International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

DeepSegNet: An Innovative Framework for Accurate Blood Cell Image Segmentation

D. Vetrithangam[1], (Corresponding author), Dr. Subba Reddy[2], Naresh Kumar Pegada[3], Dr. B. Arunadevi[4], Dr. M. Pompapathi[5], Amar Choudhary[6], Ashok Bekkanti[7]

[1]Professor, Department of Computer Science & Engineering,
Chandigarh University, Mohali, Punjab-140413, India.
e-mail: vetrigold@gmail.com

[2]Professor and HoD, Department of Computer Science & Engineering,
Sai Rajeswari Institute of Technology, Proddatur, YSR Dist, Andhra Pradesh-516362, India.
e-mail: y.subbareddy@gmail.com

[3]Assistant Professor, Department of Computer Science & Engineering (AI&ML),
Keshav Memorial Engineering College, Peerzadiguda, Uppal, Hyderabad, Telangana, India
e-mail: pnrshkumar@gmail.com

[4]Professor, Department of Electronics & Communication Engineering,
Dr. N.G.P Institute of Technology, Coimbatore, India.
e-mail: arunadevi@drngpit.ac.in

[5]Associate Professor, Department of Information Technology,
RVR & JC College of Engineering, Guntur, Andhra Pradesh, India
e-mail: manasani.pompapathi@gmail.com

[6]Assistant Professor, Department of Electronics & Communication Engineering,
Alliance College of Engineering and Design, Alliance University, Bengaluru, Karnataka-562106, India.
e-mail: amar.giet.ece@gmail.com

[7]Assistant Professor, Department of Computer Science & Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.
e-mail: ashok.bekkanti@gmail.com

Abstract - Image segmentation plays a crucial and indispensable role in computer vision, as it allows the partitioning of an image into meaningful regions or objects. Among its numerous applications, image segmentation holds particular significance in the domains of medical diagnosis and healthcare. Its vital role in this field stems from its ability to extract and delineate specific anatomical structures, tumors, lesions, and other critical regions from medical images. In medical diagnosis, accurate and precise segmentation of organs and abnormalities is paramount for effective treatment planning, disease monitoring, and surgical interventions. Blood cell image segmentation is highly valuable for medical diagnosis and research, particularly in the domains of hematology and pathology. Precisely segmenting blood cells from microscopic images is essential, as it offers critical insights into various blood-related disorders and diseases. Although deep learning segmentation models have exhibited promising results in blood cell image segmentation, they suffer from several limitations. These drawbacks encompass scarce data availability, inefficient feature extraction, extended computation time, limited generalization to unseen data, challenges with variations, and artifacts. Consequently, these limitations can adversely impact the overall performance of the models. Blood cell image segmentation encounters persistent challenges due to factors like irregular cell shapes, which pose difficulties in boundary delineation, imperfect cell separation in smears, and low cell contrast, leading to visibility issues during segmentation. This research article introduces the innovative DeepSegNet framework, a powerful solution for precise blood cell image segmentation. The performance of widely-used segmentation models like PSPNet, FPN, and DeepLabv3+ is enhanced through the use of sophisticated preprocessing techniques, improving generalization capability, data diversity, and training stability. Additionally, the incorporation of diverse dilated convolutions and feature fusion further contributes to the improvement of these models. The Improved PSPNet, Improved FPN, Deep Lab V3, and Improved Deep Lab V3+ achieved 98.25%, 99.04%, 98.23%, and 99.31% accuracy, respectively, and the Improved Deep Lab V3+ model outperformed well and produced a Dice Coefficient of 99.32% and Precision of 99.38%. The proposed DeepSegNet framework improves overall performance with an increased accuracy of 8.91%, 3.72%, 17.73%, 22.83%, 7.96%, 9.61%, 17.36%, 6.22%, 13.32%, and 14.32% compared to the existing models. This framework, which can be applied to accurately identify and quantify different cell types from blood cell images, is instrumental in diagnosing a variety of hematological disorders and diseases.

