Abstract

Cervical cancer is a type of cancer that occurs in the cells of the cervix, the lower part of the uterus that connects to the vagina. Various strains of the human papillomavirus (HPV) play a role in causing most cervical cancer. A Papanicolaou test (pap smear test) is the most reliable screening method for identifying cancerous cells. Pathologists visually examine the slide after performing the test to look for malignancy. But this is a time-consuming and challenging task, often limited by the availability of certified professionals. Additionally, the diagnostic efficacy of visual screening is compromised by the increased workload of pathologists, leading to decreased diagnostic accuracy. Thus, a computer-aided diagnosis system is essential in order to enhance the quality of the Pap test results and enable early diagnosis. Nuclei segmentation is a crucial step in automated cervical cancer detection, as the nucleus carries essential information about the alterations occurring during the disease's progression. Thus, automated segmentation of nuclei can significantly and ultimately influence the diagnostic results. Recent surveys found that neural networks outperformed various traditional segmentation methods. We conducted experiments on the Herlev dataset using Unet with ResNet and EfficientNet encoders, PAN (Pyramid Attention Network), and LinkNet with a ResNet encoder. For each of these models, our results showed IOU scores of 0.76, 0.938, 0.79, and 0.69, respectively. The proposal can help experts correctly assess cervical cell lesions and provide better healthcare for patients.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cervical cancer is one of the most common causes of death among females. It is the uncontrolled growth of cancer cells in or on the cervix. Since this cancer is slow-growing, it provides better chances for early detection and effective treatment. Cervical cancer is associated with high-risk strains of the human papillomavirus (HPV). Precancerous lesions that eventually advance to cervical cancer can be identified by cytologic screening of cervical cells. The Papanicolaou test (pap smear test), a cervical cancer screening test in which cells from the outside and inside of the cervix are used as samples for testing, locates HPV-induced changes that have the potential to progress to cancer as well as detects cervical cancer cells. Pap smear slides are obtained after a pap smear examination. Pap smear slides contain cell images that consist of nuclei, cytoplasm, etc. The shape, colour, and texture of the nuclei are abnormal when an infected cervical cell is examined. The cancer cells show nucleus enlargement and irregularity. A pathologist visually examines the slide, which is a laborious and time-consuming process due to the limited number of qualified pathologists and the high workload they face. Therefore, automating the screening process would yield faster and more

accurate results, leading to a decrease in cervical cancer mortality rates. Nuclei segmentation is a crucial step when implementing an automated system, as the nucleus carries essential information about the alterations occurring during the disease's progression. Hence, by automating the segmentation of nuclei, we can enhance the performance of the classification model, resulting in a faster and more efficient diagnosis of cervical cancer. Medical image segmentation plays a significant role in computer-aided diagnosis of diseases as it gives a clear picture of the images and hence increases the accuracy and efficiency of CAD. There are several techniques and models that have been developed to perform this task of segmentation. Among them, the early approaches were based on edge detection, machine learning, template matching techniques, etc. But using these traditional methods, the segmentation task was still a challenging one due to the difficulty in representing the features of images. The development of deep learning techniques led to the achievement of hierarchical feature representation of images and also provided excellent segmentation results. Some popular medical image segmentation tasks include cell nuclei segmentation, colorectal cancer segmentation, liver, and liver tumour segmentation, lung, lung tumour segmentation, etc.

In the initial stage of our research, we implemented the Unet architecture using ResNet and EfficientNet encoders. Our method was tested on the Herlev dataset. The nuclei weren't precisely segmented. In the next stage, we improved the preprocessing of the images, performed hyperparameter tuning on the implemented models, and compared these models to LinkNet with the ResNet encoder and PAN (Pyramid Attention Network). The Unet with EfficientNet encoder provided the best results.

Chapter 2

Problem Definition

The problem definition of the project is described below:

1. Given a slide image from the Herlev Pap smear dataset as input, our objective is to localize the nucleus within the image using a deep learning approach.

2. Comparison of different deep learning-based approaches for accurate automated nuclei segmentation in the Herlev Pap smear dataset.

Chapter 3

Related Work

Nuclei segmentation is a crucial step for cancer detection, grading, and prognostic diagnosis. This is because changes in tissue properties and cell morphology during disease progression can alter the appearance of nuclei[7]. However manual segmentation of nuclei is time-consuming and prone to errors. Therefore automating the nuclei segmentation eases the process and can greatly benefit medical professionals.

Deep learning emerges as a powerful tool in the medical field for different purposes like medical imaging, and other health care applications. Several researchers have explored the utility of deep learning models to automate nuclei segmentation. This part further discusses the current state-of-the-art methods for automated nuclei segmentation in cytology procedures.

[1] proposes a combination of Nested Unet and EfficientNet model for the nuclei segmentation of the cell images in a cryonuseg dataset. This model performed better when compared to other Unet models for this dataset. The Cryonuseg dataset has images from 10 human organs and is used mainly for nuclei segmentation of cryosectioned H&E-Stained Histological images. The

images in this dataset are of size 512 x 512. In this model, they used EfficientNet as the encoder for the nested Unet model. The encoder part will extract the feature map, and the decoder part will upsample the feature map to a 2-D segmentation map.The building blocks of EfficientNet is MBConv (Mobile Bottleneck Convolution) which has squeeze and excitation layers for optimization. It is the most compact and high-performing model and thus can extract the feature map efficiently.The model's performance was analyzed using three metrics Dice Score metric, PQ (Panoptic Quality) metric, and AJI (Aggregated Jaccard index) metric.The average Dice score obtained is 0.929, average AJI is 0.473 and average PQ is 0.503. Improvements can be made to the model by enhancing its architecture and increasing the number of training inputs using GAN (Generative Adversarial Networks). These can be used as the future scope of improvement of the segmentation model.

[2] proposes a method of automated cervical nuclei segmentation using deformable multipath ensemble model(D-MEM). For detecting cervical cancer, the morphological properties of the cell nuclei are analysed. The approach adopts a U-shaped convolutional network as a backbone network, in which dense blocks are used to transfer feature information more effectively. To increase the flexibility of the model, deformable convolution was used which deals with different nuclei irregular shapes and sizes. To reduce the predictive bias, they further construct multiple networks with different settings, which form an ensemble model. The skip connection allows the gradient to flow directly through different layers, which maintains the information magnitude. To enhance the connection between layers, dense blocks are introduced. As the abnormal nuclei exhibit different shapes, the traditional Unet model cannot deal with varying shapes. Thus, deformable convolution is introduced to

enhance the transformation modelling capability.

[3] Automatic detection of colon cancer is implemented in the present work through segmentation and classification of the abdominal region and polyp identification found over the walls of the colon. The digital image processing technique has helped to identify the colonic region in abdominal 2D CT images. They have used The TCIA database contains 825 cases of patients,and 941771 CTscan images are available with detailsof.xls and the format of DICOMimages. The convolutional residual neural network rightly triggers convolutional encoding and decoding actions. Choosing a small pattern (kernel) and recognizing its occurrence from the input image is carried out through convolution. Filters of the triconvolutional layer are arranged in 1x1, 3×3, and 1×1 fashion, respectively. An activating unit called ReLu (rectified linear unit) follows each of these convolutional layers. In between the convolution and ReLu layers lies the batch norm layer, whose purpose is to minimize the internal covariance shift and to speed up the training process.

[4] proposes a multi-task network based on U-net with ResNet-34 encoder for the segmentation process. The model proposed is applied on four datasets: the 2014 ISBI dataset, BNS, MoNuSeg, and nucluesSeg dataset.Also, context encoding layers are added to the encoding layers of U-net which identify the contextual information of different resolutions. Predicting the nuclei region is the primary task. The auxiliary task is to predict the nucleus boundaries. Predicting the boundaries improves the performance of the primary task. An attention learning module is also introduced behind each skip connection in the network. This helps in learning focused attention. The context encoding layer was developed based on the dense atrous convolution (DAC) block

proposed by [5]. To recognize the importance of each branch of the block, a Squeeze and Excitation (SE) layer is added, which denotes channel attention. The original features are also added to the layer along with the results. A codec block is proposed to obtain the key items and query items in the attention mechanism as the attention weight is determined by the correlation between key (critical features of the value items) and query items.

