Spring 2018

Epithelium detection and cervical intraepithelial neoplasia classification in digitized histology images

Sri Venkata Ravitej Addanki

EPITHELIUM DETECTION AND CERVICAL INTRAEPITHELIAL NEOPLASIA CLASSIFICATION IN DIGITIZED HISTOLOGY IMAGES

by

SRI VENKATA RAVITEJ ADDANKI

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER ENGINEERING

2018

Approved by

R. Joe Stanley, Advisor
William V. Stoecker
Kurt Kosbar

ABSTRACT

Cervical cancer is one of the most deadly cancers faced by women. It is the second leading cause of cancer death in women aged 20 to 39 years. In order to detect cancer at early stages, pathologists analyze the epithelium region from the cervical histology images. These histology images have a pre-cervical cancer condition called cervical intraepithelial neoplasia (CIN) determined by pathologists. This study deals with automating the process of epithelium detection and epithelium CIN classification in digitized histology images. For epithelium detection, the objective is to detect epithelium regions in microscopy images from non-epithelium regions and background. convolutional neural networks, both shallow and deep networks are used for epithelium detection. The highest epithelium detection accuracy of 98.84% is obtained using transfer learning on VGG-19 architecture, pre-trained on the ImageNet dataset. For CIN classification, the epithelium region is divided into 5 segments along the medial axis and patches from each segment were used for training the deep learning model. Vertical segment level classification probabilities from deep learning model are obtained and further classified using SVM, LDA, MLP, logistic and RF classifiers. The highest image level accuracy obtained is 77.27% for MLP classifier using voting.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. R. Joe Stanley, for providing me the opportunity to work in his lab and allowing me to pursue my research interests. I am grateful for his continuous support, funding, and guidance throughout my graduate program. I would like to thank Haidar Ali Mubarak for sharing his ideas and helping me out when I hit a roadblock. I express my deep gratitude for the time he spent with me during my graduate program. I would also like to thank my committee members Dr. Kurt Kosbar and Dr. William V. Stoecker.

I would like to thank my mother K .N Lakshmi and my family for the moral and financial support throughout my graduate program.

The National Library of Medicine (NLM) supported this research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

1. INTRODUCTION

Cervical cancer is one of the most deadly cancers faced by women; in fact, it is the second leading cause of cancer death in women aged 20 to 39 years. The number of cases reported in 2017 is 12,820 [1]. Papanicolaou (Pap) test is used for screening cervical cancer and its precursor lesions. In this test, a biopsied cervical tissue histology slides are used to estimate the extent of cancer. These histology images are interpreted by expert pathologists [2]. Pathologists seek to detect cervical intraepithelial neoplasia (CIN), which is a pre-malignant condition for cervical cancer. Intra-epithelium region is the region of interest for pathologists for determining the CIN grades. A cervical biopsy is classified as normal (no CIN lesion) or one of three CIN grades: CIN1 (mild dysplasia), CIN2 (moderate dysplasia), or CIN3 (severe dysplasia) by identifying the atypical cells in the epithelium by the visual inspection of histology slides [3]. So far this epithelium detection is done manually by pathologists in large svs histology slides. The dimensions of the images (height x width) obtained from these slides are typically in the range of >20000x30000 and < 90000x70000. Figure 2.2. shows a cervical histology slide at 0.4x zoom and resized for display purpose. Due to these large sizes of the images, detecting the intra-epithelium is time-consuming as the slides often come with zoom levels of 20x and the area of the intra-epithelium region is very less when compared to the non-epithelium and white background, which is usually associated with these images. The research presented here is about automating this tedious process, which in turn simplifies the workflow for epithelium CIN discrimination. Section 2 presents research for epithelium detection and Section 3 provides algorithms developed for epithelium CIN classification cervical cancer classification. Figure 2.3. shows a block which contains an intra-epithelium region. Figure 3.6. shows the intra-epithelium segmented from the block and further divided into 5 segments along the medial axis for CIN classification. Figure 3.1. shows an example of different CIN grades from these segmented images.

2. EPITHELIUM DETECTION IN CERVICAL HISTOLOGY IMAGES

2.1. OVERVIEW

In this Section, research is presented for epithelium detection from digitized cervical histology slide images. Conventionally, the epithelium region is manually cropped and segmented from histology slide images, which has been the basis for many studies [16], [17], [18], and [19]. This research presents a novel approach for detecting the epithelium in histology slide images that can be used to extract the epithelium region, which would eliminate the epithelium region-cropping step required in previous studies [16], [17], [18], and [19]. The algorithm for epithelium detection from digitized histology slide is shown in Section 2.3.

2.2. RELATED WORK

Convolutional neural networks (CNN) are very efficient in object recognition/classification, segmentation. The success of these CNN can be seen in the large-scale image and video recognition [4][5][6] on Image Net [7] challenge. These CNN feed on large data and identify low-level to high-level features from the images. The advent of graphical processing units (GPU) has facilitated the cause. Before the advent of CNN, features are extracted manually and traditional machine learning algorithms like support vector machines (SVM), logistic regression, random forest (RF) etc., are applied for the classification purposes. The main issue here is that different domains need specific handcrafted features. CNN does not need features to be extracted manually, instead, they use filters and convolve them with the images to extract the features. The weights of filters are updated during the training process. Deep CNN [6] have been successful in further improving the accuracy. Apart from object recognition, the CNN has been successful in the biomedical Images. In [8], various use cases are given for biomedical images like nuclei and epithelium segmentation, mitosis in cells using CNN, specifically AlexNet[4]. This research explores methods to detect the intra-epithelium using various CNN architectures, comprising of both shallow and deep CNN.

2.3. EPITHELIUM DETECTION ALGORITHM

The algorithm for epithelium detection is presented in Figure. 2.1. The steps of this algorithm are presented in greater detail in the following sections.

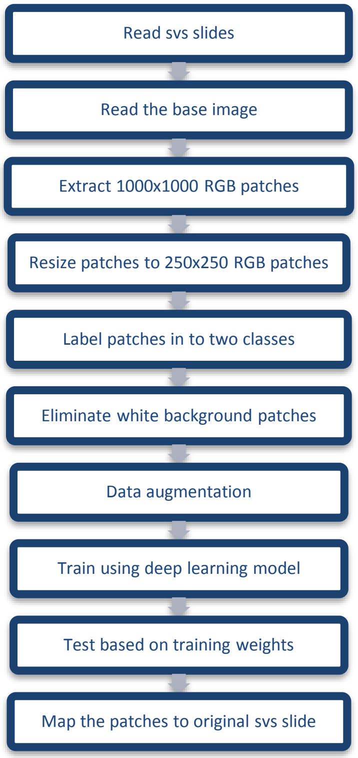
Read svs slides
Read the base image
Extract 1000x1000 RGB patches
Resize patches to 250x250 RGB patches
Label patches in to two classes
Eliminate white background patches
Data augmentation
Train using deep learning model
Test based on training weights
Map the patches to original svs slide

Figure 2.1 Epithelium detection algorithm

2.4. PREPROCESSING

Preprocessing of data is crucial in biomedical image processing. In this work, the dataset is created by extracting the patches from the raw scan scope virtual slides (svs). Most of the preprocessing for this work is performed using sklearn [9] libraries. Open slide [10] library is used to read the images from the svs slides. Dimensions of images are typically in the range above 20000x30000 and below 90000x70000. For example, In Figure 2.2. a sample slide image is shown. The base image that can be obtained using this svs slide has the dimensions 38333x42064.

