 4.4 shows the architecture of Cell-Graph and MIL combined together.

$$z = \sum_{i=1}^{K} \alpha_i h_i, \tag{4.8}$$

$$\alpha_i = \frac{exp\{w^T tanh(V h_i^T)\}}{\sum_{j=1}^{K} exp\{w^T tanh(V h_j^T)\}}, \tag{4.9}$$

4.4 Datasets & Performance Measures

In this section, we explain the dataset details used for nuclei subtype cell segmentation and tumor classification. We then explain the performance metrics used for evaluation.

4.4.1 Datasets

We evaluate the segmentation framework on the lizard dataset [2] which contains colon cancer. The dataset is collected from 6 different sources which contains image regions extracted from WSIs. The colon image regions are sampled from original data sources at $20\times$ objective magnification. The dataset contains $495,179$ nuclei with $101,413$ lymphocytes, $28,466$ plasma cells, $4,824$ neutrophils and $3,604$ eosinophils, $112,309$ connective cells and $244,563$ epithelial cells. The image patch size used for training and testing is 256 × 256 pixels. We split the dataset into 4 fold cross validation with 70% training and 10% validation and 20% testing dataset. We report the average of 4 fold cross validation in table 4.1.

We evaluate the tumor classification framework on two challenging datasets. First, we use [58] which contains 2080 RoIs acquired from 106 H&E stained breast carcinoma WSIs. The WSIs are scanned at 0.25 um/pixel for 40× magnification. The RoIs are annotated as Normal, Benign, Atypical, Ductal carcinoma, in situ and Invasive. The dataset contains 305 Normal, 462 Benign, 387 Atypical, 503 Ductal carcinoma, 423 Invasive. To have fair comparison, we use 4 fold cross validation similar to [58]. We also evaluate on another dataset that is focused on colon cancer [46]. The dataset contains 139 images with an average size of 4500 × 7500 pixel size taken at 20× magnification. Each RoI is classified into one of the three classes (Normal, low-grade, high-grade). The dataset contains 71 Normal, 33 low-grade and 35 high-grade RoIs. We split the dataset into three folds and use the entire RoI for cell network construction.

4.4.2 Performance measures

We discuss the performance measures used for subtype nuclei cell segmentation and tumor tissue classification. For fair comparison, we use binary Dice score to measure the separation of all nuclei from background, AJI, and multi-class for instance segmentation. For measuring tumor tissue classification, we use weighted F1-scores and accuracy.

4.5 Implementation Details

To train the segmentation method, we use detectron2 codebase built on top of PyTorch [44]. We first normalize each image with the mean and standard devia-

Model	DICE	AJI	PQ	multi-PQ
Cell profiler	62.3 ± 8.2	37.6 ± 5.4	31.2 ± 4.9	23.5 ± 5.6
U-Net	77.4 ± 5.2	54.1 ± 7.8	52.9 ± 4.5	33.2 ± 4.3
Hover-Net	81.3 ± 3.2	60.2 ± 3.9	58.2 ± 3.8	44.5 ± 4.1
CIA-Net	82.3 ± 3.3	60.8 ± 3.8	57.3 ± 3.6	42.1 ± 3.9
CD-Net	83.5 ± 2.9	63.4 ± 3.4	61.1 ± 3.4	46.2 ± 3.8
Mask-RCNN	81.2 ± 4.1	58.3 ± 3.5	55.8 ± 4.1	43.1 ± 3.9
Cell-RCNNV3	83.4 ± 3.8	61.2 ± 3.8	60.8 ± 3.1	44.1 ± 3.0
BRP-Net [65]	85.1 ± 3.1	64.9 ± 3.2	62.8 ± 2.9	46.9 ± 2.1
TranSeg	**87.2** ± 2.7	**70.1** ± 2.3	**69.9** ± 1.3	**50.1** ± 0.9

Table 4.1: Comparative results on Lizard dataset [2].

tion from the training dataset. We then augment the training dataset with random cropping, flipping, color jittering, random copying nuclei cells as shown in figure and mosaic augmentation as shown in figure. As shown in figures 4.3 (a) and (b), adding mosaic and random "copy and paste" of nuclei cells significantly improves the performance. For copy and paste, we paste the nuclei cells to balance the class imbalance. All images are cropped to size 256 × 256 before using them as input to segmentation model. We use the AdamW [79] optimizer for training. The number of training epochs is set to 600 epochs. The initial learning rate is set to 0.0001 and "Poly" learning rate schedule with a decay factor as 0.1. The segmentation model is trained on server 4× RTX1080 Ti with batch size of 16.

To train , we run TranSeg on two datasets and extract subtype nuclei cell features for each nuclei and its corresponding spatial coordinates. We use 7 graphs and each graph has 500 subtype nuclei cells extracted. We perform this operation after every 10 epochs to make sure that the observes different graphs.