[6]This paper proposes a novel approach for segmentation of cervical nuclei that combines fully con-volutional networks (FCN) and graph-based approach (FCN-G). FCN is trained to learn the nucleus high-level features to generate a nucleus label mask and a nucleus probabilistic map. The mask is used to construct a graph by image transforming. The map is formulated into the graph cost function in addition to the properties of the nucleus border and nucleus region. The prior constraints regarding the context of nucleus-cytoplasm position are also utilized to modify the local cost functions. The globally optimal path in the constructed graph is identified by dynamic programming. Validation of this method was performed on cell nuclei from Herlev Pap smear dataset. The method shows a Zijdenbos similarity index (ZSI) of 0.92 ± 0.09.

[7]This paper presents a method for multi-class cell segmentation into Nuclei and Cytoplasm regions. Using the Herlev dataset for evaluation, this work achieved good performance by using state-of-the-art classification architectures such as EfficientNet combined with Feature Pyramid Networks to complete the segmentation task. Performance metrics for both classes show that the approach is robust enough to complete the task. The model achieved a 0.91 F1 score, 0.85 IoU, 0.91 Precision, 0.92 Recall, and 0.96 Specificity

as class average with a very low standard deviation, validated with a 5-fold cross-validation.

[8]This paper presents an approach to whole cervical cell segmentation using a mask regional convolutional neural network (Mask R-CNN). ResNet10 is used to make full use of spatial information and prior knowledge as the backbone of the Mask R-CNN. Herlev Pap Smear dataset was used for evaluation. For segmentation, when Mask R-CNN is applied on the whole cell, it outperforms the previous segmentation method in precision (0.92±0.06), recall (0.91±0.05) and ZSI (0.91±0.04).

Chapter 4

Requirements

This project involves both hardware and software components, and the specifications for each are listed below.

4.1 Hardware

- A computing system equipped with adequate GPU processing capabilities.

4.2 Software

- Google Colab
- Necessary Python Libraries

The following libraries and software packages were utilized:

Numpy: Used for data analysis and scientific computing. Large, multidimensional arrays and matrices are supported, along with a variety of mathematical operations for using these arrays.

matplotlib: Offers a variety of tools for building high-quality visualizations, simulations and graphics.

time: For timing and simulation counters

copy: A method that is used on objects to create copies of them

torchvision: Python package that is part of the PyTorch deep learning framework. Offers set of tools and utilities for working with image and video data in PyTorch. Consists of popular datasets, model architectures, and common image transformations for computer vision.

tqdm: Library that is used for creating Progress Bars and percentage displays for long-running tasks.

PIL: Stands forPython Imaging Library. Adds support for a variety of image file formats by enabling opening, manipulating, and saving. PIL provides a number of image processing capabilities, including image resizing, cropping, rotating, and filtering.

random: Library used for the generation of random numbers.

math: Provides functions for mathematical operations including functions for trigonometry, logarithmic, and exponential functions.

os: Module used for OS interaction

pandas: Library for data manipulation, analysis of results, and statistical plots. Consists of data structures and operations for working with structured data, such as data frames and series.

Chapter 5

Proposed System

In this study, our objective was to develop an accurate and efficient deep-learning model for automating nuclei segmentation in Cervical cancer cells. To achieve this goal, we compared the performance of four different models - Unet with Resnet encoder, Unet with EfficientNet encoder, PAN with Resnet encorder, LinkNet with Resnet encoder. We trained and tested these models on a dataset of Pap smear images and evaluated their performance using the Dice score and IOU score. In the following section, a system diagram (figure 5.1) is provided for the visual representation of the proposed system.

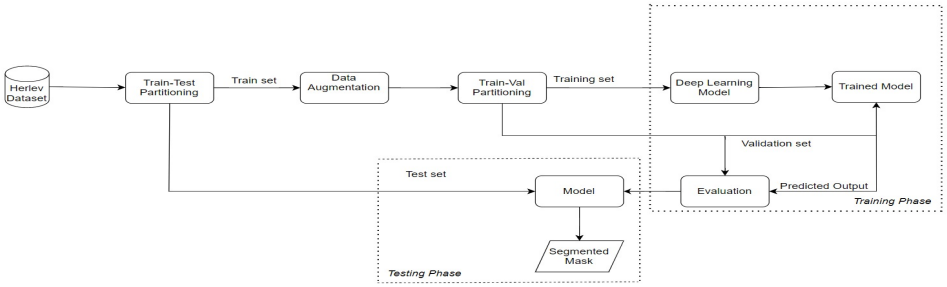
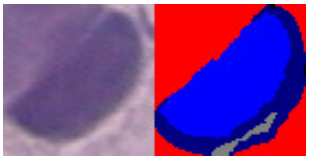


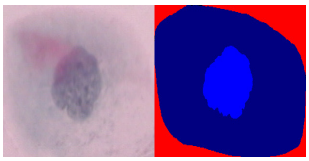
Figure 5.1: System Diagram

5.1 Dataset

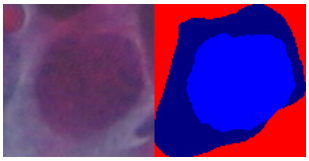
The dataset we used for this study is the Herlev dataset. It contains pap smear images with their corresponding masks. These images were distributed among 7 files. Each file represents the 7 classes (Figure 5.2) of cervical nuclei which are carcinoma in situ (Figure 5.2a), light dysplasia (Figure 5.2b), moderate dysplasia (Figure 5.2c), normal columnar (Figure 5.2d), normal intermediate (Figure 5.2e), normal superficial (Figure 5.2f) and severe dysplasia (Figure 5.2g). We merged these files into a single file, containing a total of 917 images. We then divided the dataset into training and test set. After splitting there were 317 images on the training sets and 600 images on the test sets. We augmented the training set to make our model learn the precise details.



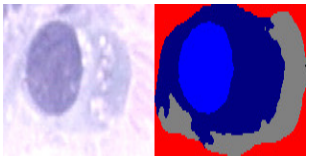
(a) Squamous cell carcinoma in situ



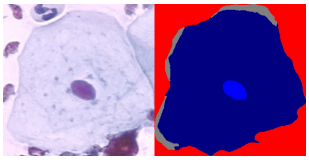
(b) Light dysplastic



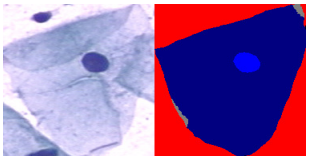
(c) Moderate dysplastic



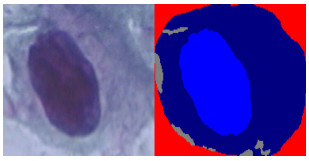
(d) Normal columnar



(e) Normal intermediate



(f) Normal superficial



(g) Severe squamous non-keratinizing dysplasia

Figure 5.2: 7 classes

5.1.1 Preprocessing of Masks

The original mask of the pap smear dataset comprised of four colors. Light blue indicates the nucleus, while dark blue represented the cytoplasm. The red color indicated the background and an unknown grey region was present in some masks. To preprocess the masks we first converted them into RGB format and used color thresholding to separate the different regions. To remove the gray region, we created separate masks for the nucleus, cytoplasm, and background. Finally, these masks were merged into a single array to get the new preprocessed masks where red indicates the nucleus, green indicates the cytoplasm and blue represents the background. The preprocessed masks are shown in Figure 5.3

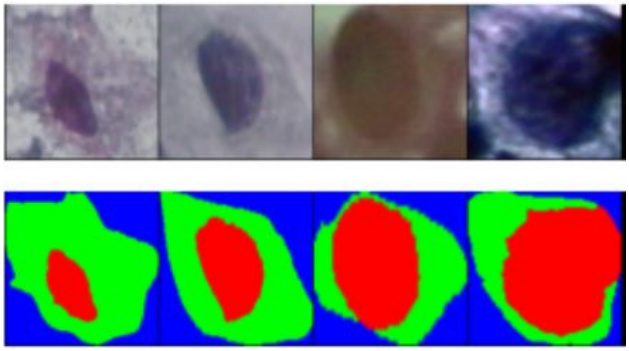


Figure 5.3: Preprocessed masks

5.2 Augmentation

Data augmentation is the process of artificially increasing the size of a dataset by generating different versions of the existing data. After train test splitting we had 340 images for training. A deep learning model generally works well when it has a huge amount of data. In general, the more data we have better will be the performance of the model. The problem with the lack of

a good amount of data is that the deep learning model might not learn the patterns or the functions from the data and hence it might not perform well. So the training set was augmented using different data augmentation techniques. For semantic segmentation tasks, we need to augment both the input image and output masks. The augmentation techniques used to enhance the dataset are horizontal flip, vertical flip, rotation, sharpness, grayscale, equalize, affine transform, posterize, brightness, hue, contrast, and saturation. torchvision.transforms module contains many transform techniques to manipulate images. After augmentation 6357 images were obtained for training. The torchvision.transforms module from the PyTorch library provides all the functions for performing different data augmentation techniques.