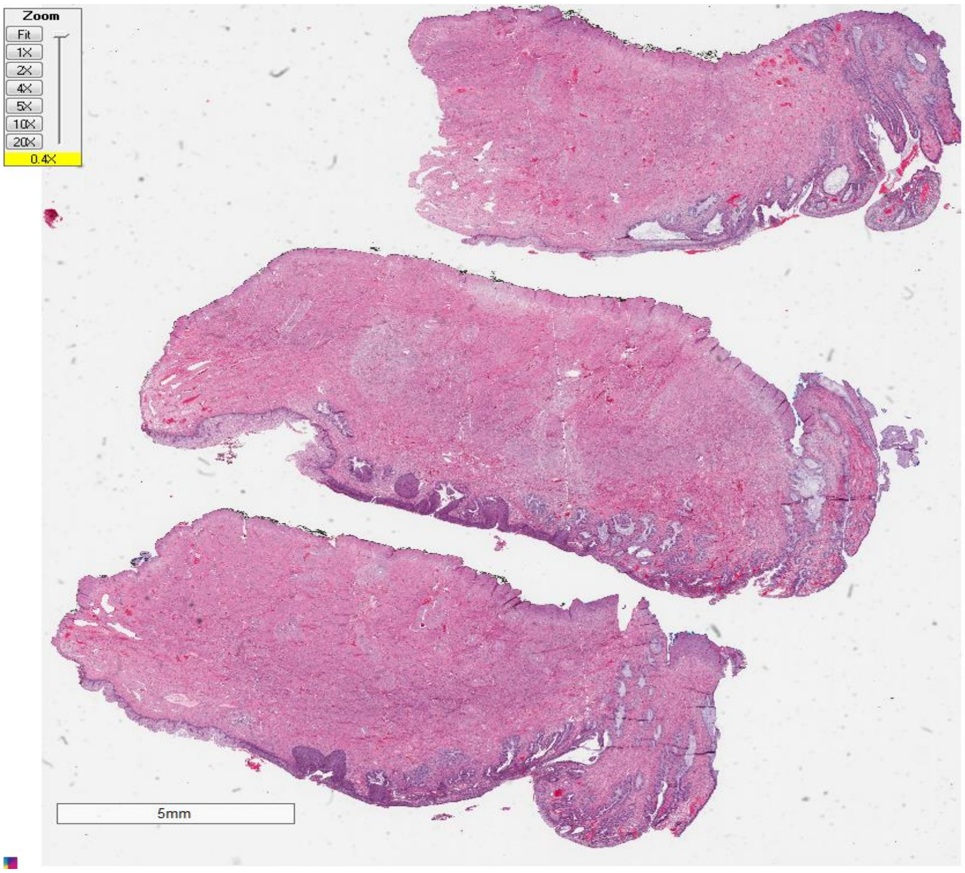

Figure 2.2 Histology slide at 0.4x zoom and resized for display purpose

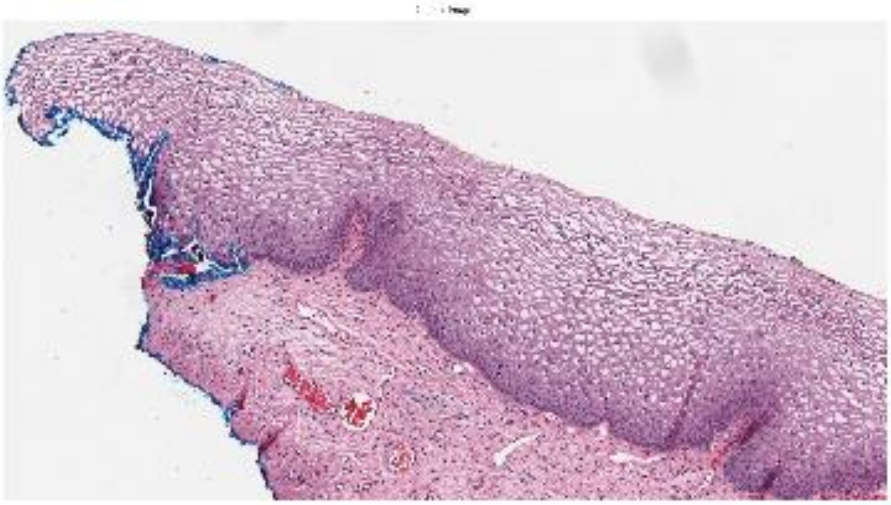

Figure 2.3 Block containing intra-epithelium region

In this work, a base image for every svs slide is considered and patches of 1000x1000 are extracted from these base images. Non-overlapping patches are considered for this work and the patches are extracted along the length and the breadth of the svs slide. For instance, 1600 patches are obtained from the image in Figure 2.2. after ignoring the patches with low information. These 1000x1000 RGB patches are resized to 250x250 RGB patches and are given as input the network. The input 250x250 patches, which contain the epithelium, are labeled as the epithelium patch and the ones containing non-epithelium or white background are labeled as the non-epithelium patch. Figure 2.4. shows the sample epithelium patch and Figure 2.5. shows the sample non-epithelium patch. These biomedical images are usually associated with large white background, such patches can be ignored by setting a threshold. The threshold can be obtained either by taking average of the RGB values of the epithelium, non-epithelium, and white background regions, or by converting the images in to grey scale, taking average of grey images and setting 0.8 as a threshold. Thereby we can eliminate the images above threshold as background images.

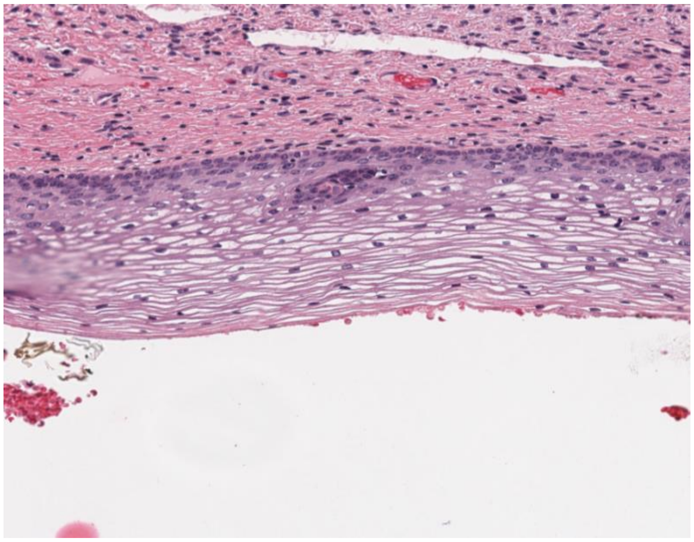

Figure 2.4 Sample epithelium patch

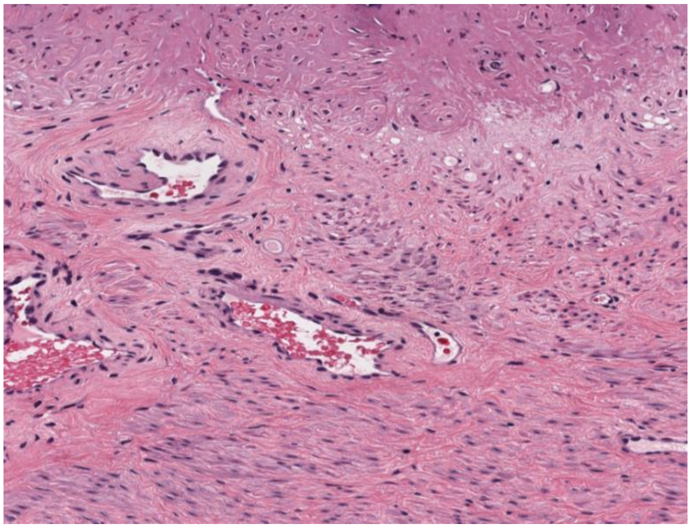

Figure 2.5 Sample non-epithelium patch

2.5. DATASET

The number of svs images that were considered for training is 50. For validation and testing, the number of svs images that were considered is 26. The svs slides are chosen such that the training svs slides are obtained in the year 2013, the validation and testing slides were obtained during 2015. Sets of patches are used for training, validation, and testing. Since the images have a lot of non-epithelium regions, the resulting dataset is skewed towards non-epithelium class. So, data augmentation is performed using data augmentor [11] for the epithelium patches. Augmentation techniques like the rotation of the image (90,180,270), image flip (horizontal and vertical), random cropping of the image with 90% of the region cropped are used. The number of patches generated for training data set is 41680, of which epithelium patches are 21839 and non-epithelium patches are 20841. The number of patches generated for validation data set is 8300 of which epithelium patches are 3000 and non-epithelium patches are 5300. The number of patches generated for testing data set is 8327, with 3131 patches containing at least some section of the epithelium region, and 5196 patches that do not contain any epithelium region. Table 2.1 provides the svs files used for generating this dataset. All of these svs files were obtained from the Pathology Department at the University of Oklahoma Health Sciences Center in collaboration with National Library of Medicine.

2.6. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks are successful because they try to mimic the functionality of the visual cortex of the brain. They are basically an extension of neural networks which use convolution operation in place of general matrix multiplication in their layers.

Table 2.1 Details of the patches from svs slides

Svs slide	**Number of patches from the svs slide for training**	**Number of patches from the svs slide for validation and testing**
1	307	
2	2002	
3	571	
4	170	
5	300	
6	992	
7	265	
8	495	
9	153	
10	314	
11	1132	
12	1225	
13	244	
14	539	
15	99	
16	253	
17	1144	
18	827	
19	537	
20	80	
21	191	
22	152	

Table 2.1 Details of the patches from svs slides (Cont.)

23	159	
24	289	
25	218	
26	756	
27	77	
28	447	
29	130	
30	189	
31	135	
32	94	
33	1179	
34	383	
35	1080	
36	446	
37	112	
38	108	
39	283	
40	116	
41	182	
42	277	
43	370	
44	502	
45	1195	
46	584	
47	924	
48	204	
49	976	
50	370	

Table 2.1 Details of the patches from svs slides (Cont.)

51		908
52		1180
53		81
54		672
55		816
56		674
57		1101
58		386
59		190
60		648
61		268
62		803
63		791
64		1560
65		805
66		21
67		16
68		7
69		39
70		57
71		23
72		40
73		24
74		25
75		62
76		59

If we use a two-dimensional image I(i,j) and two-dimensional kernel K(m,n),can be defined as below. Figure 2.6 shows the convolution operation performed on an image. Notice that based on the number of filters the depth of the input to the next layer increases. Filter size affects the size of the input to the next layer. Several Convolution layers are stacked for a deep learning CNN with filters of varying sizes, hen by maxpooling layers and followed by a fully connected neural network. Both shallow and deep learning CNN are considered for the classification of epithelium and non-epithelium patches.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \qquad (2.1)$$

Below are the four different architectures considered for this purpose.

a) 5 layer CNN

b) 8 layer CNN

c) VGG-19 (FC -1024)

d) VGG -19 (FC-500)

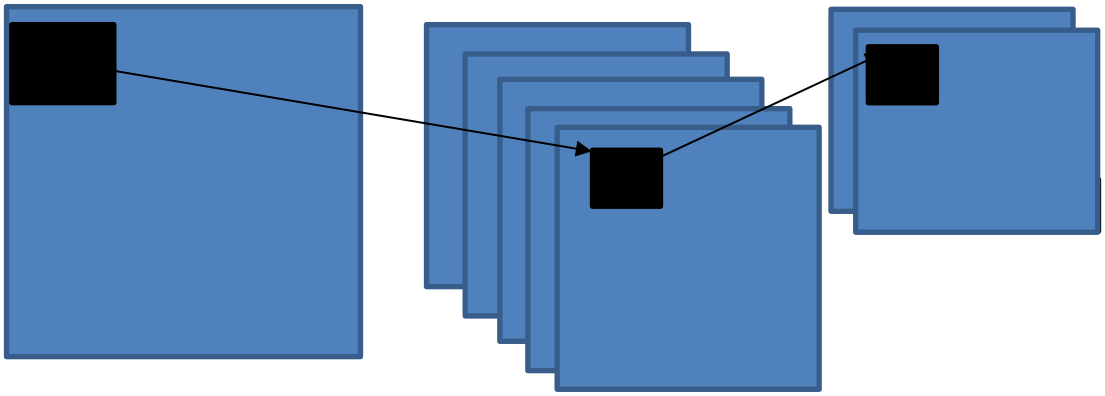

Figure 2.6 Convolution operation

2.7. IMPLEMENTATION OF THE DEEP LEARNING MODEL

This work is implemented in Python 3.6 using Keras [12]framework with Theano[13] as backend. The training dataset is taken as a numpy array and fed to the model. The hyperparameters used for CNN8 model are as follows.

a) Batch size= 32

b) Epochs=100.

c) Loss= binary cross entropy

d) Optimizer= RMS prop

e) Learning rate=1e-5

f) Zero padding

The layer details and the filter sizes are described in Table 2.2. For instance, Conv3-64 indicates that 64 filters of 3x3 sizes are used. Dropout (0.1) indicates the probability of neurons that are dropped each time for training the network. This helps to avoid overfitting and acts as a regularization technique for neural networks. Similarly, the rest of the table can be interpreted.