Model	F1	Normal	Benign	UDH	FEA	ADH	DCIS	Invasive
CA-CNN	52.8 ± 1.9	50.3 ± 0.9	44.3 ± 1.2	41.3 ± 2.4	31.6 ± 3.3	51.6 ± 3.0	57.3 ± 0.9	86.0 ± 1.4
Patch-GAMIL	59.7 ± 1.3	55.5 ± 0.7	46.3 ± 1.5	44.7 ± 2.1	39.1 ± 3.0	68.2 ± 2.8	61.8 ± 1.2	84.1 ± 1.2
CGCNet	43.6 ± 0.5	30.8 ± 5.3	31.6 ± 4.6	17.3 ± 3.3	24.5 ± 5.2	58.9 ± 3.5	49.3 ± 3.4	75.3 ± 3.2
CG-GNN	55.9 ± 1.0	58.7 ± 6.8	40.8 ± 3.0	46.8 ± 1.9	39.9 ± 3.5	63.7 ± 10.4	53.8 ± 3.8	81.0 ± 3.3
HACT-Net	61.5 ± 0.8	61.5 ± 2.1	47.4 ± 2.9	43.6 ± 1.8	40.4 ± 2.5	74.2 ± 1.4	66.4 ± 2.5	88.4 ± 0.1
GAMIL w/o SNC	60.2 ± 1.2	**64.5** ± 3.9	46.5 ± 2.3	45.2 ± 2.4	43.9 ± 1.6	72.1 ± 1.2	65.7 ± 1.8	86.3 ± 0.8
GAMIL	**63.4** ± 0.5	63.6 ± 2.1	**49.4** ± 2.1	**48.2** ± 2.1	**45.9** ± 1.1	**77.5** ± 0.9	**68.5** ± 1.3	**89.9** ± 0.8

Table 4.2: BRACS: Mean of weighted F1 scores across four folds and for each class across four folds. GAMIL w/o SNC refers to out proposed method without using subtype nuclei cell features. Results are shown in %.

4.6 Experimental Results

4.6.1 Segmentation Results

For segmentation, we compare our proposed method with both semantic segmentation and instance segmentation. For fair comparison, we perform the same data augmentation on all the competitor methods. We perform 4 fold cross validation and report the average results across 4-folds. All competitor methods where tuned to perform best on the lizard dataset [2]. **Cell Profiler** is a software based cell analysis method, which uses thresholding and series of post processing methods. **U-Net** [24] is a semantic segmentation model where each pixel predicted belongs to one of the classes. **HoVer-Net** [49] is a semantic segmentation model, which predicts horizontal and vertical gradient maps to separate the neighbouring boundaries. **CD-Net** [64] unlike HoVer-Net, predicts centripetal directions from the center of nucleus. **CIA-Net** [80] uses a contour-aware mechanism to separate the neighbouring nuclei cells. In the instance segmentation framework, we use **Mask-RCNN** [81] which is a two stage pipeline, where pre-defined anchors are generated in the first stage and segmentation is performed on the later stage. **Celll-RCNNV3** [28], similar to Mask-RCNN, uses panoptic attention based merging to use the semantic and instance level capabilities. In **BRP-Net** [65], boundary and segmentation masks are predicted in parallel and fused to segment overlapping nuclei cells. Finally, Our proposed TranSeg uses transformer based encoder and classification head to predict the nuclei cell classes for each proposed nuclei. From table 4.1 shows the overall improvement in all metrics when compared to the competitor methods. We also show the box-whisker plot in figure 4.5, where TranSeg results in a skewed distribution when compared to competitor methods.

4.6.2 Classification Results

For tumor grade classification, we compare our proposed method with methods that use only tissues, only nuclei cells and both tissues and nuclei cells as features. Table 4.2 shows the comparative results of BRACS dataset and table 4.3 shows the results of CRC dataset. We compare the proposed method with the following competitor methods and tune the hyper-parameters to achieve the best performance for each method.

CA-CNN [55] is a patch based classification method which uses context-aggregation module. **Patch-GAMIL** [53] is a patch based classification method, where multiple graphs are constructed with random patches of size 224×224. Patch-GAMIL uses Multi Instance Learning to classify the tumor class. **CGC-Net** [10] is a Cell Graph CNN. We construct the graph with nuclei cell hand-crafted features extracted using HoVer-Net. **CG-GNN** [58] is a Cell Graph with GNN as the graph architecture unlike CGC-Net, where Adaptive GraphSage is used as the architecture. **HACT-Net** [58] combines Tissue Graph and Cell graph in a hierarchical approach.