HorizontalFlip: It involves producing a mirrored version of an original image by reflecting it across a vertical axis. The RandomHorizontalFlip() function is used to perform HorizontalFlip in PyTorch library.

VerticalFlip: It involves creating a mirror image of an original image by flipping it vertically along the horizontal axis. It is performed by using the RandomVerticalFlip() function provided by PyTorch library.

Rotation: Images are rotated by a predetermined angle. The RandomRotation() function provided by PyTorch library is used to perform this technique

AdjustSharpness: This technique modifies the sharpness of an image by adding or removing some level of sharpness. It is adjusted by applying a blur or sharpen filter. The level of adjustment is determined by a factor between a specified range that controls the intensity of the blur or sharpen

filter. RandomAdjustSharpness() is the function used for this.

Grayscale: It involves converting a color image to grayscale by removing the color channels (red, green, and blue) and keeping only the brightness channel. The Grayscale() function of the PyTorch library performs this technique. The Number of input channels is a function parameter.

ColorJitter: It is an image transformation that randomly changes the brightness, contrast, saturation, and hue of an image. ColorJitter() function performs this augmentation technique. Brightness, contrast, saturation, and hue are taken as input parameters of this function.

Equalize: It enhances the contrast of an image by balancing the histogram of the image. RandomEqualize() function performs this augmentation technique.

Affine: Affine transformations can be used to perform a variety of geometric transformations on images, such as rotation, scaling, and translation. RandomAffine() function of the torchvision.transforms module performs this function.

Posterizarion: It involves reducing the number of colours in an image. In the end, the image has fewer colour shades and looks more like a poster. RandomPosterize() function of the torchvision.transforms module performs this function. Bits is an input parameter for this function that represents the number of bits for each color channel to keep.

5.3 Deep learning models

5.3.1 Unet with ResNet encorder

In the area of medical image segmentation, the neural network architecture, UNet is widely used. We are proposing to use a UNet architecture with a ResNet-based encoder for nuclei segmentation in single-cell pap smear (cytology) images. ResNet (Residual Neural Network) is an artificial neural network architecture built of a series of residual blocks (ResBlocks) (as in Figure 5.5) that uses skip connections to jump over some layers, enabling the construction of much deeper feedforward neural networks with hundreds of layers. The Unet architecture contains the encoder part, decoder part, and skip connections (see Figure **??**. ResNet can be employed as the encoder/downsampling component of the Unet, utilizing the advantages of both models to produce an effective outcome.

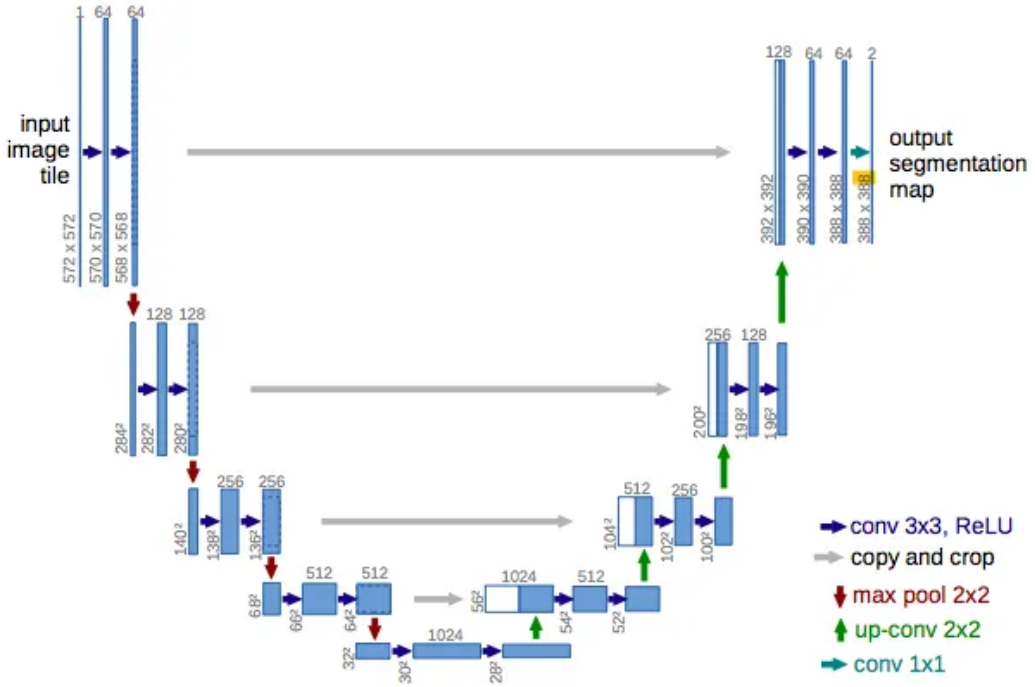


Figure 5.4: Unet Architecture

Architecture

The encoder or downsampling path of the Unet architecture captures the context of the image, producing feature maps. The decoder or upsampling path is used to enable precise localization using transposed convolutions. Transposed convolutions or deconvolutions add pixels between and around the existing pixels. The encoder section of the UNet is substituted with a pre-trained ResNet encoder.

The ResNet architecture is constructed by connecting a series of Residual blocks (ResBlocks). They contain two connections from its input, one passing through a series of convolutions, activation functions (relu), and batch normalizations, and the other skipping this path. These are called cross or skip connections. (see Figure 5.5).

These skip connections address the vanishing gradient problem, as a supplementary path is provided for the gradient to flow through, allowing for better information propagation during backpropagation. It also enables the model to learn an identity function, which helps to maintain the accuracy of the output as the depth of the network increases.

Pre-trained ResNet-50, a 50-layered (48 convolutional layers, one MaxPool layer, and one average pool layer) deep learning model is used as the encoder of the Unet. It is a part of the ResNet family of models [11]. ResNet-50 and ResNet 101 provided the best results among all the other ResNet models which include ResNet-18 and ResNet-34. ResNet-50 was selected due to its relatively lower parameter count. The model is pre-trained on the ImageNet dataset. As a result, the model can use the pre-trained features from the large-scale ImageNet dataset, which could help it perform more effectively. ResNet-50 achieves a top-5 validation error of 5.25%.

The 2-layer ResBlocks in ResNet34 are replaced with 3-layer bottleneck

residual blocks in Resnet-50 allowing it to learn more complex features. These blocks use 1×1 convolutions enabling much faster training of each layer. (refer to Figure 5.5)

The integration of a ResNet50 encoder into the UNet architecture for image segmentation results in a powerful and effective neural network. This architecture has the ability to learn both high-level and low-level features of the input image, enabling it to achieve high levels of accuracy even when trained with a limited amount of data.

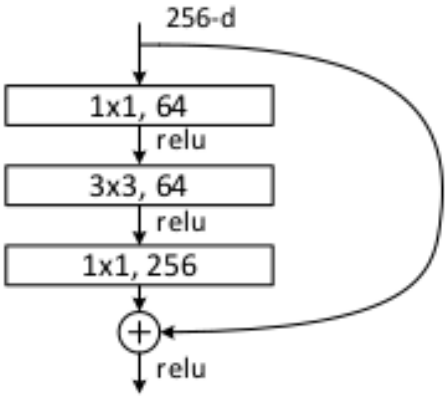


Figure 5.5: ResBlocks

5.3.2 Unet with EfficientNet encorder

Unet with Efficientnet encoder is another model we proposed, combining two powerful architectures. The EfficientNet architecture is known for its efficiency and state-of-the-art performance in various computer vision tasks. It has been used as a pre-trained model for transfer learning and yielded better performances. Also as the name suggests it is computationally efficient and has achieved a state of art result on the ImageNet dataset which is 84.4% accuracy. Unet architecture as discussed in the previous section is widely used in medical image segmentation and consists of a decoder and encoder part. So in this proposed model, we replaced the encoder of Unet with efficient

net architecture to achieve the benefits of both models and thus improve the performance of the segmentation.