Index terms: Blood Cell, FPN, Segmentation, PSPNet, Deep Lab V3+, Deep Learning, Image

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

1. INTRODUCTION

Over the past two decades, numerous research teams have dedicated their efforts to creating computerized systems capable of analyzing various types of medical images and extracting valuable information to support medical professionals[1][2][3]. Blood cells are a vital part of the circulatory system and are responsible for carrying oxygen, fighting infections, and maintaining overall health. Blood cells can be classified into three main types. Firstly, red blood cells have the crucial responsibility of transporting oxygen from the lungs to the body's various tissues and organs. Secondly, white blood cells, further referred as leukocytes, are integral components of the immune system, acting as defenders against infections and foreign invaders like bacteria, viruses, and pathogens. Finally, the body uses thrombocytes, tiny cell fragments, to help blood clot, which helps it correct wounds and control excessive bleeding. The nucleus' and cytoplasm's shape, color, size, and texture vary amongst these groupings. Red blood cells make up a far larger portion of a blood smear than white blood cells do[4][5].Leukocyte cells containing granules are referred to as granulocytes, which are composed of neutrophils, basophils, and eosinophils. Agranulocytes are cells that lack granules. The role of cells in protecting organisms from infections is crucial, and specialists can utilize their precise concentrations to identify the presence or absence of significant pathological conditions[6][7].Image segmentation has numerous practical uses in medical imaging, such as the analysis of anatomical features, aiding in diagnosis and therapy planning, and identifying tumors and other diseases. Image analysis holds significant importance in accomplishing various essential objectives, like gathering information, conducting screening and investigation, enabling diagnosis, offering therapy and control, and facilitating monitoring and evaluation. Automated systems and computerized tools have also been developed to assist in blood cell analysis, aiding in faster and more accurate diagnoses[8].Cell classification holds significant importance, particularly in laboratories. For instance, counting a patient's blood cells is utilized to collect data regarding cells that are not commonly exist in peripheral blood but could be released during specific disease processes, aiding hematologists in their diagnosis and analysis. The computer-aided system was proposed with the aim of automating the process of detecting and identifying different blood cell types from blood smear images [9][10][11]. Segmentation involves the segregation of a digital image into various parts, aiming to simplify and transform the image representation into a more meaningful and easily analyzable format[12].Segmentation can be divided into supervised or unsupervised learning and classification. Accurately validating a segmentation output necessitates access to the "ground truth" as a primary requirement[15].The "ground truth" encompasses the true size, shape, or other spatial features of the object of interest. Extraction of blood cells from a complex backdrop and segmentation of each cell into various components of morphology, like the nucleus, organelles, cytoplasm, holes, and others, are the two main objectives of blood cell segmentation[13][14]. The White Blood Cells in Microscopic Bone Marrow images are segmented and classified using the Fuzzy C-means (FCM) algorithm[16].For partitioning the White Blood Cells (WBC) into their two dominant elements, the nucleus and cytoplasm, nucleus segmentation and cytoplasm segmentation are employed, utilizing pixel-intensity thresholding techniques; this helps in finding diseases like acute leukemia[17]. Marker-controlled watershed algorithms could avoid over segmentation issues and utilized morphological operation functions to eliminate unwanted objects [18]. To segment the nucleus, the Self-Dual Multiscale Morphological Toggle (SMMT) operator provides good accuracy[19]. The Lexicographical Ordering Scheme (LOS) is applied to partition images that emphasize the essential color of a specific part of area [20]. The Sobel edge detector and the Watershed transform separated overlapping cells during image segmentation[21].20 general segmentation loss functions on four typical 3D segmentation tasks were used, and models produced the Dice similarity coefficients of 0.9547, 0.9566, 0.9345, and 0.9463 for the Dice (Batch, no square), Batch Square Dice, sample Dice (no square), and sample square dice, respectively[22].The straightforward separation of the cytoplasm and nucleus areas is made possible by the combination of the linear contrast approach and color segmentation that uses HSI (Hue, Saturation, and Intensity) [23].Using samples of the WBC's nuclei and sub-images, a probability density function was used to create a probability map; mean-shift clustering was utilized for region segmentation. Subsequently, a morphological opening technique was utilized to a green image to enhance the granules' degree of intensity [24]. The researcher has developed multiple algorithms and techniques for blood cell image segmentation, but certain gaps remain in these methods.