For our proposed method, first we do not use the subtype nuclei cell features extracted from TranSeg which we call GAMIL w/o SNC. Rather we use nuclei cell features extracted from HoVer-Net, similar to HACT-Net. Finally, we use GAMIL with subtype nuclei cell features extracted from TranSeg. For BRACS dataset, in table 4.2 shows that improves the state-of-the-art HACT-Net by over 1.9% and for CRC dataset, improves by over 1.8%.

4.6.3 Computational time and memory analysis

We report the computational time taken to run TranSeg on a 1000×1000 image tile over 10 runs with other competitor methods. We use 1× RTX 1080Ti with 12 GB RAM to report the time complexity. The time taken by Mask-RCNN for segmentation and classification is 103.4 seconds. We also report the time complexity of HoVer-Net, which is 11.04 seconds. Meanwhile, our proposed method (TranSeg) takes 11.98 seconds which is similar to Hover-Net. The reason for having approximately 10× faster computation than Mask-RCNN is that we do not use Anchor based approach to generate proposal boxes. We also use a simple encoder and decoder architecture when compared to HoVer-Net, which allows our proposed method to perform similar to HoVer-Net. We also emphasize that, our model generates nuclei cell latent features that HoVer-Net does not generate. The number of learnable parameters consumed by HoVer-Net on a 256×256 image is 52.84 million parameters whereas, our proposed method consumes 61.26 million parameters.

The time complexity of a general non-sparse graph is $O(n^2)$ [77] where n is the number of nodes in the graph. In our proposed we consider multiple subsets of graphs which make the time complexity $k \cdot O(m^2)$, where $m \ll n$. Time taken to run is 2.18 seconds.

Model	Acc	Normal	Low-grade	High-grade
CA-CNN	95.1	97.4	93.2	96.2
Patch-GAMIL	97.1	97.9	95.6	97.2
CGCNet	95.9	97.8	94.2	95.8
HACT-Net	97.3	98.2	96.2	96.8
GAMIL w/o SNC	97.2	97.3	95.8	96.9
GAMIL	**99.1**	**99.2**	**98.4**	**99.4**

Table 4.3: CRC: Mean of Accuracy scores across four folds and for each class across four folds. Acc refers to Accuracy. Results are shown in %.

4.7 Ablation studies

4.7.1 Need for subtype nuclei cell features

We first look into the need for subtype nuclei cell features for classifying different tumor tissue types. To do this, we use our subtype nuclei cell features directly into the architectures which were built based on nuclei cells. For this, we use HACT-Net [58], CGC-Net [10], CG-GNN [58] by replacing their nuclei cell features by our subtype nuclei cells features generated from the classification head of the segmentation network. Table 4.4 shows the performance improvement when subtype nuclei cell features are used as node in the cell-graph of their respective architectures in BRACS dataset.

4.7.2 Classification Head

We investigate different classification heads to empirically find the need for cross-attention based classification head. We compare with , and ML-Decoder classification heads. The difference between ML-Decoder and is the removal self-attention layer and non-learnable queries in ML-Decoder. In table 4.6, we see that GAP uses no

Model	weighted F1 w/o SNC	weighted F1 w SNC
HACT-Net	61.5 ± 0.8	**62.9** ± 2.5
CGC-Net	43.6 ± 0.5	**50.1** ± 1.8
CG-GNN	55.9 ± 1.0	**59.2** ± 2.1

Table 4.4: Need for subtype nuclei cell features on BRACS dataset. SNC refers to subtype nuclei cell features.

Segmentation Head	PQ	Params (Millions)
Mask-RCNN	59.12 ± 3.98	1.86
Cell-RCNNV3	64.29 ± 4.1	2.56
BRP-Net	62.67 ± 3.76	10.72
Occlusion-aware	**68.12** ± 3.45	0.56

Table 4.5: Ablation study on segmentation heads on lizard dataset.

additional learnable parameters and ViT [82] uses 9.56 million parameters which is 10× more than Ml-Decoder which is used as our classification head. ML-Decoder reduces the number of parameters as well as improves the multi-class on lizard dataset.