Since the problem under consideration is binary classification, binary cross entropy loss is used to represent the network and this loss is optimized to solve the problem. The loss equation is a below.

$$Loss = -(ylog\,(p) + (1-y)log(1-p)) \qquad (2.2)$$

Equation 2.2 Binary cross entropy loss

Where y is the binary indicator 0 or 1

p is the predicted probability

Table 2.2 Details of the deep learning model

<table>
<tr><td colspan="4">Architecture of the models</td></tr>
<tr><td>5 layer CNN
(5 weight layers)</td><td>8 layer CNN
(8 weight layers)</td><td>VGG-19 (FC -1024)
(19 weight layers)</td><td>VGG -19 (FC-500)
(19 weight layers)</td></tr>
<tr><td colspan="4">Input image(250x250 RGB patch)</td></tr>
<tr><td>Conv3-32</td><td>Conv3-32
Conv3-32</td><td>Conv3-64
Conv3-64</td><td>Conv3-64
Conv3-64</td></tr>
<tr><td colspan="4">Maxpool</td></tr>
<tr><td>Conv3-32</td><td rowspan="2">Dropout(0.1)</td><td>Conv3-128
Conv3-128</td><td>Conv3-128
Conv3-128</td></tr>
<tr><td>Maxpool</td><td colspan="2">Maxpool</td></tr>
<tr><td>Conv3-64</td><td>Conv2-64</td><td>Conv3-256
Conv3-256
Conv3-256
Conv3-256</td><td>Conv3-256
Conv3-256
Conv3-256
Conv3-256</td></tr>
<tr><td colspan="4">Maxpool</td></tr>
<tr><td rowspan="2">FC-64</td><td rowspan="2">Dropout(0.2)</td><td>Conv3-512
Conv3-512
Conv3-512
Conv3-512</td><td>Conv3-512
Conv3-512
Conv3-512
Conv3-512</td></tr>
<tr><td colspan="2">Maxpool</td></tr>
<tr><td>Dropout(0.5)</td><td>Conv2-128</td><td>Conv3-512
Conv3-512
Conv3-512
Conv3-512</td><td>Conv3-512
Conv3-512
Conv3-512
Conv3-512</td></tr>
<tr><td>FC-1</td><td>Maxpool</td><td colspan="2">Maxpool</td></tr>
<tr><td>Sigmoid</td><td>Dropout(0.3)</td><td>FC-1024</td><td>FC-500</td></tr>
<tr><td></td><td>FC-100</td><td colspan="2">Dropout</td></tr>
<tr><td></td><td>Dropout(0.5)</td><td>FC-1024</td><td>FC-500</td></tr>
<tr><td></td><td>FC-100</td><td>FC-1</td><td>FC-1</td></tr>
<tr><td></td><td>FC-1</td><td colspan="2" rowspan="2">Sigmoid</td></tr>
<tr><td></td><td>Sigmoid</td></tr>
</table>

2.8. TRANSFER LEARNING

The number of parameters for the deep CNN is huge in the order of millions. For instance, the VGG network has ~30 million parameters to train the network. So this requires very high computational cost, in order to overcome this transfer learning is used, whereby a pre-trained network on a different/ similar dataset is used and the weights are trained for our dataset by freezing some layers of the network. This works because the initial layers in the CNN detect the low-level features in the image like edges, curves etc. These low-level features are common to most of the problems that we consider. In this work, the VGG-19 fully connected (FC 1024) network, which is trained on the Imagenet dataset is used for transfer learning. The first 5 layers are frozen and the rest of the layers are trained by the epithelium dataset.

2.9. RESULTS

The validation results for the various models are as below; the highest accuracy obtained from the validation data set is 98.84%. The number of epochs used for training all the architectures is 100. The weights for this model are taken and the highest accuracy obtained for the test data is 98.82%. The Tables 2.3, 2.4 describes the performance metrics for classification like accuracy, precision, recall and F-score.

2.10. EPITHELIUM DETECTION IN A SINGLE SVS IMAGE

To validate the intra-epithelium detection, we considered a single svs image as in Figure 2.2. and detected the intra-epithelium regions as highlighted. The below Figures 2.7-2.12 are zoomed in, vertically flipped images of Figure 2.2. This shows the region of interest is detected and their corresponding confidence map, which is helpful for the pathologists to identify the epithelium region.

Table 2.3 Accuracy of the deep learning model

Models	Validation Accuracy	Test Accuracy
CNN-5	91.63%	89.25%
CNN-8	96.84%	92.02%
VGG-19(FC 1024)	98.68%	**98.82%**
VGG-19(FC 500)	**98.84%**	98.76%

Table 2.4 Performance metrics for classification

Models	TP	FP	FN	Precision	Recall	F-score
CNN-5	7433	377	517	0.9517	0.9350	0.9433
CNN-8	7664	288	375	0.9638	0.9534	0.9585
VGG-19(FC 1024)	8230	23	74	0.9972	0.9911	0.9941
VGG-19(FC 500)	8224	43	60	0.9948	0.9928	0.9938

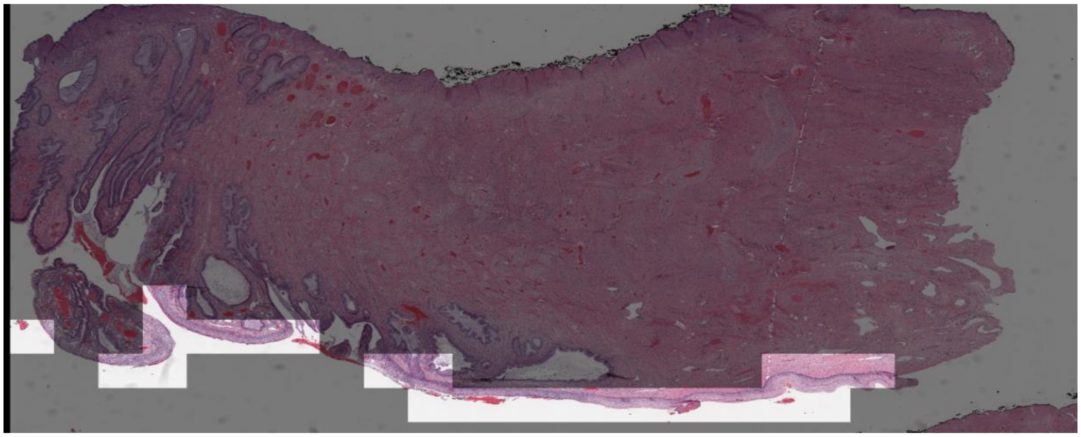

Figure 2.7 Intra epithelium detection on a single svs image part1

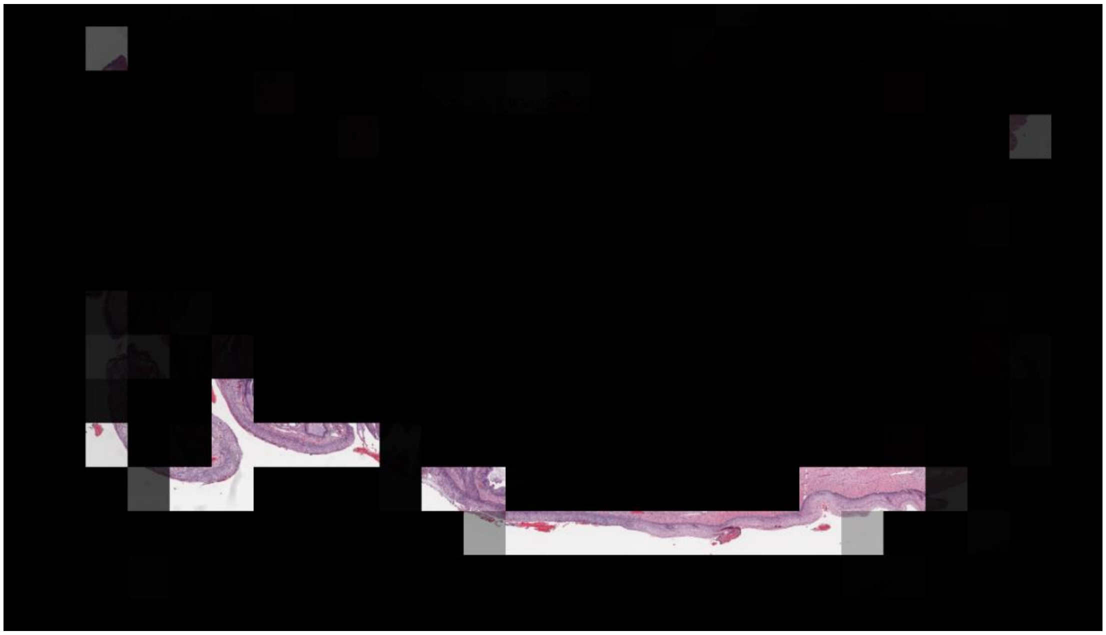

Figure 2.8 Confidence map of intra-epithelium detection on a single svs image part1