Architecture

EffiecientNet is a family of convolutional networks which has versions from B0 to B7 based on the number of parameters. We experimented with these different versions for the encoder and found that all of them achieved comparable performance, however, we chose the B0 version as it has less number of parameters.

EffievientNetb0 has 5.3 million parameters and consists of mainly 7 blocks while starting with a 3x3 convolution layer[10]. Each block contains Mobile Inverted Residual Bottleneck layers(MBconv layers) to which squeeze and excitation layers are also added. MBconv layers have three main components (see figure 5.6)

Depthwise convolution: This type of convolutional layer applies a single filter to each input channel which results in the reduction of the number of parameters and makes the feature extraction computationally effective.

Pointwise Convolution: These are the convolutional layers with a 1x1 filter to perform a linear transformation of the input tensor. In efficientnetb0 this layer of MBconv is to increase the number of channels after depthwise convolution.

Skip Connection: A skip connection is established from the output layer of the block to the input layer to add information. It helps to improve the flow of gradients through the network.

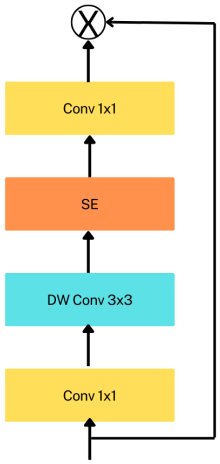


Figure 5.6: MBConv Layer

The EfficientNetb0 architecture also utilizes a compound scaling approach that simultaneously scales up the network in terms of width, depth, and resolution, ultimately resulting in improved performance. Depth scaling involves adding more layers to the network, to make it deeper. Width scaling increases the number of filters in each layer, making the network wider and resolution scaling involves increasing the size of input image, which results in increase in number of pixels. The scaling coefficients are identified empirically through grid search over the scaling values. The Architecture of the EfficientNetb0 is given in Figure 5.7

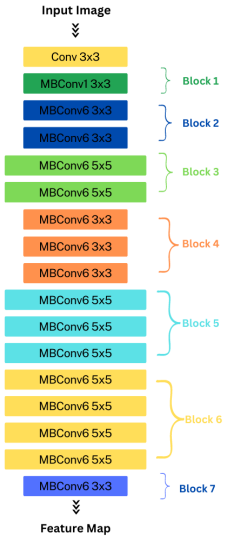


Figure 5.7: EfficientNet b0 Architecture

The proposed model was obtained by combining the EfficientNet encoder with the Unet architecture, as illustrated in Figure 5.8. The EfficientNet b0 encoder extracts the high level features from the input image and the decoder of the Unet transforms these features back into the pixel space to produce the output segmentation map, here the map with segmented nucleus. We utilized a pre-trained model from the PyTorch segmentation models and achieved promising performance. Further details regarding the training process will be discussed in the next sections.

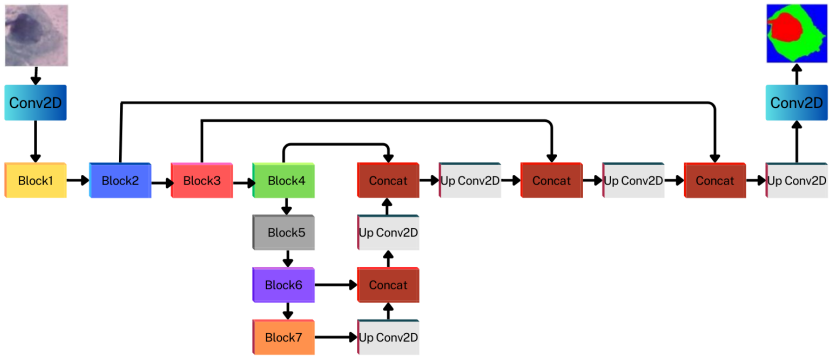


Figure 5.8: Unet with EfficientNet b0 Architecture

5.3.3 PAN(Pyramid Attention Network) with ResNet encorder

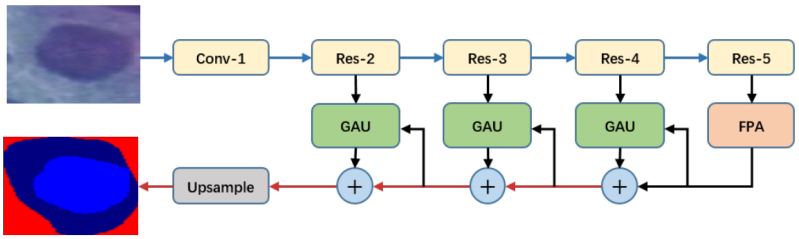


Figure 5.9: PAN architecture

A Pyramid Attention Network(PAN)[10] exploits the impact of global contextual information in semantic segmentation. Different from most existing works, attention mechanism and spatial pyramid is combined to extract precise dense features for pixel labeling instead of complicated dilated convolution and artificially designed decoder networks. A Feature Pyramid Attention module is introduced to perform spatial pyramid attention structure on

high-level output and combine global pooling to learn a better feature representation, and a Global Attention Upsample module is introduced on each decoder layer to provide global context as a guidance of low-level features to select category localization details

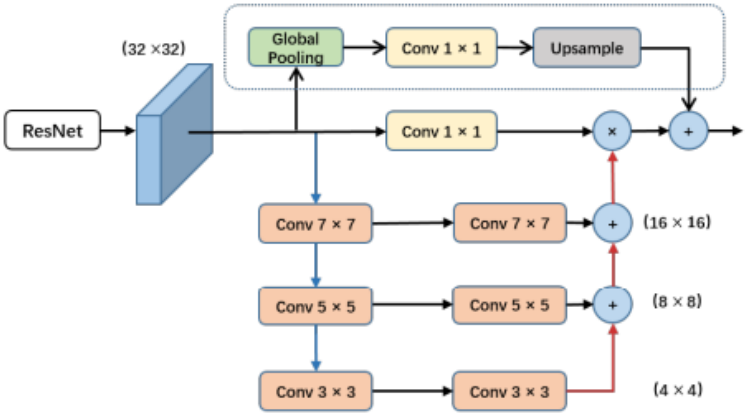


Figure 5.10: Feature Pyramid Attention Structure

Feature Pyramid Attention(FPA)

The pyramid attention module fuses features from under three different pyramid scales by implementing a U-shape structure like Feature Pyramid Network. To better extract context from different pyramid scales, it has 3×3, 5×5, 7×7 convolution in pyramid structure respectively(Fig 5.10). Since the resolution of high-level feature maps is small, using large kernel size doesn't bring too much computation burden. Then the pyramid structure integrates information of different scales step-by-step, which can incorporate neighbor scales of context features more precisely. Then the origin features from CNNs is multiplied pixel-wisely by the pyramid attention features after passing through a 1×1 convolution. The model also has global average pooling branch adding with the output features.

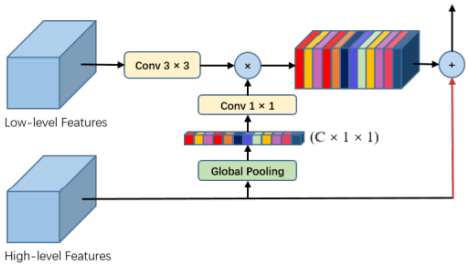


Figure 5.11: Global Attention Upsample module structure

Global Attention Up-sample(GAU)

Global Attention Up-sample module performs global average pooling to provide global context as a guidance of low-level features to select category localization details. A 3×3 convolution is present on the low-level features to reduce channels of feature maps from CNN's(Fig 5.11). The global context generated from high-level features is through a 1×1 convolution with batch normalization and ReLU non-linearity, then multiplied by the low-level features. Finally high-level features are added with the weighted low-level features and up-sampled gradually. This module deploys different scale feature maps more effectively and uses high-level features provide guidance information to low-level feature maps in a simple way.

5.3.4 LinkNet with ResNet encorder

LinkNet is a deep convolutional neural network (CNN) architecture that was proposed in 2017 by Chaurasia et al. It was designed for semantic segmentation of images, which is the task of assigning a label to each pixel in an image. LinkNet is specifically designed to be efficient, both in terms of memory usage and training time, while still achieving high accuracy on this task.