4.7.3 Segmentation Head

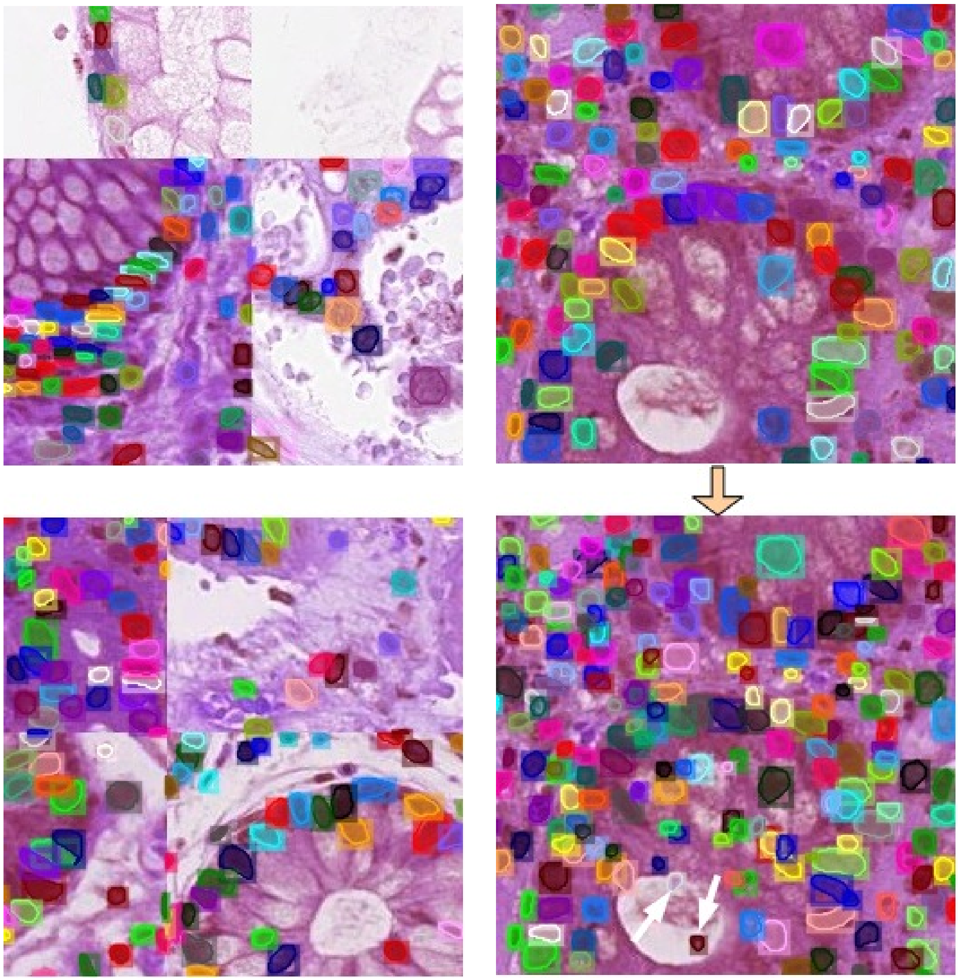
Classification Head	multi-PQ	Params (Millions)
	43.12	0
[82]	45.76	9.56
ML-Decoder	**49.71**	0.28

Table 4.6: Ablation study on classification heads on lizard dataset.

We explore different segmentation heads used in nuclei cell instance segmentation. We compare with Cell-RCNNv3 [28], Mask-RCNN [81], BRP-Net [65] with occlusion aware segmentation head. To have a fair comparison, we use the segmentation head from the competitor methods into our segmentation framework keeping other components fixed. We use multi- metric to measure the performance as we are interested in how the segmentation head performs on overlapping nuclei cells. Occlusion aware segmentation head adds a little overhead on the computation but significantly improves multi- over the competitor methods.

4.8 Conclusion

In this work, we show that by using subtype nuclei cell features improves the tumor grade classification performance. To do this, we have proposed a subtype nuclei cell segmentation framework that can accurately segment overlapping nuclei cells and classify each nuclei cell into different subtypes more accurately. We have extensively evaluated the proposed segmentation method with state-of-the-art segmentation methods and show the significant performance improvement mainly in multi-PQ metric. We have also proposed a Graph based Attention Multiple Instance Learning method that uses subtype nuclei cell features and evaluated on tumor grade classification datasets. Using , we show that tumor grade classification performance increases when compared with state-of-the-art classification methods.



(a) (b)

Figure 4.3: We show random mosaic data augmentation in (a), where random patches from 4 images are clubbed together to form a new image and random ”copy and paste” of nuclei cells in (b).