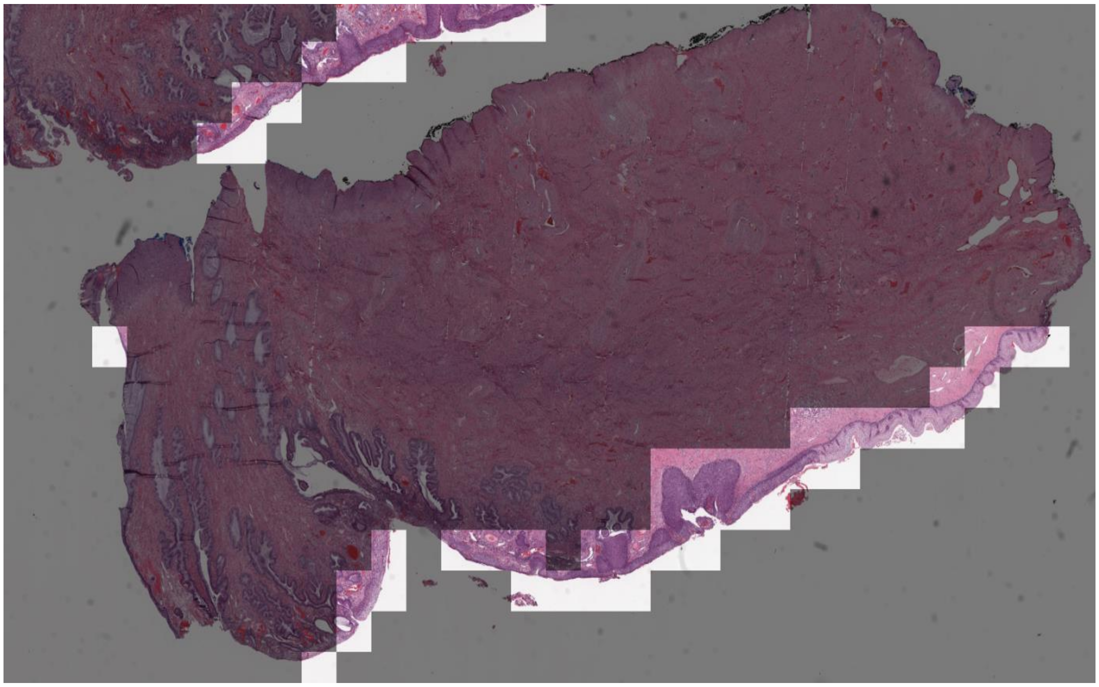

Figure 2.9 Intra epithelium detection on a single svs image part2

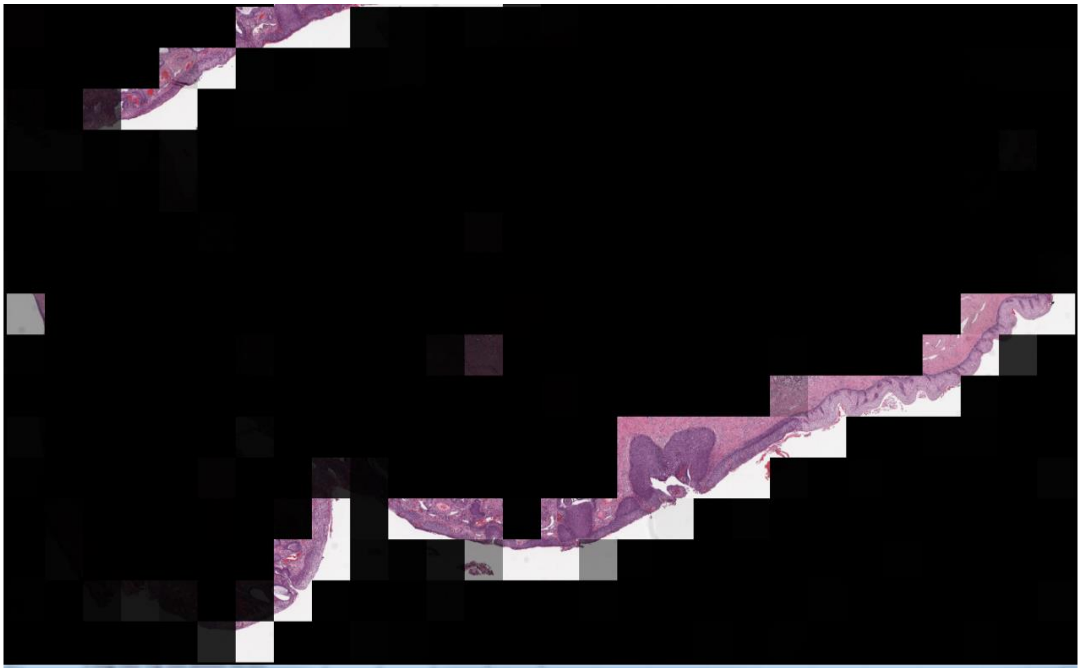

Figure 2.10 Confidence map of intra-epithelium detection on a single svs image part2

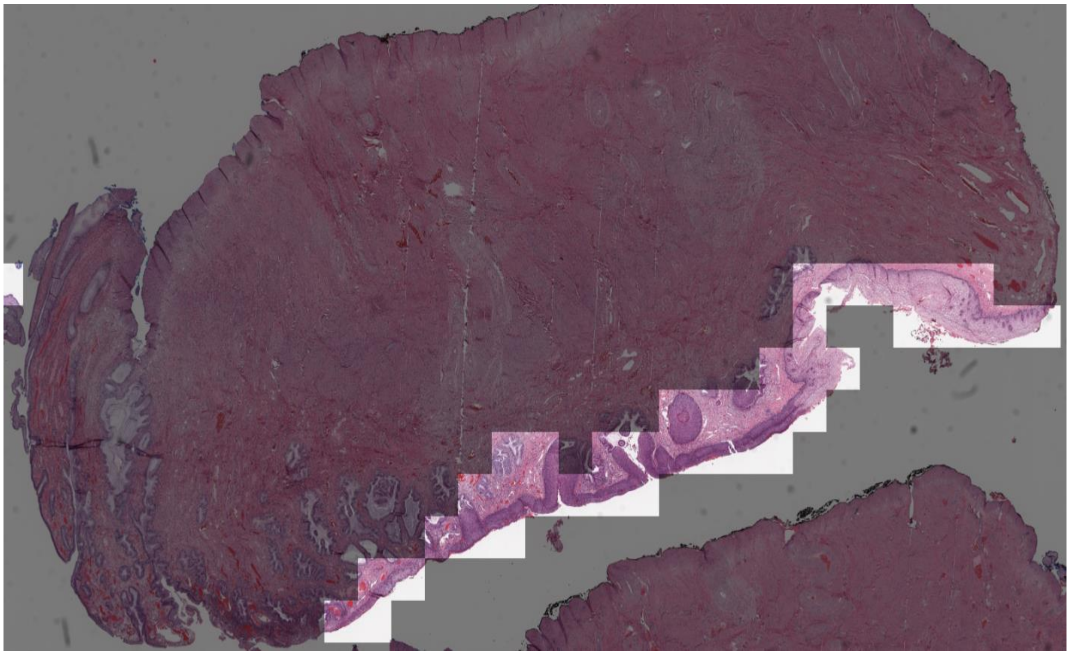

Figure 2.11 Intra-epithelium detection on a single svs image part3

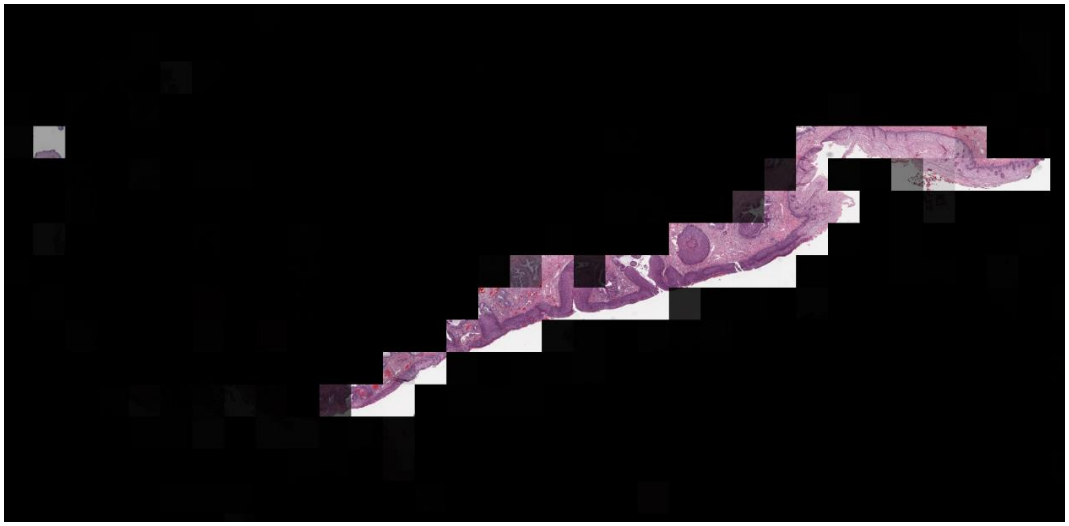

Figure 2.12 Confidence map of intra-epithelium detection on a single svs image part3

2.11. FALSE DETECTION

Below patches, as shown in Figures 2.13-2.16, were falsely detected as the non-epithelium region. The reason for misclassification could be due to staining of the histology slide, narrow cross section of epithelium region and small size of nuclei.