Here we have used linknet with resnet 34 encoder.The ResNet-34 encoder is pretrained on the ImageNet dataset. The model is designed to segment

images into 3 classes and the output of the model will be passed through a Softmax activation function.Linknet can use either softmax or sigmoid activations but using softmax provided better performance. Then while completing 40 epochs it got iouscore of 0.79 in the validation set and 0.69 in the test set.

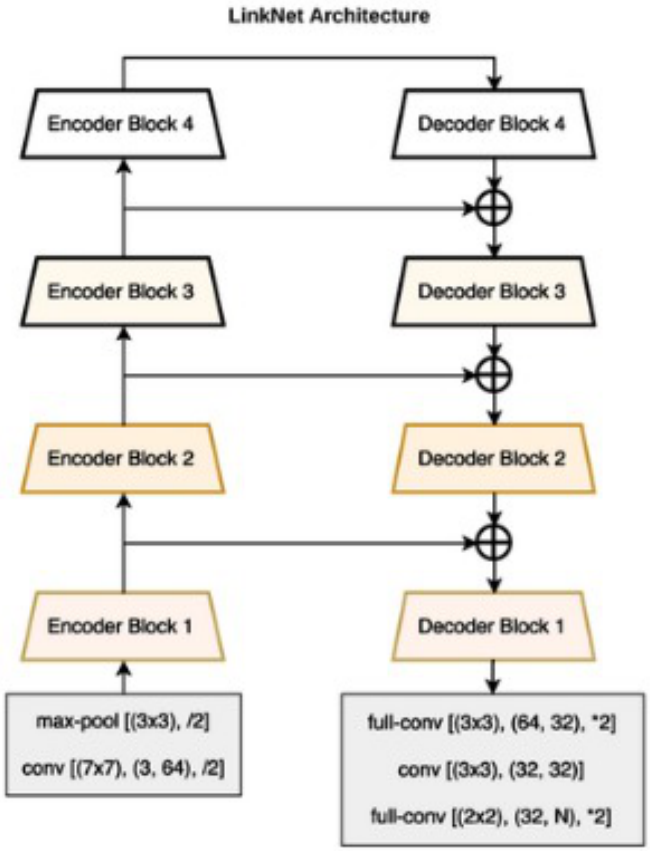


Figure 5.12: LINKNET architecture

Linknet Architecture

The architecture of LinkNet is based on an encoder-decoder structure, where the encoder extracts high-level features from the input image and the decoder produces the segmentation map. The key innovation of LinkNet is the use of "skip connections" that connect the encoder and decoder blocks at

corresponding levels. These skip connections allow the model to preserve spatial information and improve segmentation accuracy, while also reducing the number of trainable parameters and improving training efficiency. LinkNet uses a modified version of the ResNet encoder architecture, which includes batch normalization and a modified activation function. The decoder includes upsampling layers to restore the resolution of the feature map, and a set of 1x1 convolutions to reduce the number of channels and produce the final segmentation map. Overall, LinkNet-34 is a powerful and efficient semantic segmentation architecture that leverages the strengths of both the LinkNet and ResNet-34 models to produce accurate and fast segmentation results.

5.4 Model Training

Training the proposed models is a crucial step in developing deep learning techniques as it enables the model to learn the features and patterns in the data and perform accurate predictions. It includes optimizing the parameters to minimize the loss function and improve the performance. This section discusses the methods we employed during the training and provides insights into the training process for each model.

Early Stopping: This technique is implemented to prevent overfitting. It is a regularization technique that suspends the training process once the validation dataset performance starts to deteriorate. The performance of the model is tracked using the validation loss. The patience parameter which specifies the maximum number of epochs that the model is allowed to continue training without seeing any improvement in the validation loss is set to 3 and the delta parameter which specifies the minimum amount of improve-

ment that is considered significant is taken as 0.0001.

Batch Normalization: A pre-processing technique used to standardize data is normalization. Batch Normalization is a normalization technique done between the layers of a Neural Network instead of in the raw data. This technique has been employed to improve the model's learning rate and prevent overfitting.

Optimizer: Adam optimizer is used. This optimization algorithm is an extension of the stochastic gradient descent (SGD) algorithm. Adam optimizer modifies the learning rate for each network weight separately, unlike SGD, which maintains a single learning rate. It is more computationally efficient than other optimization algorithms.

Activation Function: Softmax activation function is used as the activation function for all the implemented approaches. It is a mathematical function that creates a probability distribution of possible outcomes from a vector of real numbers of the same size. Mathematically softmax is defined as, $S(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$

The dataset is divided into 80% training, 10% validation, and 10% test sets

5.4.1 Unet with ResNet encoder

The encoder part of the model uses the ResNet-50 architecture that has been pre-trained on the 'ImageNet' dataset. The model is trained for 60 epochs. Hyperparameter tuning is performed. The learning rate is set to 0.0001. Early stopping and batch normalization are implemented as mentioned above. Adam optimizer and softmax activation functions are used.

The validation batch size is set to 32 and the training batch size is set to 32. These optimum values that achieve a compromise between the trade-off amongst model performance, computational resources, and convergence time required trial and error and experimentation with all of these hyperparameters.

5.4.2 Unet with EfficientNet encorder

The proposed model uses the pre-trained EfficientNet b0 as the encoder with the encoder weight initialized to 'ImageNet'. The model is trained for 60 epochs with an initial learning rate of 0.0001, which is reduced to 0.00001 after 25 epochs. The learning rates are finalized after multiple runs for hyperparameter tuning. The validation set is processed in batches of 4, while the training set is processed in batches of 8. Early stopping, batch normalization, optimizer, and activation functions are selected as same as the parameters mentioned at the beginning of this section.

5.4.3 PAN(Pyramid Attention Network) with ResNet encorder

For this model resnet34 is used as the encoder and ImageNet is given as the encoder weight initially. As it is a multiclass segmentation, softmax is used as the activation function. Adam is used as the optimizer. The model is run for 40 epoches. Early stopping condition was given, so the model stopped at the 35th epoche.

5.4.4 LinkNet with ResNet encorder

The model uses ResNet encoder which are pre-trained weights from the ImageNet dataset. The activation function used here is softmax. The model is trained for 40 epochs and applied hyperparameter tuning. Early stopping and batch normalization are implemented.

Chapter 6

Result and Analysis

We are able to localize the nucleus within the slide image from the Herlev Pap smear dataset using deep-learning approaches. Comparative analysis is performed on the four models including Unet with ResNet and EfficientNet encoders, PAN (Pyramid Attention Network), and LinkNet with a ResNet encoder. The model's performance is evaluated using widely accepted metrics such as precision, recall, and F1 score to assess their overall effectiveness in accurately segmenting nuclei.

Dice loss is a widely-used loss function used to calculate the similarity between images and is similar to the Intersection-over-Union (IoU) heuristic.

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

The aforementioned equation is that of the dice coefficient, also known as the dice loss, and the variables p_i and g_i stand for pairs of corresponding pixels representing the prediction and ground truth, respectively. The range of the loss function is between 0 and 1. The Dice loss metric takes into account both local and global loss information, which is a crucial factor in achieving

high levels of accuracy.

One of the most frequently employed metrics in semantic segmentation is the **Intersection-Over-Union (IoU)**, also referred to as the Jaccard Index. IoU is defined as the area of union between the predicted segmentation and the ground truth divided by the area of overlap between the two. This metric has a range of 0 to 1, where 0 denotes no overlap and 1 denotes perfectly overlapping segmentation. The mean IoU of the image is determined for multi-class segmentation by averaging the IoU of each class.

Pixel accuracy is the percent of pixels in your image that are classified correctly. Pixel accuracy is frequently reported both globally across all classes and separately for each class.

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

Here, a True Positive (TP) denotes a pixel that is accurately predicted to belong to the specified class, whereas a True Negative (TN) denotes a pixel that is accurately identified as not belonging to the specified class. FP and FN denote False Positive and False Negative respectively.

Precision is a metric that quantifies the number of correct positive predictions made. It hence calculates the accuracy for the minority class. It is computed as the ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted.

$$Precision = \frac{TP}{TP+FP}$$

Recall is a metric that quantifies the number of correct positive predic-

tions made out of all positive predictions that could have been made. Unlike precision which only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions.