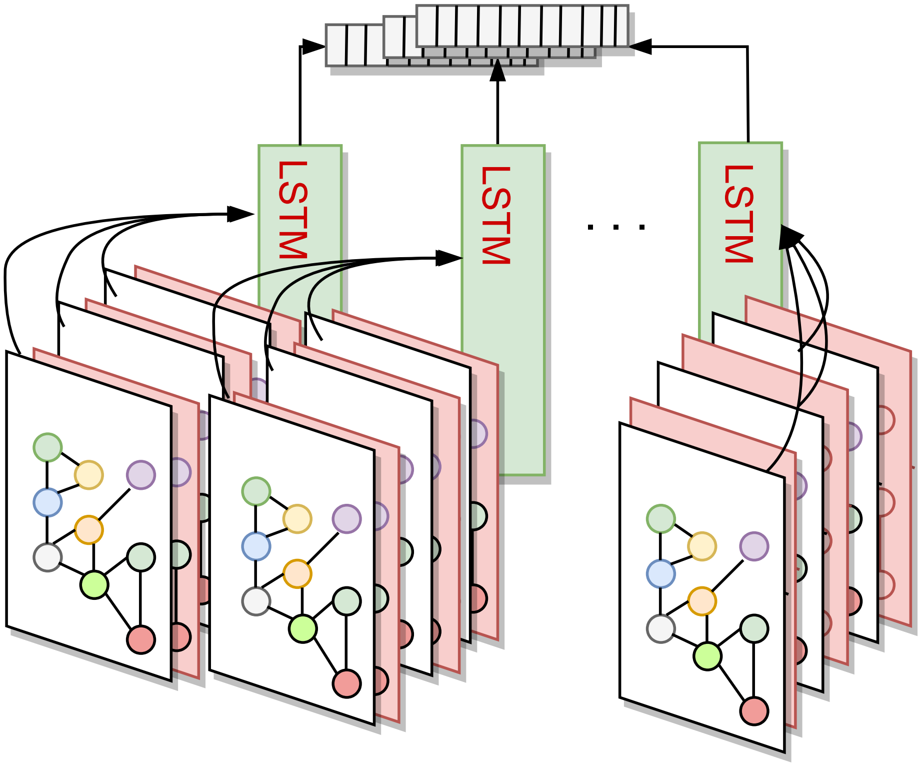


Figure 4.4: :Each graph contains random sampling of nuclei cell features and each graph outputs a embedding vector from LSTM. The embedding vectors from each graph is used to predict the class output.

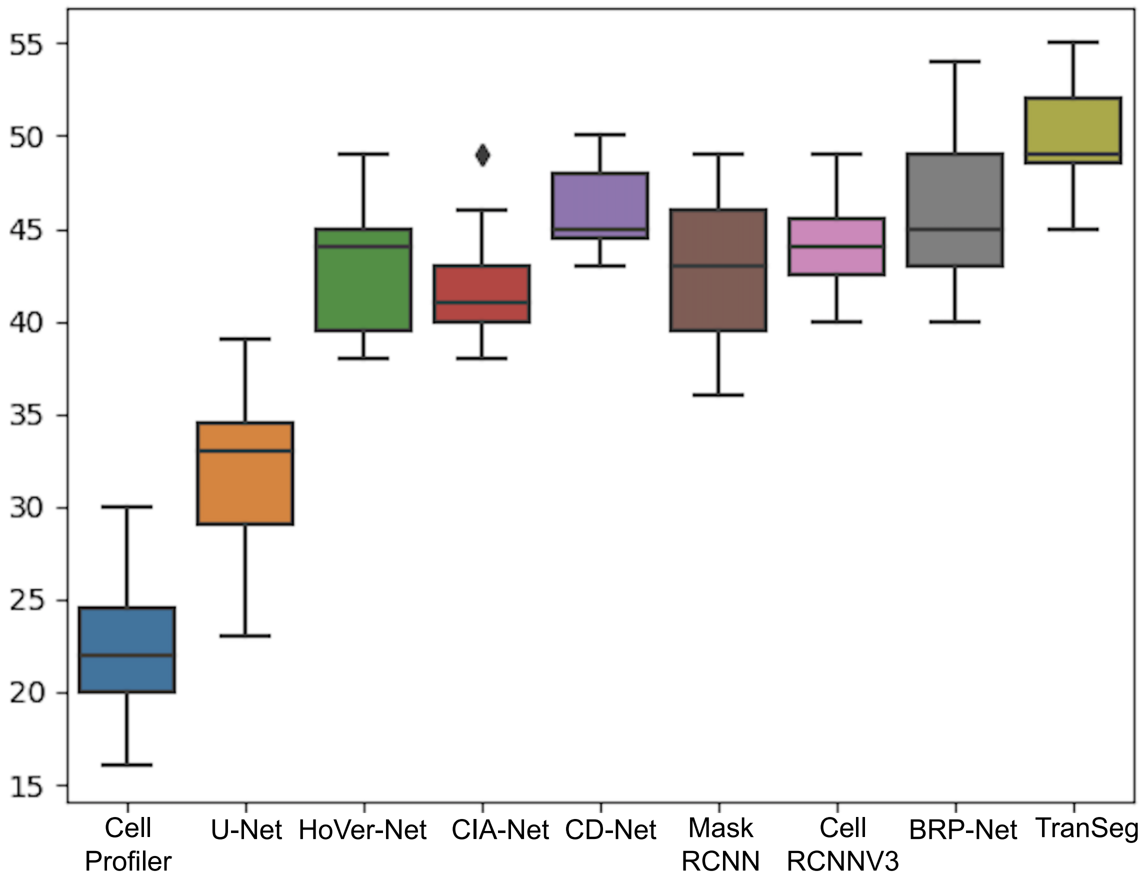


Figure 4.5: Box-whisker plot of multi-PQ on the lizard dataset.