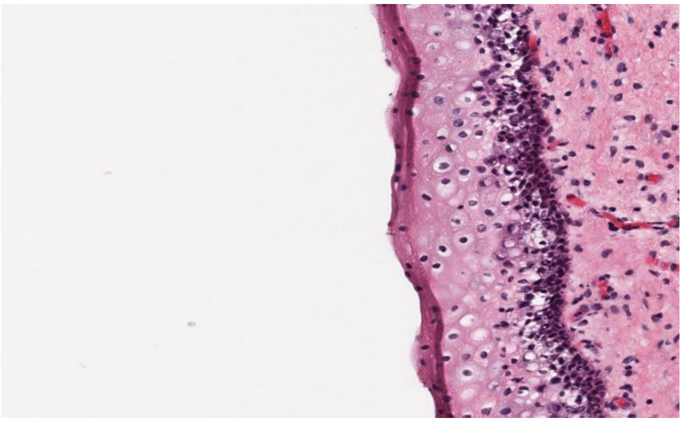

Figure 2.13 Misclassified as the non-epithelium patch example1

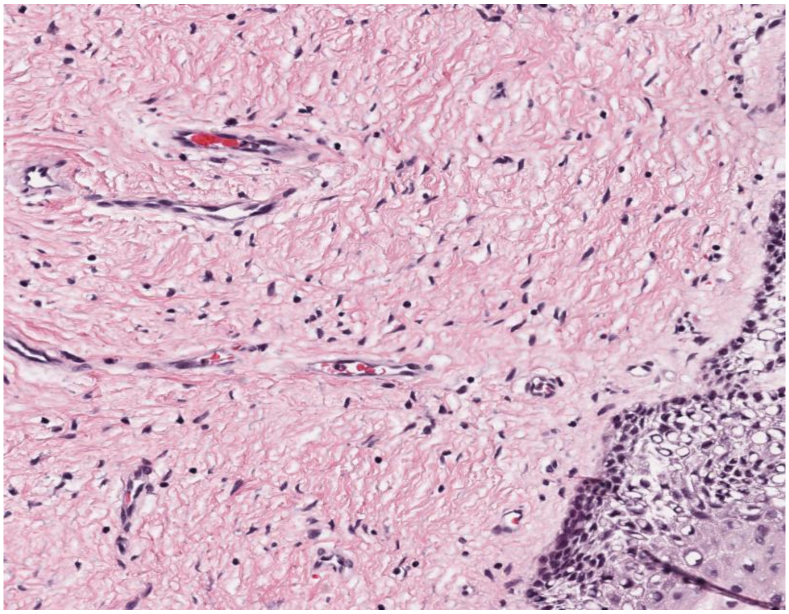

Figure 2.14 Misclassified as the non-epithelium patch example2

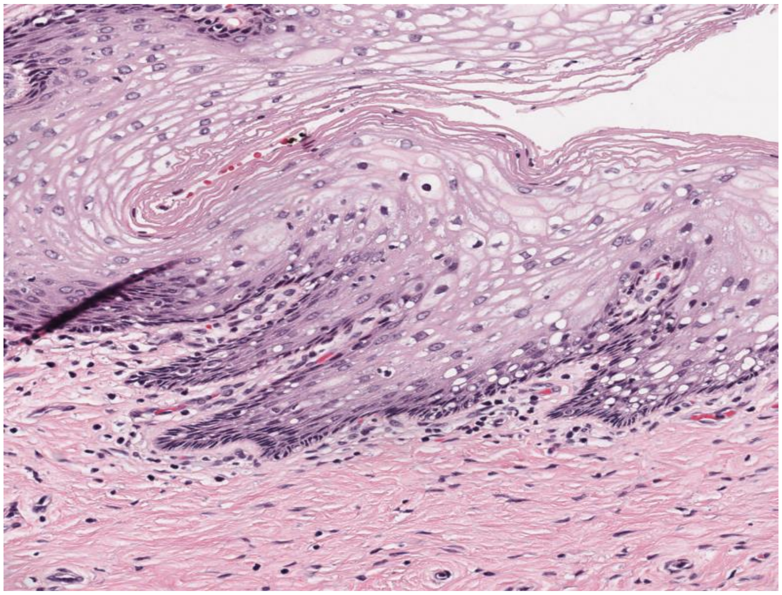

Figure 2.15 Misclassified as the non-epithelium patch example3

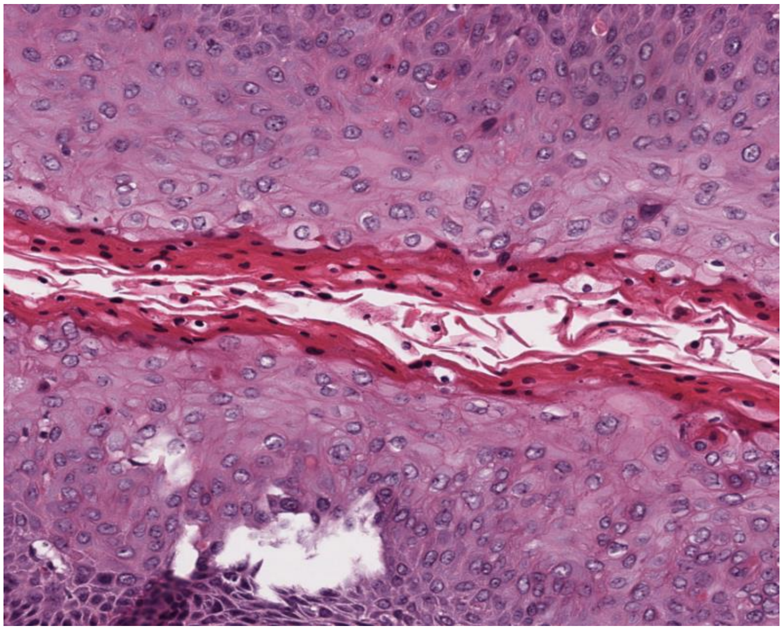

Figure 2.16 Misclassified as the non-epithelium patch example4

Below patches, as shown in Figures 2.17-2.20, were falsely detected as the epithelium region. The primary reason for these misclassifications could be due to minimal presence of nuclei in the patches.

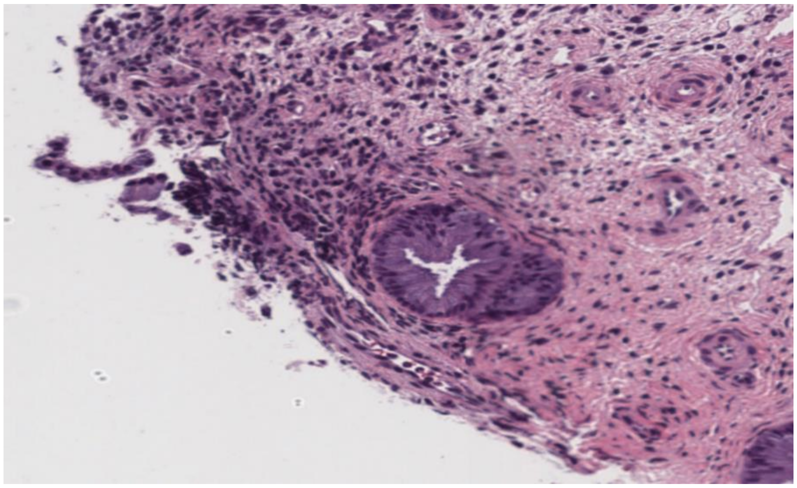

Figure 2.17 Misclassified as the epithelium patch example1

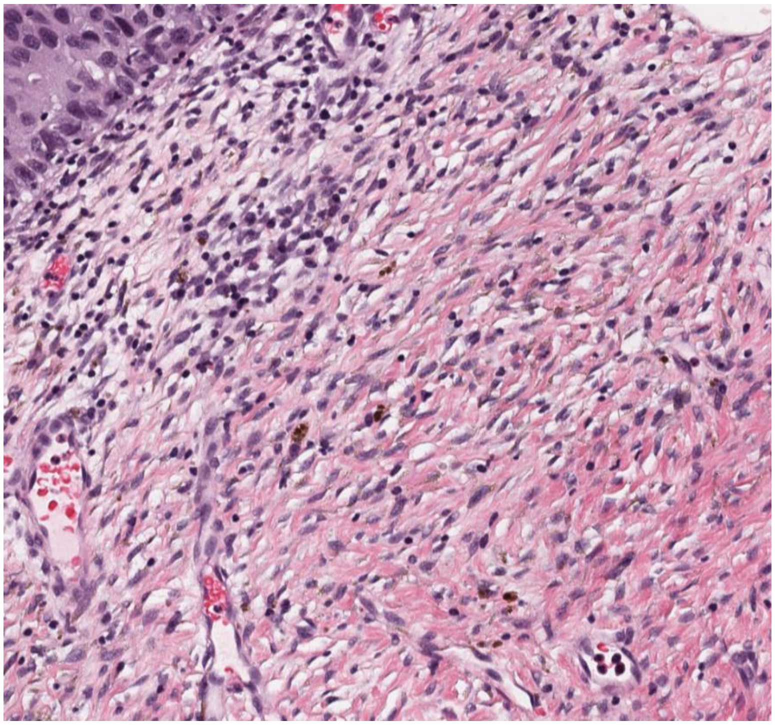

Figure 2.18 Misclassified as the epithelium patch example2

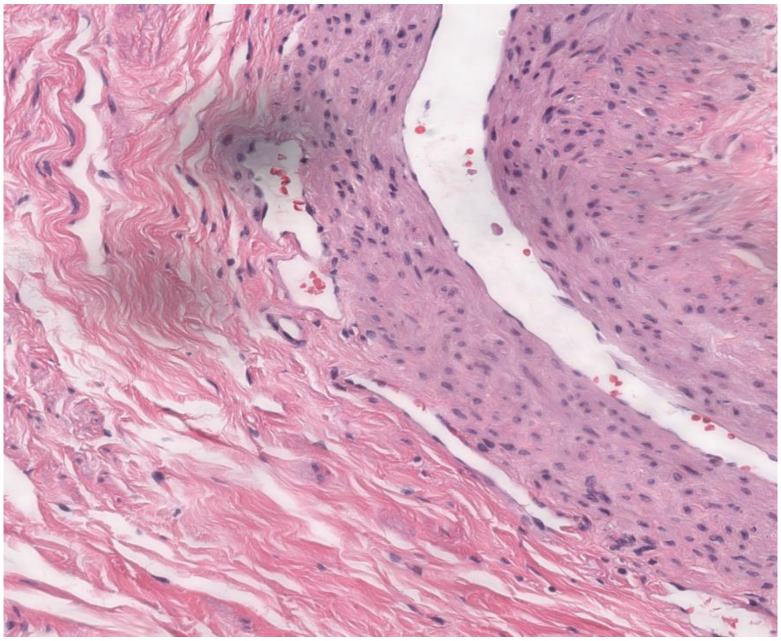

Figure 2.19 Misclassified as the epithelium patch example3

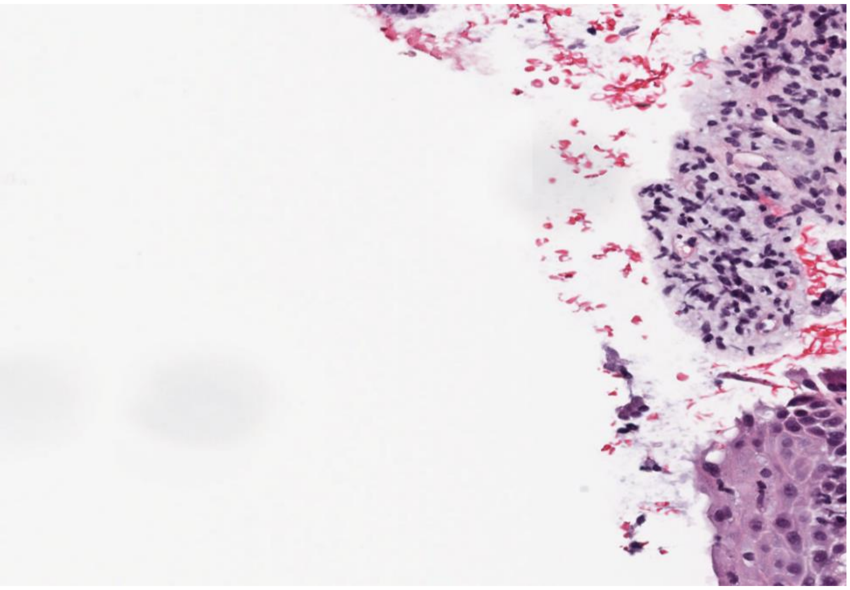

Figure 2.20 Misclassified as the epithelium patch example4

3. CERVICAL INTRAEPITHELIAL NEOPLASIA (CIN) CANCER CLASSIFICATION

3.1. OVERVIEW

Cervical cancer detection in early stages is crucial for its treatment. With the advent of digitization of the medical field, biomedical image analysis is playing a crucial role in assisting the pathologists in cervical cancer assessment. In this research, cervical intra-epithelial neoplasia (CIN), a pre-cancerous condition, is used for digitized histology image analysis of the epithelium region. This work focusses on automating the epithelium region analysis process to serve as a diagnostic aide for pathologists for improving CIN assessment capability. Figure 3.2 shows the algorithm for digitized histology image analysis used in this CIN classification study. In this work, CIN classification is performed on a 117 digitized histology image dataset, which was manually extracted from .svs files from Table 2.1. For epithelium classification, the CIN categories include Normal, CIN1, CIN2, CIN3 as illustrated in Figure 3.1. Earlier research [15][16] focused on splitting the epithelium region into 10 vertical partitions(segments) along the medial axis for individual vertical partition CIN classification which is fused for whole epithelium image for the CIN classification.