1. Accuracy: Existing techniques lack high accuracy in blood cell segmentation. Achieving precise and reliable segmentation results is essential for various medical applications, including disease diagnosis and cell counting.

2. Small Datasets: Some techniques rely on small datasets for training blood cell segmentation models. Insufficient data can limit the model's capacity for generalization to diverse cell types and variations present in blood cell images.

3. Noisy Input Images: Blood cell images can suffer from noise, artifacts, and staining inconsistencies, which can negatively impact the accuracy of segmentation. Robust

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

segmentation methods are needed to handle noisy input images effectively.

4. Lack of Preprocessing: Insufficient preprocessing of blood cell images may impact the quality of the segmentation outcomes. Proper preprocessing techniques, such as noise reduction, contrast enhancement, and color normalization, are crucial for improving segmentation accuracy.

5. Long Training Time: Some segmentation techniques require a long time to train, which can be impractical for real-time or time-sensitive medical applications. More efficient segmentation methods that reduce training time are desirable.

6. Segmentation System Failures: Some models may occasionally fail to correctly segment blood cells due to the complexity of cell shapes and appearances. Robust segmentation algorithms are necessary to minimize false positives and false negatives.

7. Variations and Artifacts: Blood cell images can exhibit variations in cell size, shape, and staining patterns. Additionally, imaging artifacts may be present. Segmentation models should be able to handle such variations and artifacts effectively.

8. Interpretability and Explainability: In the context of blood cell image segmentation, the importance of interpretability and explainability lies in instilling trust and confidence in the model's outputs. Understanding the regions contributing to the segmentation can aid clinicians and researchers in accurate cell analysis.

The objectives of the paper are:

1. To conduct a background study and literature review for Blood cell image segmentation and other image segmentation models.
2. To propose a novel framework titled "DeepSegNet" for accurate Blood cell image segmentation with a good dice coefficient and less segmentation time.
3. To test and validate the proposed framework on various experimentation parameters like accuracy, recall, precision, and Dice coefficient.
4. To compare the proposed framework with existing techniques like Unet, WBC-Net, self-supervised learning techniques, Circle Hough Transform, and so on.

Organization of Paper

The rest of the paper is organized as follows: Section 2 provides a thorough analysis of image segmentation methods. Section 3 covers the materials and methods used in the proposed DeepSegNet framework. Section 4 explains the system model, architecture, and working principles of the proposed DeepSegNet framework. Section 5 explains the results produced by the framework and provides a detailed comparative analysis. Section 6 concludes the paper with future scope.

2. LITERATURE REVIEW

Deep segmentation models have been widely used for blood cell image segmentation, leveraging their ability to accurately segment cells and improve the efficiency of analysis and diagnosis in various medical applications. Banik et al.[25] proposed a new WBC nucleus segmentation technique that depends on color space conversion and the k-means algorithm. The WBC is accurately localized and separated from the entire blood smear image, utilizing the location information from the segmented nucleus. This method achieved results of 97.57% accuracy, precision of 87.63%, recall of 96.08%, and sensitivity of 97.92%. This method used a dropout layer to prevent the model from overfitting, but this model took computation time. Lu et al. [26] proposed WBC-Net, which depends on UNet++ and ResNet. By employing a context-aware feature encoder with residual blocks, the WBC-Net extracts multi-scale features. Moreover, mixed skip pathways are introduced on dense convolutional blocks to gather and combine image features at diverse scales. The proposed technique achieved results of 98.48% precision, 98.21% dice coefficient, and 0.57% false Negative rate with the use of a deep supervision structure in WBC-Net. Without a shape-aware loss, the model may struggle to capture and understand these intricate shape details, resulting in inaccurate segmentation. Li et al.[27] proposed a new technique by integrating the neural ordinary differential equations (NODEs) and U-Net networks to segment the blood smear image, and the network depth was increased by adjusting the acceptable error margin of the ODE block. This technique achieved 95.3% pixel accuracy and 90.61% mean intersection over union, but it is lacking in multiscale feature fusion. Zheng et al.[28]suggested a self-supervised learning technique with the advantages of unsupervised initial separation, supervised segmentation enhancement, and an effective cluster sampling strategy to reduce the time. The intrinsic characteristics of every test pixel are represented using color features. This approach produced the lowest over segmentation-rate (OR) of 0.69% and overall error rate (ER) of 5.24%. Mandyartha et al.[29]used the global and Adaptive Thresholding methods to segment the white blood cell images. The advantage of the adaptive thresholding method is that each partial area of the image's threshold value is calculated. This method produced average precision and recall of 91.79% and 94.03%, while global thresholding achieved 23.38% and 99.39%, respectively, but only 35 blood smear test images were used