CHAPTER 5

Concluding Remarks and Future Directions

5.1 Concluding Remarks

This dissertation focuses on improving the performance of cancer staging from histology images. To this end, we are the first to our knowledge to use granular information to perform cancer staging. To this end, we observe and conduct experiments on using sub type cell features as granular information to predict the cancer stage. We have focused our research on ways to extract, gather and expand the sub type nuclei cell dataset.

To do this, we have divided the experimental study into three sections. In the first section, we investigate the spatial relationship of the tissue to understand if the spatial relationship helps. In the second section, we create a framework to use shape constraint and segment the nuclei cells which resembles the shape of the training dataset. In the third section, we create a framework to use the extracted nuclei cell features and use the spatial relationship to build a graph structure that can be used to do cancer staging. We show through series of experiments that the proposed framework are used to improve the performance of the cancer staging.

5.2 Future Directions

This research shows the roadmap on how to improve the performance of cancer stages from the histology images. We have shown that by using granular information such as subtype cell information, we could improve the performance of cancer staging.

On the future directions, one could easily adapt the granular features such as subtype cell features to perform survival rate of the patient.

The survival rate of the patient refers to identifying the survival of the patients given the Whole slide images of the patient. The survival rate is a down stream task which is very similar to that of the cancer staging where granular features plays a very important cue to improve the performance. One could easily replace the classification head of the cancer staging with that of the survival rate analysis task.

REFERENCES

[1] P. Gupta, S.-F. Chiang, P. K. Sahoo, S. K. Mohapatra, J.-F. You, D. D. Onthoni, H.-Y. Hung, J.-M. Chiang, Y. Huang, and W.-S. Tsai, "Prediction of colon cancer stages and survival period with machine learning approach," *Cancers*, vol. 11, no. 12, p. 2007, 2019.

[2] S. Graham, M. Jahanifar, A. Azam, M. Nimir, Y.-W. Tsang, K. Dodd, E. Hero, H. Sahota, A. Tank, K. Benes, *et al.*, "Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 684–693.

[3] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6210–6219.

[4] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 708–717.

[5] J. Yao, X. Zhu, and J. Huang, "Deep multi-instance learning for survival prediction from whole slide images," in *MICCAI 2019*, ser. LNCS, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., vol. 11764. Cham: Springer, 2019, pp. 496–504.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] H. Muhammad, C. S. Sigel, G. Campanella, T. Boerner, L. M. Pak, S. Büttner, J. N. M. IJzermans, B. G. Koerkamp, M. Doukas, W. R. Jarnagin, A. L. Simpson, and T. J. Fuchs, "Unsupervised subtyping of cholangiocarcinoma using a deep clustering convolutional autoencoder," in *MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer, 2019, pp. 604–612.

[8] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[9] J. Yao, J. Cai, D. Yang, D. Xu, and J. Huang, "Integrating 3d geometry of organ for improving medical image segmentation," in *MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer, 2019, pp. 318–326.

[10] Y. Zhou, S. Graham, N. Alemi Koohbanani, M. Shaban, P.-A. Heng, and N. Rajpoot, "Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[11] R. Li, J. Yao, X. Zhu, Y. Li, and J. Huang, "Graph cnn for survival analysis on whole slide pathological images," in *MICCAI 2018*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer, 2018, pp. 174–182.

[12] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," *arXiv preprint arXiv:1904.08082*, 2019.

[13] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," *arXiv preprint arXiv:1802.04712*, 2018.

[14] R. L. Ward and N. J. Hawkins, "Molecular and cellular oncology (mco) study tumour collection," *UNSW Australia*, 2015.

[15] J. Jonnagaddala, J. L. Croucher, T. R. Jue, N. S. Meagher, L. Caruso, R. Ward, and N. J. Hawkins, "Integration and analysis of heterogeneous colorectal cancer data for translational research," 2016, p. 387.

[16] D. Tellez, J. van der Laak, and F. Ciompi, "Gigapixel whole-slide image classification using unsupervised image compression and contrastive training," 2018.

[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[18] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2918–2928.

[19] A. Raju, C.-T. Cheng, Y. Huo, J. Cai, J. Huang, J. Xiao, L. Lu, C. Liao, and A. P. Harrison, "Co-heterogeneous and adaptive segmentation from multi-source and multi-phase ct imaging data: a study on pathological liver and lesion segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 448–465.

[20] C. Yang, D. Eschweiler, and J. Stegmaier, "Semi-and self-supervised multi-view fusion of 3d microscopy images using generative adversarial networks," in *International Workshop on Machine Learning for Medical Image Reconstruction*. Springer, 2021, pp. 130–139.

[21] Y. Lin, Z. Qu, H. Chen, Z. Gao, Y. Li, L. Xia, K. Ma, Y. Zheng, and K.-T. Cheng, "Label propagation for annotation-efficient nuclei segmentation from pathology images," *arXiv preprint arXiv:2202.08195*, 2022.