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

F1 score measures the model's accuracy. It combines the precision and recall scores of a model. This accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1score = \frac{2*Precision*Recall}{Precision+Recall}$$

Zijdenbos Similarity Index (ZSI) is an evaluation metric commonly used in analyzing the performance of medical image segmentation proposed by Alex P. Zijdenbos [13]. ZSI measures the spatial overlap between the predicted segmentation and the ground truth segmentation, by calculating the ratio of the intersection between the two masks to their union.

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

During the initial phase of Evaluation and Result Analysis, we used Unet with ResNet and EfficientNet encoders. The U-net with the ResNet101 encoder provided an accuracy of 78 % and an IoU score of 0.72 on the test dataset. U-net with the EfficientNet encoder provided an accuracy of 75 % and an IoU score of 0.64 on the test dataset. As the accuracy was low, the predicted masks were only able to display the segmented cell, not the nucleus.

In the second phase, we used four different models- Unet with ResNet

and EfficientNet encoders, PAN(Pyramid Attention Network), and LinkNet with ResNet encoder and their performance was evaluated using the above-mentioned evaluation metrics which are IoU sore, Accuracy, precision, recall, f1 score and ZSI (Table 6.1)

Unet with ResNet encoder produced training results with a loss of 0.016, 0.96 IoU score, and accuracy of 96%. The validation performance includes a loss of 0.031, an IoU score of 0.93, and an accuracy of 95%. The dice loss and IoU score of the train and validation set over the epochs are shown in Figure 6.1, and Figure 6.2 respectively.

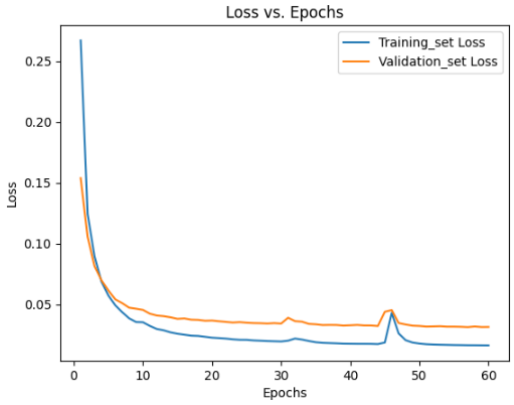


Figure 6.1: Loss vs Epoch - Unet with Resnet Encoder

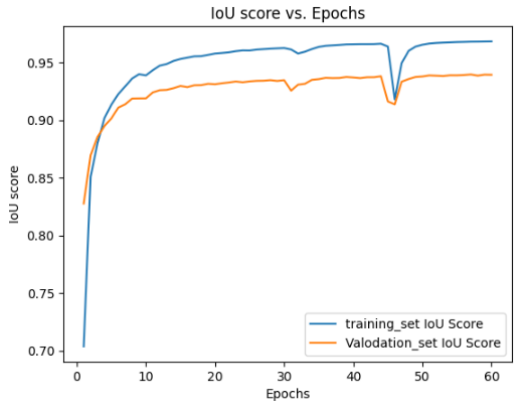


Figure 6.2: IoU score vs Epoch - Unet with Resnet Encoder

The performance of the model on the test data showed a dice loss of

0.1403, an IoU score of 0.76, and an accuracy of 88%. The image, ground truth, and segmented nuclei are displayed in Figure 6.3. The augmentation was enhanced, masks were preprocessed, and substantial hyperparameter tuning was done, but couldn't obtain significant progress in segmenting the nuclei of the test data.

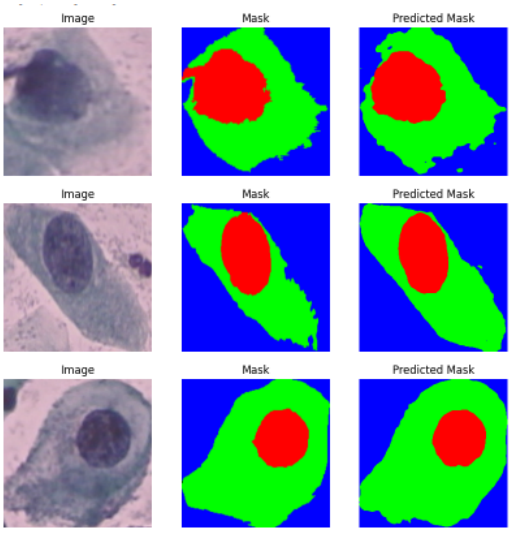


Figure 6.3: Segmented Nuclei

Unet with EfficientNet encoder provided a training performance with loss of 0.02, 0.95 IoU score and 96% accuracy and a validation performance with loss of 0.028, IoU score 0.94 and 95% accuracy. The dice loss and IoU score of train and validation set over the epochs are shown in Figure 6.4, and Figure 6.5 respectively.

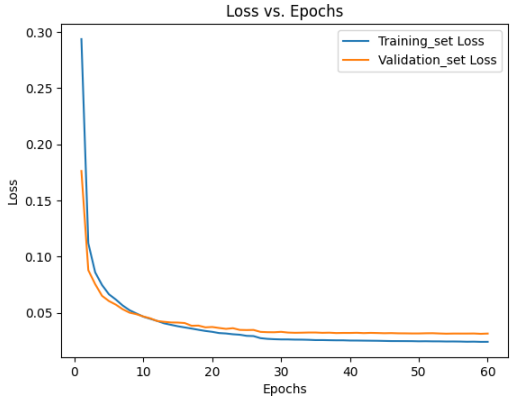


Figure 6.4: Loss vs Epoch- Unet with EfficientNet Encoder

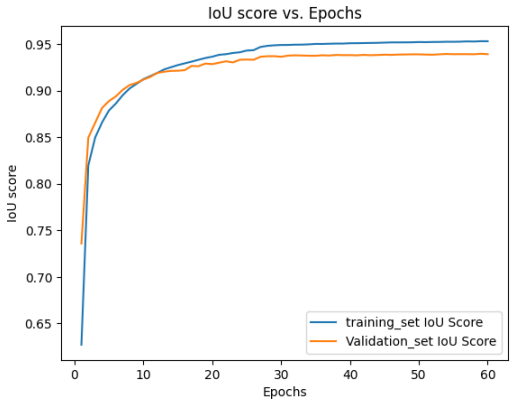


Figure 6.5: IoU score vs Epoch - Unet with EfficientNet Encoder

The test data performance provided a dice loss of 0.031, an IOU score of 0.938, and accuracy of 95%. The image, ground truth, and segmented nuclei are shown in Figure 6.6 for the performance analysis.

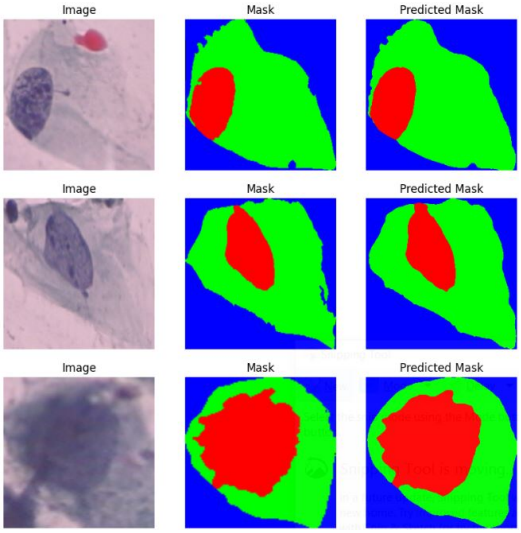


Figure 6.6: Segmented Nuclei

PAN provided an accuracy of 95%, an IoU score of 0.93, and a loss of 0.03. For the validation set the accuracy is 94%, the IoU score is 0.91 and the loss is 0.04. The dice loss and IoU score of train and validation set over the epochs are shown in Figure 6.7, and Figure 6.8 respectively.