during testing. MohdSafuan et al. [30]used segmentation methods to perform white Blood Cell (WBC) count analysis in Blood Smear images. The Circle Hough Transform (CHT) played a vital role in identifying and counting the WBC; this method produced segmentation accuracy of 96.92%. The advantage of this paper is that the color space correction process improved the WBC segmentation accuracy. Zhang et al. [31] proposed a new framework called K-Net, which segments instances and semantic types using a group of learnable kernels. Every kernel was made dynamic for its respective group in the input image by using the kernel update approach. This framework produced a semantic segmentation of 54.3% mIoU, and its instance segmentation performance was 60% to 90%. Chen et al.[32]proposed a DRINet model that combines an unpooling phase, a deconvolutional phase with residual inception phases, and a convolutional phase. This model captures both local and global contextual data efficiently through the extraction of features at multiple scales by using filters of distinct sizes within a single layer, but the drawback of this technique is that it makes training more difficult and testing slower and it produced a dice coefficient of 83.47%.Gao et al. [33] proposed a new image segmentation technique with various benefits, such as maximum variance and histogram valley threshold. This method achieved 54.1091% of the SNR value, but its feature extraction is not highly efficient. Liet al.[34]used a semi supervised method to broaden the transformation in a more universal manner, and the advantage of this method is that it includes scaling and optimizing the consistency loss. This method achieved a Jaccard index (JA) of 78.1%, a dice coefficient (DI) of 86%, pixel wise accuracy (AC) of 94.1%, sensitivity (SE) of 86.2%, and specificity (SP) of 96.8%.Amit et al.[35]proposed a new segmentation method using the additional elements of diffusion-probabilistic models. The segmentation map is refined by adding the encoding layers and decoders. This method produced a dice coefficient of 81.59%, but it took 200 diffusion steps. Jha et al.[36] proposed DoubleU-Net to capture more semantic information through lesion boundary segmentation and learned features from ImageNet. The advantage of VGG-19 is that it concatenates U-Net and produces a dice coefficient (DSC) of 0.7649. Liu et al. [37] proposed a new toolkit segmentation method for designing segmentation models and optimizing their performance. It achieved a result of 80.67% mIoU. Valanarasuet al.[38] proposed a new UNeXt method to perform the partition of medical images. The advantage of this technique is that the input channels are shifted when entered into the MLP network to improve local dependencies. This method produced the results of an F1 score of 90.41% and an IoU score of 82.78%. Gao et al. [39] proposed a new Hybrid Transformer Architecture to perform medical image segmentation. The transformer is initialized into convolutional networks; this method was tested for 150 epochs, took much computation time, and scored a 93.1% dice coefficient. Zhou et al. [40] proposed a new UNet++ architecture to perform semantic and instance segmentation. The skip connections are employed to integrate the features of various semantic scales in the networks of the decoder, and a pruning scheme is devised to increase the inference speed of UNet++, which achieved 91.36% of the dice coefficient. Chen et al. [41] proposed a new TransUNet architecture to perform medical image segmentation. Transformers and U-Net are combined to make the encoder, which increases the accuracy of segmentation and achieves 89.71% of DSC. Baumgartne et al. [42]proposed a novel segmentation technique to partition the images; this model adds the conditional probability distribution to the segmentation method, and this achieved a normalized cross correlation of 0.8453.Huang et al.[43]proposed a novel MISS Former network to perform medical Image segmentation, and this transformer assists in assisting in capturing both long-range dependencies and local context within multi-scale features. It achieved a dice coefficient of 81.96%. MissFormer could be prone to overfitting if it possesses a substantial number of parameters, and is trained on a short amount of data. To address this problem, regularization strategies and data augmentation may be required.