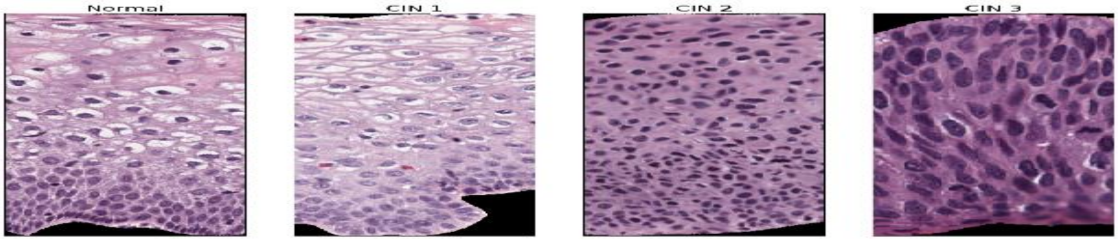

Figure 3.1 Shows an example of different CIN grades from the segmented images

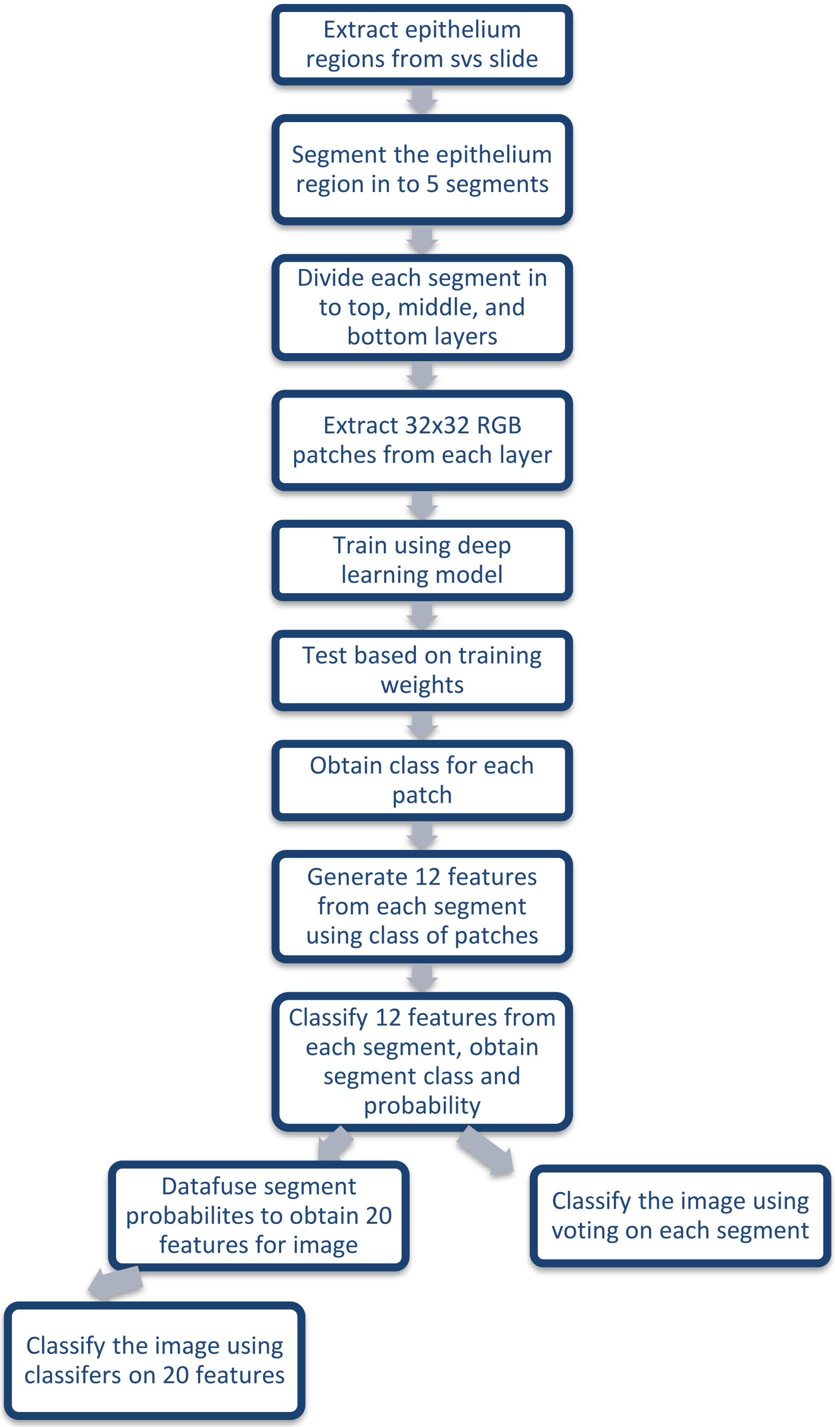
Extract epithelium regions from svs slide
Segment the epithelium region in to 5 segments
Divide each segment in to top, middle, and bottom layers
Extract 32x32 RGB patches from each layer
Train using deep learning model
Test based on training weights
Obtain class for each patch
Generate 12 features from each segment using class of patches
Classify 12 features from each segment, obtain segment class and probability
Datafuse segment probabilites to obtain 20 features for image
Classify the image using voting on each segment
Classify the image using classifers on 20 features

Figure 3.2 Algorithm steps for digitized histology image analysis

3.2. PREPROCESSING

Deep learning requires more data (typically in the order of thousands) for training the network. From the 117 epithelium images in the dataset, preprocessing operations are performed to generate the input data for the deep learning network. In generating the 117 epithelium image dataset, the preprocessing operations consist of extracting the data, epithelium region from the raw svs histology slides as shown in Figure 3.3.

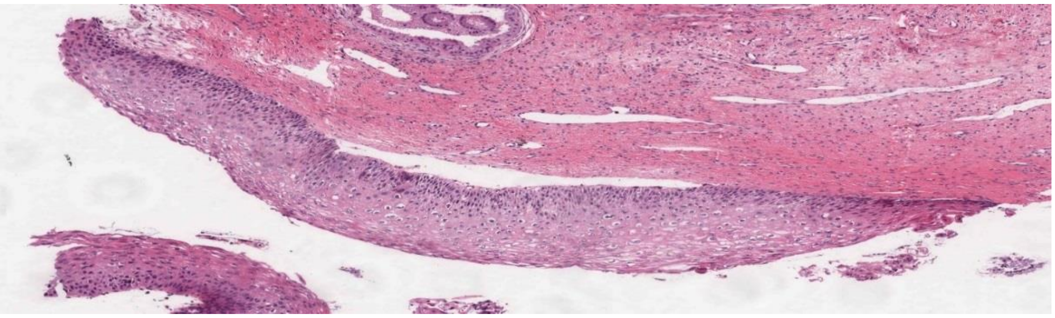

Figure 3.3 Epithelium and non-epithelium region from histology slides

The epithelium region is masked out and only epithelium region is extracted since it is the region of interest. Since these are biomedical images, the orientation of the epithelium region is not uniform in the images as shown in Figure 3.4.

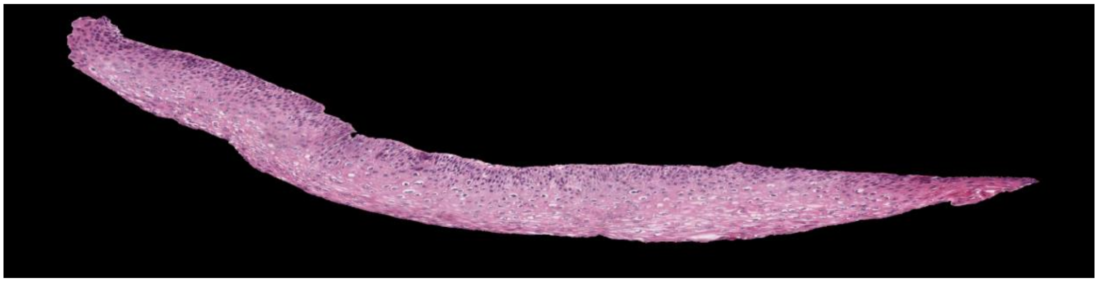

Figure 3.4 Epithelium region extracted from histology slides

In order to bring a standard form to the data, the images are oriented horizontally as shown in Figure 3.5.

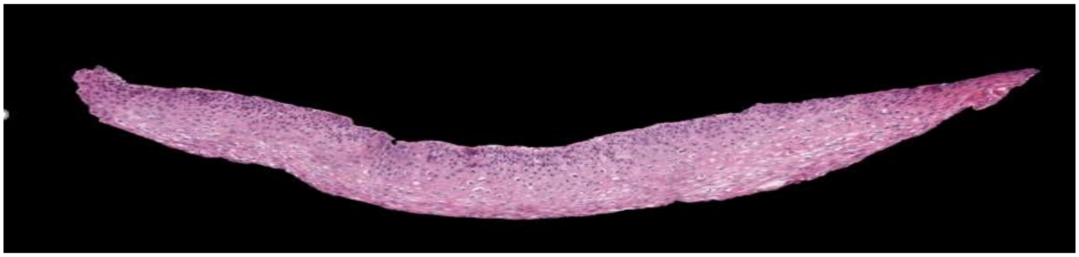

Figure 3.5 Epithelium region with orientation

The medial axis for these images in the epithelium region is determined based on the algorithm from [15] and is shown by the blue line in Figure 3.6. Orthogonal segments from this medial axis are generated and each image is split into 5 segments as shown in Figure 3.6. based on the method from[15].

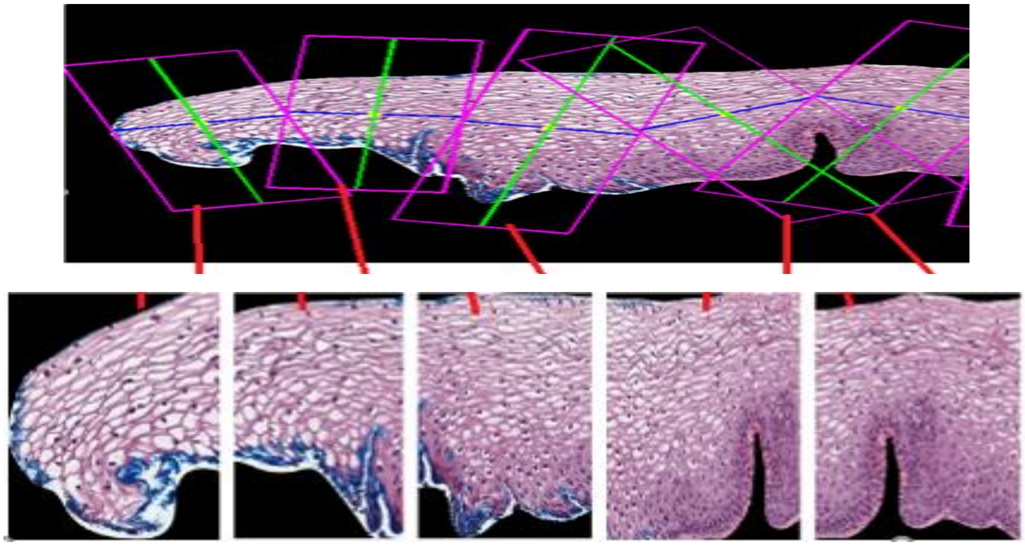

Figure 3.6 Epithelium region divided into 5 segments for CIN classification

Each segment is further split in to top, middle and bottom regions of equal size as shown in Figure 3.7. The whole segment is also shown in the below figure.