[22] X. Yang, H. Li, and X. Zhou, “Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2405–2414, 2006.

[23] C. Gunduz-Demir, M. Kandemir, A. B. Tosun, and C. Sokmensuer, “Automatic segmentation of colon glands using object-graphs,” *Medical image analysis*, vol. 14, no. 1, pp. 1–12, 2010.

[24] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[25] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[26] H. Chen, X. Qi, L. Yu, and P.-A. Heng, “Dcan: deep contour-aware networks for accurate gland segmentation,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 2487–2496.

[27] X. Li, Y. Wang, Q. Tang, Z. Fan, and J. Yu, “Dual u-net for the segmentation of overlapping glioma nuclei,” *Ieee Access*, vol. 7, pp. 84 040–84 052, 2019.

[28] D. Liu, D. Zhang, Y. Song, H. Huang, and W. Cai, “Panoptic feature fusion net: a novel instance segmentation paradigm for biomedical and biological images,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2045–2059, 2021.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[30] F. Zhao and X. Xie, “An overview of interactive medical image segmentation,” *Annals of the BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[31] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *International journal of computer vision*, vol. 82, no. 2, pp. 113–132, 2009.

[32] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 105–112.

[33] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3d image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2011, pp. 603–610.

[34] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 616–625.

[35] A. Raju, Z. Ji, C. T. Cheng, J. Cai, J. Huang, J. Xiao, L. Lu, C. Liao, and A. P. Harrison, "User-guided domain adaptation for rapid annotation from user interactions: a study on pathological liver segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 457–467.

[36] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3159–3167.

[37] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE transactions on medical imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[38] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, *et al.*, "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE transactions on medical imaging*, vol. 36, no. 2, pp. 674–683, 2016.

[39] N. A. Koohbanani, M. Jahanifar, N. Z. Tajadin, and N. Rajpoot, "Nuclick: a deep learning framework for interactive segmentation of microscopic images," *Medical Image Analysis*, vol. 65, p. 101771, 2020.

[40] Y. Chen, X.-H. Yang, Z. Wei, A. A. Heidari, N. Zheng, Z. Li, H. Chen, H. Hu, Q. Zhou, and Q. Guan, "Generative adversarial networks in medical image augmentation: a review," *Computers in Biology and Medicine*, p. 105382, 2022.

[41] H. Zheng, L. Lin, H. Hu, Q. Zhang, Q. Chen, Y. Iwamoto, X. Han, Y.-W. Chen, R. Tong, and J. Wu, "Semi-supervised segmentation of liver using adversarial learning with deep atlas prior," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2019, pp. 148–156.

[42] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7472–7481.

[43] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2517–2526.

[44] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[45] K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, *et al.*, "Gland segmentation in colon histology images: The glas challenge contest," *Medical image analysis*, vol. 35, pp. 489–502, 2017.

[46] R. Awan, K. Sirinukunwattana, D. Epstein, S. Jefferyes, U. Qidwai, Z. Aftab, I. Mujeeb, D. Snead, and N. Rajpoot, "Glandular morphometrics for objective grading of colorectal adenocarcinoma histology images," *Scientific reports*, vol. 7, no. 1, pp. 1–12, 2017.

[47] G. Aresta, T. Araújo, S. Kwok, S. S. Chennamsetty, M. Safwan, V. Alex, B. Marami, M. Prastawa, M. Chan, M. Donovan, *et al.*, "Bach: Grand challenge on breast cancer histology images," *Medical image analysis*, vol. 56, pp. 122–139, 2019.

[48] T. Sakinis, F. Milletari, H. Roth, P. Korfiatis, P. Kostandy, K. Philbrick, Z. Akkus, Z. Xu, D. Xu, and B. J. Erickson, "Interactive segmentation of medical images through fully convolutional neural networks," *arXiv preprint arXiv:1903.08205*, 2019.

[49] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot, "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical Image Analysis*, p. 101563, 2019.

[50] M. Khened, A. Kori, H. Rajkumar, G. Krishnamurthi, and B. Srinivasan, "A generalized deep learning framework for whole-slide image segmentation and analysis," *Scientific reports*, vol. 11, no. 1, pp. 1–14, 2021.

[51] H. H. N. Pham, M. Futakuchi, A. Bychkov, T. Furukawa, K. Kuroda, and J. Fukuoka, "Detection of lung cancer lymph node metastases from whole-slide histopathologic images using a two-step deep learning approach," *The American journal of pathology*, vol. 189, no. 12, pp. 2428–2439, 2019.

[52] X. Zhu, J. Yao, F. Zhu, and J. Huang, "WSISA: Making survival prediction from whole slide histopathological images," in *CVPR*, 2017, pp. 7234–7242.