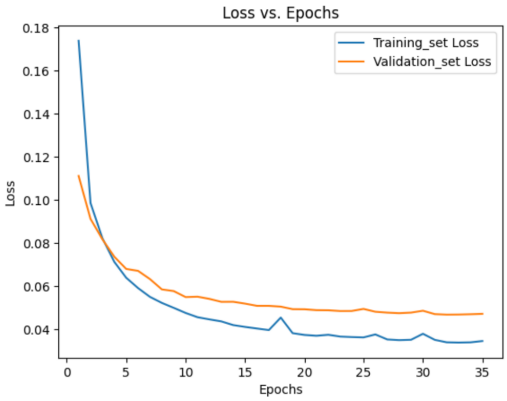


Figure 6.7: Loss vs Epoch- PAN

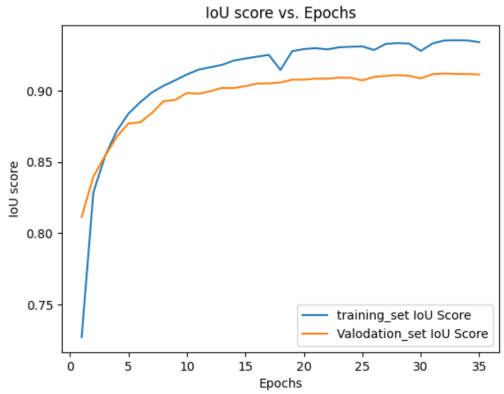


Figure 6.8: IoU score vs Epoch - PAN

The test data performance provided a dice loss of 0.11, an IOU score of 0.79, and an accuracy of 89%. The image, ground truth, and segmented nuclei are shown in Fig 6.9 for the performance analysis.

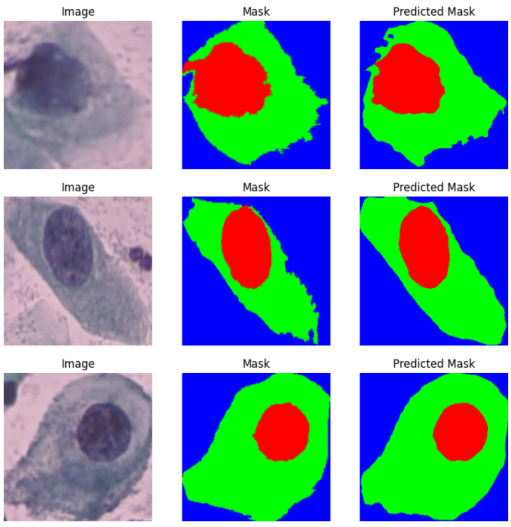


Figure 6.9: Segmented Nuclei

Linknet with Resnet 34 encoder provided a training accuracy of 91% and IoU score of 0.801 and a loss of 0.22. The validation results are of accuracy 90% an IOU score of 0.799 and a loss of 0.22. The dice loss and IoU score of the train and validation set over the epochs are shown in Figure 6.10, Figure6.11 respectively

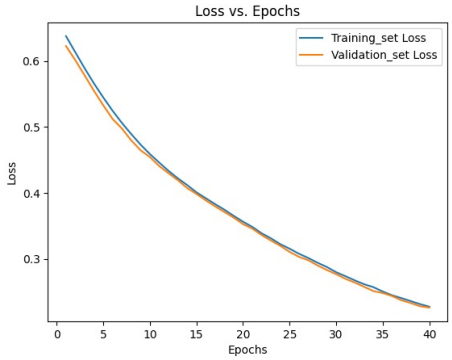


Figure 6.10: Loss vs Epoch - Linknet

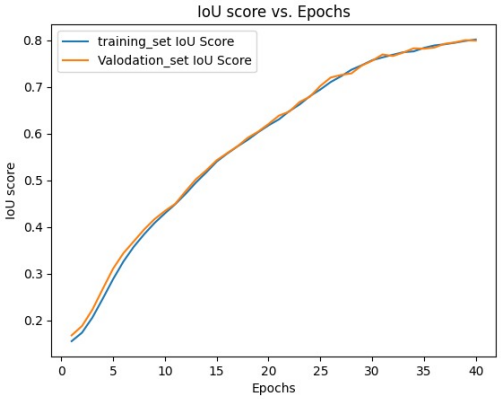


Figure 6.11: IoU score vs Epoch- Linknet

The test data performance provided a dice loss of 0.3, an IOU score of 0.69, and an accuracy of 86%. The image, ground truth, and segmented nuclei are shown in Figure 6.12 for the performance analysis.

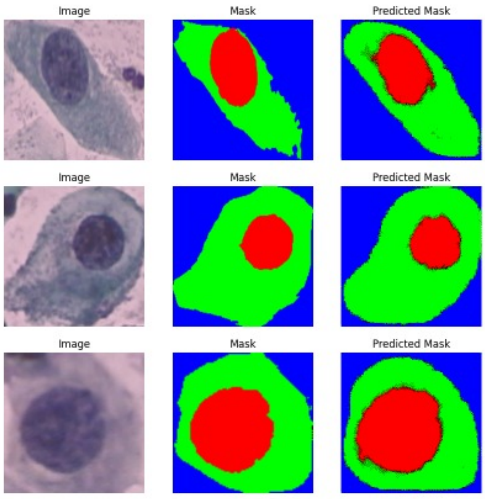


Figure 6.12: Segmented Nuclei

Figure 6.13 illustrates the nucleus segmentation results from four different models implemented, including Unet with ResNet and EfficientNet encoders, PAN (Pyramid Attention Network), and LinkNet with a ResNet encoder. The results demonstrate that the Unet model with the EfficientNet encoder outperforms the other models in terms of precisely segmenting the nuclei.

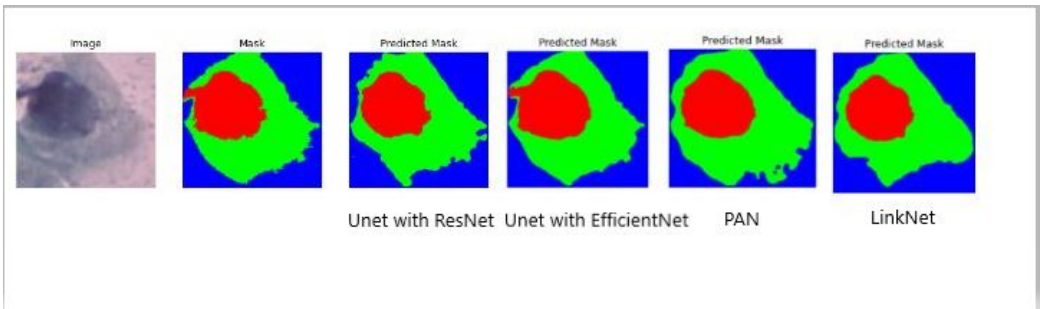


Figure 6.13: Comparison of Segmented Nuclei

From Table 6.1, it is evident that Unet with EfficientNet encoder provides better performance when compared with other models.

Table 6.2 compares the performance of our proposed models with the

state-of-the-art methods.

Methods	Unet with Resnet encoder	Unet with Efficient net encoder	PAN with Resnet encoder	Linknet with Resnet encoder
IoU Score	0.76	0.938	0.79	0.69
Accuracy	88%	95%	89%	86%
Precision	0.67	0.974	0.756	0.667
Recall	0.70	0.976	0.758	0.895
F1 score	0.63	0.975	0.75	0.604
ZSI	0.78	0.938	0.823	0.724

Table 6.1: Evaluation of different model settings.

Methods	Precision	Recall	F1 score	ZSI
UNet + Residual module (with SE block)[3]	0.9724	0.962	-	0.97
UNet + GDLA [7]	0.888 ± 0.11	0.936 ± 0.13	-	0.913 ± 0.09
D-MEM[2]	0.946±0.06	0.984±0.00	-	0.933±0.14
Mask R-CNN[8]	0.92±0.06	0.91±0.05	-	0.91±0.04
Mask-RCNN + LFC-CRF [9]	0.96 ± 0.05	0.96 ± 0.11	-	0.95 ± 0.10
CNN (VGGNet)[10]	0.89	0.91	0.90	-
FCN and FCN-G[6]	-	-	-	0.92 ± 0.09
Unet with Efficient net encoder	**0.974**	**0.976**	**0.975**	**0.938**

Table 6.2: Comparison of the state-of-the-art methods and proposed models.