Li et al. [44] proposed a new Image Projection network; the new concept of excluding retinal layer segmentation and projection maps has been introduced. This network produced a Foveal avascular zone segmentation of 88.61% and a retinal vessel (RV) segmentation of 88.15%, but this network faced quantification problems in OCTA images. Zhuet al. [45]used PSPNet to segment the coronary angiography image using the spatial PP module and feature maps into a certain number of regions, and this method achieved an accuracy of 95.7%.Guoet al.[46]used CNN to design multimodal image segmentation of medical images; it added the advantages of fusing the features at convolutional layers, and this technique produced a dice coefficient of 85%, but this has the drawback that depending on the number of modalities and the severity of the misalignment, good voxel-level correlation with incorrect registration across various modalities in an incoming patient can result in drastically lower prediction ability within the misaligned region. Xie et al. [47] proposed a new framework to perform medical image segmentation that takes advantage of integrating a CNN and a transformer and adds the advantages of the deformable self-attention mechanism. This framework achieved an average score of 85%, but it took much time and has taken around 1000 epochs. Zhang et al. [48]proposed a new architecture called TransFuse, in which Transformers and CNNs are combined in parallel. This architecture increases global dependency and low-level spatial detail, and it produced a dice coefficient of 87.2% and an accuracy of 94.4%, but memory and computational time will

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

be increased during training. Jun et al. [49] proposed a new approach called Segment Anything Model (SAM) that explains the usage of SAM to augment image input for general medical image segmentation, and this was used for uncertainty estimations. Li et al.[50] proposed a new model called Eres-UNet++ to segment the liver CT images; The depth of the feature map is coupled with geographical data using the attention module to improve the segmentation method's efficiency, and it produced a dice coefficient of 95.6% and an accuracy of 89.3%, but the model increased the segmentation time.

The existing models and techniques in blood cell image segmentation have limitations concerning high accuracy and certain gaps. Some methods relied on small datasets, leading to limited accuracy and variation in the Dice coefficient, while others faced challenges with noisy input images or lacked preprocessing, resulting in time-consuming training processes. To tackle these issues, deep learning models need to exhibit robustness in handling variations and artifacts. Augmenting training data with realistic variations can improve their generalization capabilities. Additionally, to instill trust and confidence in the model's image segmentation and predictions, interpretability and explainability play vital roles in the context of blood cell image segmentation. Furthermore, some models lack the ability to derive both local and global data and suffer from dependency problems. Additionally, many techniques encounter difficulties due to high computation times. Understanding the key features or regions contributing to the image segmentation can prove beneficial for clinicians and researchers in analyzing and validating the results. Consequently, a "DeepSegNet" framework has been developed to address the identified gaps in existing image segmentation methodologies, with a particular emphasis on blood cell image segmentation. The primary aim of this framework is to increase accuracy, Dice coefficient, and overall performance by providing an improved approach to blood cell image segmentation.

3. MATERIALS AND METHODS

3.1 Materials

3.1.1 Dataset

The Blood Cells Image Dataset consists of 17,092 images of individual normal cells. These cells were acquired using the CellaVision DM96 analyzer in the Core Laboratory at the Hospital Clinic of Barcelona. The CellaVision DM96 analyzer is a commonly used instrument in clinical laboratories for the automated analysis of blood cells. It uses digital imaging technology to capture high-resolution images of individual cells, allowing for detailed examination and analysis. By using this analyzer, the dataset provides a comprehensive collection of normal cell images, enabling researchers and practitioners to study and understand the characteristics, morphology, and variability of different cell types. This dataset (https://prod-dcd-datasets-cache-zipfiles.s3.eu-west1.amazonaws.com/snkd93bnjr-1.zip) can serve as a valuable resource for various applications, such as training and evaluating machine learning models for cell classification, developing image analysis algorithms, or supporting research in the field of hematology. The utilization of the CellaVision DM96 analyzer in capturing these images ensures a standardized and reliable data collection process, which is essential for maintaining consistency and accuracy in the dataset. Figure 1 illustrates few of the Blood cell images available in the dataset:

i)Basophil: Basophils are subsets of white blood cells (leukocytes) that play a role in the immune response. They contain granules that release histamine and other substances during allergic reactions and inflammation.