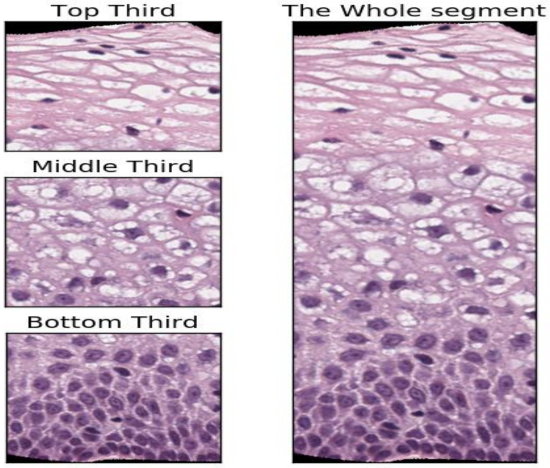

Figure 3.7 Epithelium segment divided into top, middle and bottom parts

Patches/chunks of size 32x32 are extracted from these segments as shown in Figure 3.8. These are used for training the deep learning model.

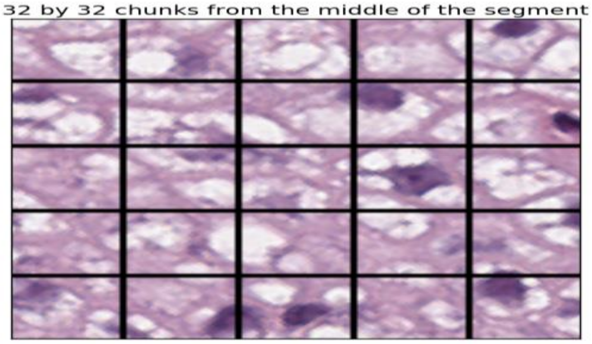

Figure 3.8 Chunks generated from the segment

3.3. DATASET

The dataset used for CIN classification is from 117 Images. The number of segments that are generated from these images is 585 with 5 segments obtained from each image. The patches from the top, middle, and bottom segments are generated and these patches are used for training the network. Based on the images they correspond to, the patches are split into 5 folds. So, 5 fold cross validation is performed. At any given point in time, four folds are used for training and the remaining fold is used for testing purpose. The data is split evenly between the folds based on the images. The number of 32x32 RGB patches generated in the top segment is 40321, the number of 32x32 RGB patches generated in the middle segment is 59627, the number of 32x32 RGB patches generated in the bottom segment is 40420.

3.4. DEEP LEARNING MODEL

The deep learning model used for the prediction is specified in the Table3.1. It has two convolution layers with 32 filters of size 3x3 for extracting the features of the images using convolution operation. It is followed by maxpooling layer to identify the maximum feature contributed in the filter size of 2x2 from the convoluted image. It is followed by then the dropout layer with a probability of 0.1 for better regularization and to prevent the overfitting of the network, then one convolution layer with 64 filters of size 2x2 followed by a maxpooling layer of size(2,2). It is followed by one more convolution layer with 128 filters of size 2x2 followed by a maxpooling layer of size (2,2), then dropout layer with a probability of 0.2. It is then followed by two fully connected dense layer with 500 neurons with dropout layer with a probability of 0.3, followed by the 4 neurons for the prediction of the classes.

3.5. DATA FUSION USING FEATURES FROM CNN MODEL

The classification results from the CNN model obtained are at the patch level, each patch is classified into either normal, CIN1, CIN2, or CIN3. In order to obtain the classification at the image level, the following data fusion technique is used. Calculating the patch level accuracy for the entire vertical segment gives 12 feature vectors for each vertical segment. As illustrated in Figure 3.7 each vertical segment is divided into 3 layers (top, middle, and bottom). The overall classification of these layers into each of the four classes is calculated from the below equation. Therefore, each layer gives four probabilities corresponding to each class, which in turn gives 12 features for the entire vertical segment. These 12 features from all the vertical segments of the images are used for further classification of vertical segments to obtain the vertical segment level accuracy using leave one image out approach. Here the features from all the images segments except one image are used for training, the left out image segments are used for testing. The classifiers like SVM [20], MLP [21], LDA [22], Logistic, RF [23] classifiers are used for this purpose. Once this segment level classification is done, we have the prediction class for each segment and their prediction probabilities for each segment. These prediction classes at the segment level are used as a basis for a voting scheme, where the max number of segments classified into one class is taken as the overall image classification. The four prediction probabilities from each vertical segment corresponding to 20 features for a single image, as there are 5 vertical segments. These 20 features are further used for training the above-mentioned classifiers to obtain the image level accuracy.

$$Prob(x = class|layer\ y) = \frac{Number\ of\ patches\ classified\ as\ class\ x}{Number\ of\ patches\ in\ \ layer\ y} \quad (3.1)$$

Equation 3.1 Probabilities of the layer corresponding to the four classes

3.6. IMPLEMENTATION OF THE MODEL

This work is implemented in Python 3.6 using Keras [12] framework with Theano[13] as the backend software. The patches from the top, middle and bottom layers are fed to three networks with the architecture mentioned in Table 3.1 The training dataset for each of the layers is taken as a numpy file and given as input to the model. The hyperparameters used for CNN model are as follows. This architecture uses 5- fold cross-validation.

a) Batch size= 500

b) Epochs=300

c) Loss= categorical cross entropy

d) Optimizer= Adam

e) Learning rate=1e-5

f) Zero padding

3.7. CATEGORICAL CROSS ENTROPY LOSS

Since the problem under consideration is multi-classification, categorical cross entropy loss is used to represent the network and this loss is optimized to solve the multiclass classification problem. The loss equation is given in Equation 3.2 . Table 3.1 contains the CNN architecture and parameters used in the training process.

$$Loss = -\sum_{c=1}^{M} y_{o,c} log p_{o,c} \qquad (3.2)$$

Equation 3.2 Categorical cross entropy loss

M is the number of classes

Where y is the binary indicator 0 or 1

p is the predicted probability observation o is of class c

Table 3.1 CNN architecture

Layer type	**Layer properties**
Input	Size (3,32,32)
Convolution	32 filters, size (3,3)
Convolution	32 filters, size (3,3)
Max pooling	Size (2,2)
Dropout	0.1
Convolution	64 filters, size (2,2)
Max pooling	Size (2,2)
Convolution	128 filters, size (2,2)
Max pooling	Size (2,2)
Dropout	0.2
Dense	500 Neurons
Dropout	0.3
Dense	500 Neurons
Dense (output)	4 Neurons

3.8. RESULTS

The classification is performed after data fusion of probabilities obtained from CNN model. The classification results using segment level and at the image level using both voting scheme and using 20 feature vectors is tabulated below. For testing, classification is performed using leave one out image strategy using support vector machines(SVM), linear discriminant analysis(LDA), multi-layer perceptron(MLP), logistic and random forest(RF) classifiers. The best image accuracy obtained is 77.27% for MLP classifier using voting.

3.8.1. Classification Accuracy Using Leave-one-out Approach. The classification accuracy using different datasets is shown in Tables 3.2 and 3.3. The 90-image dataset is a subset to of 117-image dataset used in [15], the dataset was created by excluding the images in which vertical segment cannot be divided into three layers as they contain thin epithelium. The 66-image dataset used for comparison are the same images used in [16] which is also a subset of the images from the 117-image dataset. The highest accuracy obtained for the 10 segments approach in [16] is 77.27%, which is the same as the highest accuracy obtained using 5 segments.

3.8.2. Patch Level CIN Classification. The patch level CIN classification for the top, middle, and bottom layers is tabulated in Tables 3.5 and 3.6. The fully trained model accuracy is based on the weights obtained at the end of the training, the best model accuracy is based on the weights obtained at the lowest loss while training. These results specify the accuracy of the patch, whether that is correctly classified as Normal, CIN1, CIN2, and CIN3. The confusion matrix for the 66-image dataset using MLP classifier is provided in Table 3.4.

Table 3.2 Classification accuracy using different methods

	SVM	LDA	MLP	Logistic	RF
Using all the images in the dataset(117 Images)					
Segment Accuracy	62.45%	64.62%	62.82%	61.37%	61.19%
Image accuracy (Voting)	**63.25%**	**66.67%**	**66.67%**	**61.54%**	**61.54%**
Image Accuracy (20 Features)	59.83%	63.25%	61.54%	59.83%	**58.12%**
Leaving images with less information (90 Images)					
Segment Accuracy	64.22%	64.00%	64.89%	**62.89**%	60.22%
Image accuracy (Voting)	68.89%	**68.89**%	**67.78**%	62.22%	63.33%
Image Accuracy (20 Features)	**72.22**%	58.89%	**58.89**%	60.00%	**65.56**%
Using 66 image dataset					
Segment Accuracy	69.09%	70.61%	72.73%	70.30%	70.61%
Image accuracy (Voting)	**71.21**%	**74.24**%	**77.27**%	71.21%	74.24%
Image Accuracy (20 Features)	68.18%	68.18%	**63.64**%	**74.24**%	**75.76**%
Using only top and middle layers of 66 image dataset					
Segment Accuracy	70.91%	70.61%	69.39%	69.70%	68.79%
Image accuracy (Voting)	**74.24**%	**75.76**%	**74.24**%	**71.21**%	**74.24**%
Image Accuracy (20 Features)	66.67%	68.18%	68.18%	68.18%	69.70%