Chapter 7

Conclusion

Early detection and appropriate treatment are key to lowering the risk of cervical cancer. So the detection of the presence of mutated cells performs a significant role in treating this common cancer among women. The performance of our models, especially the Unet-EfficientNet encoder with an IoU score of 0.93, indicates their potential to improve the efficiency and accuracy of cervical cancer screening. This is especially significant given that manual screening by pathologists is a time-consuming and challenging process due to the limited availability of certified pathologists. By automating the segmentation of nuclei from cervical cell images, our proposed models can serve as a valuable tool to aid pathologists in the screening process, leading to faster and more accurate diagnoses.

In the first phase of our research, we propose Unet architecture with ResNet and EfficientNet encoders to perform the automated segmentation of nuclei from the cell images. The proposed models resulted in an average performance over the data. And as the second phase we compared the performance of four different models- Unet with Resnet encoder, Unet with EfficientNet encoder, PAN(Pyramid Attention Network) with Resnet encoder,

and Linknet with Resnet encoder - for the task of nuclei segmentation. After training and testing each model, we found that Unet- EfficientNet encoder achieved the highest performance with an IoU score of 0.93. This implies that the model is able to segment the nuclei from the cell images almost accurately.

The reported results were obtained on the preprocessed masks, as our models were unable to segment the unidentified grey areas that were only present in some of the ground truth images. Additionally, the ground truth segmentation masks were only provided for individual nuclei, despite the presence of multiple nuclei in the input images. Therefore, future work could explore improving the model to segment multiple nuclei while also addressing the challenge of overlapping nuclei.

References

[1] Cell nuclei segmentation in cryonuseg dataset using nested unet with efficientnet encoder, https://ieeexplore.ieee.org/document/9748537

[2] Automated Segmentation Of Cervical Nuclei In Pap Smear Images Using Deformable Multi-Path Ensemble Model, https://ieeexplore.ieee.org/document/8759262

[3] Nucleus segmentation and classification using residual SE-UNet and feature concatenation approach incervical cytopathology cell images, https://journals.sagepub.com/doi/full/10.1177/15330338221134833

[4] AL-Net: Attention Learning Network Based on Multi-Task Learning for Cervical Nucleus Segmentation, https://ieeexplore.ieee.org/document/9656603

[5] 3D U-Net: A 3D Universal U-Net for Multi-domain Medical Image Segmentation, https://link.springer.com/chapter/10.1007/978-3-030-32245-8_33

[6] Combining fully convolutional networks and graph-based approach for automated segmentation of cervical cell nuclei,https://www.researchgate.net/publication/315492754_Combining_

fully_convolutional_networks_and_graph-based_approach_for_automated_segmentation_of_cervical_cell_nuclei

[7] Cervical Cell Segmentation Method Based on Global Dependency and Local Attention,https://doi.org/10.3390/app12157742

[8] Segmentation and Classification of Cervical Cells Using Deep Learning, https://ieeexplore.ieee.org/abstract/document/8805065

[9] Automatic Segmentation of Cervical Nuclei Based on Deep Learning and a Conditional Random Field https://ieeexplore.ieee.org/abstract/document/8805065

[10] CNN based segmentation of nuclei in PAP-smear images with selective pre-processing https://doi.org/10.1117/12.2293526

[11] Combining fully convolutional networks and graph-based approach for automated segmentation of cervical cell nuclei https://ieeexplore.ieee.org/document/7950548

[12] Deep Residual Learning for Image Recognition, https://arxiv.org/pdf/1512.03385.pdf

[13] Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation https://ieeexplore.ieee.org/document/363096

Appendix A

Dataset - Herlev Papsmear

Herlev Pap Smear Dataset is a publicly available dataset of cervical cell images that are created by the Herlev University Hospital, Denmark. It consist of 917 images of single cells that are classified into seven classes : carcinoma in situ, light dysplasia, moderate dysplasia, normal columnar, normal intermediate, normal superficial and severe dysplasia. There are several versions of the dataset file, we chose Smear2005

Link to the dataset: `https://opendatalab.com/HErlev/download`

Appendix B

Source code

B.0.1 Preprocessing Masks

```
from google.colab.patches import cv2_imshow
pp1 = '/content/masks_split/train'
pp2 = '/content/masks_split/val'
pp = [pp1,pp2]
for p1 in pp:

  path1 = p1 + '/' + i
  for img in os.listdir(p1):
    i= p1+'/'+ img
    im=Image.open(i).convert('RGB')

    mask = cv2.imread(i)
    red = [0, 0, 255]
    gray = [128, 128, 128]
    blue = [255, 0, 0]
```

```
    cyt = [128,0,0]
    # Create the masks for each color
    nucleus_mask = cv2.inRange(mask, np.array(blue), np.array(blue))
    cytoplasm_mask = cv2.inRange(mask, np.array(cyt), np.array(cyt))
    background_mask = cv2.bitwise_or(cv2.inRange(mask, np.array(gray), np.array(
    # Merge the masks into a single array
    mask_array = np.dstack((background_mask, cytoplasm_mask, nucleus_mask))
    cv2.imwrite(i, mask_array)
```

B.0.2 Model Training Parameters

```
    loss = smp.utils.losses.DiceLoss()
metrics = [
    smp.utils.metrics.IoU(threshold = 0.5),
    smp.utils.metrics.Accuracy(threshold = 0.5),
]

optimizer = torch.optim.Adam([
    dict(params=model.parameters(), lr=0.000001),
])
```

B.0.3 Plots

```
plt.title("IoU score vs. Epochs")

plt.xlabel("Epochs")
plt.ylabel("IoU score")

plt.plot(range(1,n_epochs+1),tloss,label="Train")
```

```
plt.plot(range(1,n_epochs+1),vloss,label="Validation")

plt.legend(["training_set IoU Score","Validation_set IoU Score"])

plt.show()

plt.title("Loss vs. Epochs")

plt.xlabel("Epochs")
plt.ylabel("Loss")

plt.plot(range(1,n_epochs+1),tdloss,label="Train")
plt.plot(range(1,n_epochs+1),vdloss,label="Validation")

plt.legend(["Training_set Loss","Validation_set Loss"])

plt.show()
```

B.0.4 Evaluation metrics

ZSI

```
    def calculate_zsi(gt, pr):
    # convert to float tensor
    gt = gt.float()
    pr = pr.float()

    # calculate intersection and union
```

```
    intersection = torch.sum(gt * pr)
    union = torch.sum(gt) + torch.sum(pr) - intersection

    # calculate ZSI
    zsi = intersection / union
    return zsi

test_set= dt_test
zsi_scores = []

for i in range(50):
    image_vis = test_set[i][0]
    image, gt_mask = test_set[i]
    gt_mask = gt_mask.squeeze()
    x_tensor = (image).to(DEVICE).unsqueeze(0)
    pr_mask = best_model.predict(x_tensor)
    pr_mask = (pr_mask.squeeze().cpu().numpy().round())
    pr_mask = torch.from_numpy(pr_mask)
    zsi_score = calculate_zsi(gt_mask, pr_mask)
    zsi_scores.append(zsi_score)

print(f"Average ZSI score over the test set: {sum(zsi_scores)/len(zsi_scores)}")
```

Precision, Recall and F1-score

```
pmm = []
gmm = []
```

```
for i in range(len(os.listdir(path4))):
  image_vis = dt_test[i][0]
  image_, gt_mask = dt_test[i]
  image  = image_.numpy()
  x_tensor_ = torch.from_numpy(image).to(DEVICE).unsqueeze(0)
  pr_mk = best_model.predict(x_tensor_)



  pmm.append(gt_mask)
  gmm.append(pr_mask_)
gt_mask = torch.stack(gmm).squeeze(1)
pr_mask = torch.stack(pmm).squeeze(1)

gt_mask = gt_mask.to(DEVICE)
pr_mask = pr_mask.to(DEVICE)

print(pr_mask.size())
gt_mask = gt_mask.long()
pr_mask = pr_mask.long()

tp,fp,fn,tn = smp.metrics.get_stats(pr_mask, gt_mask, mode='multiclass',num_clas
print(tp,fp,fn,tn)

f1_score = smp.metrics.f1_score(tp, fp, fn, tn,reduction = 'macro-imagewise')

recall = smp.metrics.recall(tp, fp, fn, tn, reduction="macro-imagewise")
```

```
precision  = smp.metrics.functional.precision(tp, fp, fn, tn, reduction='macro')


print(precision)

print()

print(f1_score)

print()

print(recall)
```