ii) Eosinophil: Eosinophils are a distinct type of white blood cell that participates in the immune system's functions. They are responsible for combating parasitic infections and are also involved in allergic responses.

iii) Erythroblast: Erythroblasts are immature red blood cells located in the bone marrow. As they mature, they transform into red blood cells, responsible for transporting oxygen throughout the body.

iv) Ig (Immunoglobulin): Plasma cells create immunoglobulins, also called antibodies, in response to foreign substances (antigens) in the body. By focusing on and neutralizing particular antigens, they perform a critical role in the immune system.

v) Lymphocyte: Lymphocytes are a crucial type of white blood cell and an essential component of the immune system. They comprise B cells, T cells, and natural killer (NK) cells, each having distinct roles in recognizing and eradicating foreign invaders.

vi) Monocyte: Monocytes are a kind of white blood cell that circulate in the bloodstream and are precursors to macrophages and dendritic cells. They are crucial for immune defense and play a role in the clearance of dead cells and pathogens.

vii) Neutrophil: Neutrophils, the most plentiful category of white blood cells, are essential components of the innate immune system. Acting as the first responders to infections, they play a crucial role in phagocytosis by engulfing and eliminating bacteria and other pathogens.

viii) Platelet: The blood contains platelets, which are tiny, disc-shaped cells essential for clotting. In instances of injury or damage to blood vessels, platelets gather at the location to create a clot, aiding in the prevention of bleeding.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

3.1.2 Data Pre-PROCESSSING

Several preprocessing techniques, such as Image resizing, image normalization, Data Augmentation, Image cropping, and Data splitting, are performed on the Blood cell images. Image resizing ensures that all images in the dataset have the same dimensions, which is essential for batch processing during training and inference. The images in the dataset are of various sizes: 360 x 363, 360 x 360, and 360 x 361 are resized into a common size of 256 x 256. The image normalization process is a crucial step to ascertain that the pictures' pixel values are in a standardized range before feeding them into the model. The typical approach for image normalization in DeepLabv3 models involves rescaling the pixel values to achieve a mean of 0 and a standard deviation of 1. This process is applied channel-wise (e.g., for each color channel) to maintain the color information in the image. By performing various alterations on the original pictures, a technique called data augmentation is used to fictitiously increase the size of the training dataset. Common augmentations include random rotations, flips (horizontal and vertical), zooming, brightness adjustments, and translations. Data augmentation aids in introducing variability into the training data, which helps the model generalize better to unseen variations in the test data and reduces overfitting. Random rotations, zooming, brightness adjustments, and translations are performed on the dataset. The original image is rotated by a certain angle. Rotating the image helps the model learn to be invariant to object orientations, as different angles of the same object are treated as equivalent. The cells may appear at different
angles in the Blood cell images, so augmenting the data with rotations helps the model recognize them regardless of their orientation. Brightness adjustment involves modifying the pixel values of the image to make it brighter or darker. This augmentation technique helps the model become more robust to changes in illumination conditions. Blood cell images might have varying lighting conditions in different samples, and augmenting the data with brightness adjustments enables the model to handle these variations. By applying random translations, you create slightly different versions of the same image, which provides the model with more training examples and makes it more invariant to object positions within the image. For blood cell images, translations can be useful to account for the fact that cells may be located at different positions within the images. This ensures that the model can identify cells regardless of their location. Image cropping involves selecting a part of interest from an image and discarding the rest. Cropping can be used to focus on specific regions or objects of interest in the image. Data splitting is the process of dividing the dataset into multiple subsets for training, validation, and testing purposes. The typical split ratio is 80-20 for training and testing in the proposed work. Proper data splitting is crucial for an unbiased evaluation of the model's generalization ability.