[53] A. Raju, J. Yao, M. M. Haq, J. Jonnagaddala, and J. Huang, "Graph attention multi-instance learning for accurate colorectal cancer staging," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 529–539.

[54] Y. Su, Y. Bai, B. Zhang, Z. Zhang, and W. Wang, "Hat-net: A hierarchical transformer graph neural network for grading of colorectal cancer histology images," 2021.

[55] M. Shaban, R. Awan, M. M. Fraz, A. Azam, Y.-W. Tsang, D. Snead, and N. M. Rajpoot, "Context-aware convolutional neural network for grading of colorectal cancer histology images," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2395–2405, 2020.

[56] D. Tellez, J. van der Laak, and F. Ciompi, "Gigapixel whole-slide image classification using unsupervised image compression and contrastive training," 2018.

[57] W. H. Fridman, C. Sautès-Fridman, J. Galon, *et al.*, "The immune contexture in human tumours: impact on clinical outcome," *Nature Reviews Cancer*, vol. 12, no. 4, pp. 298–306, 2012.

[58] P. Pati, G. Jaume, L. A. Fernandes, A. Foncubierta-Rodríguez, F. Feroce, A. M. Anniciello, G. Scognamiglio, N. Brancati, D. Riccio, M. D. Bonito, *et al.*, "Hact-net: A hierarchical cell-to-tissue graph neural network for histopathological image classification," in *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis.* Springer, 2020, pp. 208–219.

[59] N. Zamanitajeddin, M. Jahanifar, and N. Rajpoot, "Cells are actors: Social network analysis with classical ml for sota histology image classification," in

International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2021, pp. 288–298.

[60] B. Ruffell and L. M. Coussens, “Macrophages and therapeutic resistance in cancer,” *Cancer cell*, vol. 27, no. 4, pp. 462–472, 2015.

[61] M. Abdolhoseini, M. G. Kluge, F. R. Walker, and S. J. Johnson, “Segmentation of heavily clustered nuclei from histopathological images,” *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[62] C. Molnar, I. H. Jermyn, Z. Kato, V. Rahkama, P. Östling, P. Mikkonen, V. Pietiäinen, and P. Horvath, “Accurate morphology preserving segmentation of overlapping cells based on active contours,” *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[63] D. Liu, D. Zhang, Y. Song, H. Huang, and W. T. Cai, “Cell r-cnn v3: A novel panoptic paradigm for instance segmentation in biomedical images,” *ArXiv*, vol. abs/2002.06345, 2020.

[64] H. He, Z. Huang, Y. Ding, G. Song, L. Wang, Q. Ren, P. Wei, Z. Gao, and J. Chen, “Cdnet: Centripetal direction network for nuclear instance segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4026–4035.

[65] S. Chen, C. Ding, and D. Tao, “Boundary-assisted region proposal networks for nucleus segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 279–288.

[66] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, and A. Campilho, “Classification of breast cancer histology images using convolutional neural networks,” *PloS one*, vol. 12, no. 6, p. e0177544, 2017.

[67] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[68] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” *arXiv preprint arXiv:1606.08415*, 2016.

[69] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[70] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[71] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[72] L. Ke, Y.-W. Tai, and C.-K. Tang, “Deep occlusion-aware instance segmentation with overlapping bilayers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4019–4028.

[73] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[74] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[75] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[76] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, and A. Noy, “Ml-decoder: Scalable and versatile classification head,” *arXiv preprint arXiv:2111.12933*, 2021.

[77] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, “Principal neighbourhood aggregation for graph nets,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 260–13 271, 2020.

[78] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International conference on machine learning.* PMLR, 2017, pp. 1263–1272.

[79] I. Loshchilov and F. Hutter, “Fixing weight decay regularization in adam,” 2018.

[80] Y. Zhou, O. F. Onder, Q. Dou, E. Tsougenis, H. Chen, and P.-A. Heng, “Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation,” in *International Conference on Information Processing in Medical Imaging.* Springer, 2019, pp. 682–693.

[81] J. W. Johnson, “Adapting mask-rcnn for automatic nucleus segmentation,” *arXiv preprint arXiv:1805.00500*, 2018.

[82] Z. Gao, B. Hong, X. Zhang, Y. Li, C. Jia, J. Wu, C. Wang, D. Meng, and C. Li, “Instance-based vision transformer for subtyping of papillary renal cell carcinoma in histopathological image,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2021, pp. 299–308.

BIOGRAPHICAL STATEMENT

Ashwin Raju received M.Sc degree in computer science engineering at The University of Texas, Arlington, Texas, USA. He is currently a Ph.D degree candidate at The University of Texas, Arlington where he focuses on applying Deep learning algorithms to solve real world problems in medical imaging.

His research interests include object localization, segmentation of tumors in medical images.

E-mail: ashwin.raju93@mavs.uta.edu (Corresponding author)