Table 3.3 CIN classification accuracy using MLP classifier

Using MLP Classifier	**Normal**	**CIN1**	**CIN2**	**CIN3**
Using all the images in the dataset(117 Images)				
Segment Accuracy	84.85%	21.50%	49.02%	72.79%
Image accuracy (Voting)	95.12%	21.74%	54.55%	70.97%
Image Accuracy (20 Features)	90.24%	8.70%	36.36%	80.65%
Leaving images with less information (90 Images)				
Segment Accuracy	93.33%	7.06%	46.67%	75.45%
Image accuracy (Voting)	97.22%	5.88%	53.33%	77.27%
Image Accuracy (20 Features)	88.89%	11.76%	20.00%	72.73%
Segment Accuracy	93.13%	17.78%	56.67%	75.38%
Using 66 image dataset				
Image accuracy (Voting)	96.88%	33.33%	58.33%	76.92%
Image Accuracy (20 Features)	87.50%	11.11%	41.67%	61.54%
Segment Accuracy	96.88%	0.00%	51.67%	66.15%
Using only top and middle layers of 66 image dataset				
Image accuracy (Voting)	100.00%	0.00%	75.00%	61.54%
Image Accuracy (20 Features)	90.63%	22.22%	50.00%	61.54%

Table 3.4 Confusion Matrix for 66 Image dataset using 66 MLP classifier

	Normal	CIN1	CIN2	CIN3
Normal	31	1	0	0
CIN1	5	3	1	0
CIN2	1	1	7	3
CIN3	0	0	3	10

Table 3.5 Patch accuracy at the top, middle, and bottom layers

Fold	Top layer (fully trained model)			Top layer (best model)		
	Training	Validation	Testing	Training	Validation	Testing
1	75.85%	50.45%	67.60%	75.75%	50.32%	68.02%
2	80.96%	47.05%	54.36%	79.34%	47.85%	57.06%
3	77.50%	49.65%	75.13%	76.34%	48.38%	76.03%
4	77.04%	50.41%	78.05%	76.51%	50.53%	78.03%
5	76.76%	78.15%	47.29%	76.76%	78.65%	48.24%
Average	**77.62**%	**55.14**%	**64.49**%	**76.94**%	**55.15**%	**65.48**%
Standard Deviation	0.02	0.12	0.12	0.01	0.12	0.11

Fold	Middle layer(fully trained model)			Middle layer (best model)		
	Training	Validation	Testing	Training	Validation	Testing
1	74.52%	46.56%	38.47%	68.41%	41.71%	55.20%
2	78.73%	50.05%	49.93%	76.34%	48.83%	53.93%
3	73.88%	46.87%	58.91%	72.78%	48.28%	62.11%
4	73.25%	44.76%	67.36%	58.45%	44.40%	72.81%
5	74.89%	67.34%	45.11%	71.93%	64.89%	45.00%
Average	**75.05**%	**51.12**%	**51.96**%	**69.58**%	**49.62**%	**57.81**%
Standard Deviation	0.02	0.08	0.10	0.06	0.08	0.09

Fold	Bottom layer (fully trained model)			Bottom (best model)		
	Training	Validation	Testing	Training	Validation	Testing
1	61.73%	36.20%	57.67%	59.30%	35.65%	56.46%
2	64.39%	34.27%	36.42%	64.37%	35.00%	37.78%
3	58.69%	38.82%	47.08%	57.52%	40.28%	53.93%
4	62.50%	35.72%	54.75%	56.90%	34.87%	64.60%
5	64.80%	47.16%	38.58%	56.23%	59.90%	47.41%
Average	**62.42**%	**38.43**%	**46.90**%	**58.86**%	**41.14**%	**52.04**%
Standard Deviation	0.02	0.05	0.08	0.03	0.10	0.09

Table 3.6 Patch accuracy for CIN grades at the top, middle, and bottom layers

Fold	Top layer			
	Normal	CIN1	CIN2	CIN3
1	98.77%	2.59%	22.43%	65.53%
2	79.29%	0.50%	43.37%	65.53%
3	84.05%	5.54%	37.16%	63.14%
4	89.16%	5.27%	47.54%	86.51%
5	73.61%	7.86%	26.93%	63.38%
Average	**84.98%**	**4.35%**	**35.49%**	**68.82%**
Standard Deviation	0.0962	0.0285	0.1065	0.0996

Fold	Middle layer			
	Normal	CIN1	CIN2	CIN3
1	62.86%	61.15%	19.31%	63.83%
2	82.36%	4.31%	31.40%	59.30%
3	72.28%	18.59%	10.82%	53.89%
4	75.67%	5.29%	46.84%	77.88%
5	90.07%	5.92%	1.11%	57.24%
Average	**76.65%**	**19.05%**	**21.90%**	**62.43%**
Standard Deviation	0.1028	0.2425	0.1785	0.0936

Fold	Bottom layer			
	Normal	CIN1	CIN2	CIN3
1	87.04%	4.82%	0.00%	56.99%
2	90.75%	0.00%	18.01%	0.40%
3	70.80%	0.00%	0.57%	41.65%
4	79.49%	0.00%	15.39%	17.84%
5	74.68%	0.16%	0.03%	48.93%
Average	**80.55%**	**1.00%**	**6.80%**	**33.16%**
Standard Deviation	0.0832	0.0214	0.0909	0.2343

4. CONCLUSION

This work successfully implements intra-epithelium detection in cervical histology images with a very good accuracy of 98.76%. Since the size of the digital histology images are very large and comes with 20x zoom level, it takes longer time for pathologists to identify the epithelium region. This work facilitates and identifies the epithelium region so that the pathologists can focus on the specific areas instead of the finding where the epithelium region is located. In this study, various CNN architectures were implemented and the best results are obtained for the VGG-19 architecture using transfer learning. Deep CNN performed better when compared to shallow networks.

The second part of the study focused on the cervical cancer classification. In this work since the dataset is small, chunks of small size were extracted and chunk level accuracy is determined. Data fusion techniques are used for determining the overall accuracy of the images. The features extracted using CNN are utilized as input to the classifiers like SVM, MLP, Logistic and RF. Maximum accuracy of 77.27% is obtained using this dataset. The accuracy can be further improved by using larger datasets and using different preprocessing techniques to generate patches of different sizes. Future work involves exploring these possibilities to further improve the accuracy of the CIN classification.

BIBLIOGRAPHY

[1] Siegel, Rebecca L., Kimberly D. Miller, Stacey A. Fedewa, Dennis J. Ahnen, Reinier GS Meester, Afsaneh Barzi, and Ahmedin Jemal. "Colorectal cancer statistics, 2017." CA: a cancer journal for clinicians 67, no. 3 (2017): 177-193.

[2] J. Jeronimo, M. Schiffman, R. L. Long, L. Neve, and S. Antani, “A tool for collection of region based data from uterine cervix images for correlation of visual and clinical variables related to cervical neoplasia,” in Proceedings. 17th IEEE Symposium on Computer- Based Medical Systems, 2004, pp. 558–562.

[3] He, L., Long, LR., Antani, S., Thoma, G. (2010) Computer Assisted Diagnosis in Histopathology. In: Zhao, Z., (Ed). Sequence and Genome Analysis: Methods and Applications. iConcept Press; pp. 271–287.

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

[5] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In European conference on computer vision, pp. 818-833. Springer, Cham, 2014.

[6] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[7] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248-255. IEEE, 2009.

[8] Janowczyk, Andrew, and Anant Madabhushi. "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases." Journal of pathology informatics 7 (2016).

[9] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12, no. Oct (2011): 2825-2830.

[10] Goode, Adam, Benjamin Gilbert, Jan Harkes, Drazen Jukic, and Mahadev Satyanarayanan. "OpenSlide: A vendor-neutral software foundation for digital pathology." Journal of pathology informatics 4 (2013).

[11] Bloice, Marcus D., Christof Stocker, and Andreas Holzinger. "Augmentor: An Image Augmentation Library for Machine Learning." arXiv preprint arXiv:1708.04680 (2017).

[12] Chollet, François. Keras: Deep learning library for theano and tensorflow. (2015). https://keras.io/

[13] Al-Rfou, Rami, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien et al. "Theano: A Python framework for fast computation of mathematical expressions." arXiv preprint(2016).

[14] Good fellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1. Cambridge: MIT press, 2016.

[15] Guo, Peng, Haidar Almubarak, Koyel Banerjee, R. Joe Stanley, Rodney Long, Sameer Antani, George Thoma et al. "Enhancements in localized classification for uterine cervical cancer digital histology image assessment." Journal of pathology informatics vol. 7, no. 1, p. 51, 2016.

[16] Almubarak H, Stanley RJ, Long R, Antani S, Thoma G, Zuna R, Frazier S. Convolutional Neural Network Based Localized Classification of Uterine Cervical Cancer Digital Histology Images. Accepted for publication in the Proceedings of the Conference for Complex Adaptive Systems: Engineering Cyber Physical Systems, Chicago, IL, October 30-November 1, 2017.

[17] Guo P, Almubarak H, Banerjee K, Stanley RJ, Long R, Antani S, Thoma G, Zuna R, Frazier SR, Moss RH, Stoecker WV. Enhancements in Localized Classification for Uterine Cervical Cancer Digital Histology Image Assessment. Journal of Pathology Informatics. 2016; 7:51. DOI:10.4103/2153-3539.197193.

[18] Guo P, Banerjee K, Stanley RJ, Long R, Antani S, Thoma G, Zuna R, Frazier SA, Moss RH, Stoecker WV. Nuclei-Based Features for Uterine Cervical Cancer Histology Image Analysis with Fusion-Based Classification. IEEE Journal of Biomedical and Health Informatics. 2016 (November); 20(6):1595-1607.

[19] De S, Stanley RJ, Lu C, Long R, Antani S, Thoma G, Zuna R. A Fusion-Based Approach for Uterine Cervical Cancer Histology Image Classification. Computerized Medical Imaging and Graphics. 2013;37(7-8):475-487.

[20] Corinna Cortes, Vladimir Vapnik. “Support-Vector Networks”, Journal of Machine Learning, v.20 n.3, p.273-297, Sept. 1995.

[21] Rosenblatt, Frank. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. No. VG-1196-G-8. Cornell Aeronautical Lab Inc. Buffalo NY, 1961.

[22] Rao, C. Radhakrishna. "The utilization of multiple measurements in problems of biological classification." Journal of the Royal Statistical Society. Series B (Methodological) 10, pp. 159-203, Jan 1948

[23] Ho, Tin Kam. "Random decision forests." In Document analysis and recognition, 1995- proceedings of the third international conference on, vol. 1, pp. 278-282. IEEE, 1995.

VITA

Sri Venkata Ravitej Addanki was born in Yanam, Puducherry, India. He received a Bachelor of Technology degree in Electronics and Communications Engineering from Pondicherry Engineering College (PEC), Puducherry, India. He worked as an IT Analyst at Tata Consultancy Services, Chennai, India. He received a Master of Science degree in Computer Engineering at Missouri University of Science and Technology in May 2018. His interests mainly included computer vision, machine learning and deep learning.