3.2 Methods

3.2.1Pyramid Scene Parsing Network

PSPNet (Pyramid Scene Parsing Network) is a deep learning

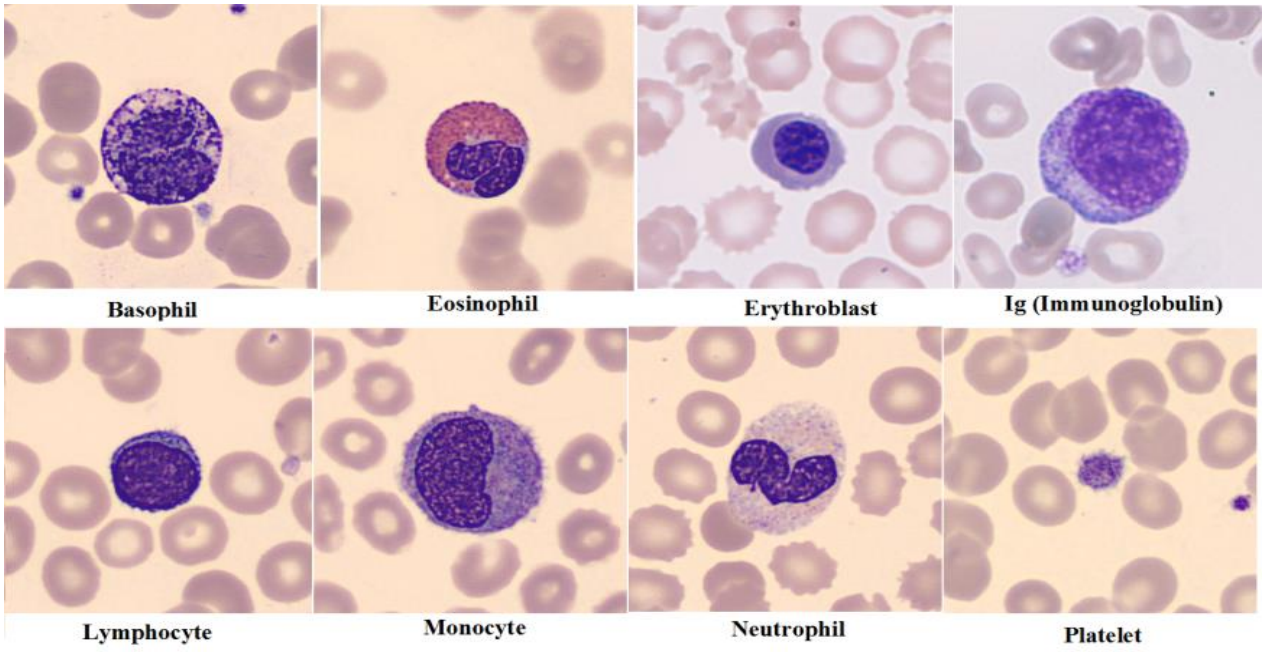

Figure 1: Sample Blood Cell images from the PBC_dataset_normal_DIB

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

architecture built for high-resolution image segmentation tasks specifically semantic segmentation. PSPNet captures rich contextual information at multiple scales in an image to improve
the accuracy of pixel-wise semantic segmentation. It achieves this by utilizing a PP module that efficiently gathers context data from distinct areas of the input image.As shown in figure 2, the input is a set of images of size (W, H). The key components of PSPNet are: 1) Encoder: It is typically a pre-trained convolutional neural network (CNN) that derives high-level information from the input image. The PSPNet used ResNet-101 or ResNet-50 as the backbone, but other CNN architectures can also be used. 2) Pyramid Pooling Module: PP Module is the core of PSPNet and is responsible for capturing contextual information at different scales. It consists of four parallel average pooling layers with different kernel sizes. These pooling layers capture information over varying receptive fields and help the network understand the scene from different perspectives. The outputs from these pooling layers are then up sampled to the original spatial resolution and combined together. 3) Decoder: The decoder takes the added feature maps from the PP Module and processes them to obtain the final segmentation mask. Up-sampling layers are frequently used to boost the spatial precision of feature maps, while skip connections from the encoder are used to preserve fine-grained features. 4) Final Convolutional Layer: The output of the decoder is sent through a 1x1 convolutional layer to produce the final segmentation mask.
The number of classes in the segmentation task corresponds to the number of output channels in this layer. Using annotated images and accompanying ground truth segmentation masks, PSPNet is trained. A suitable loss function, which determines the

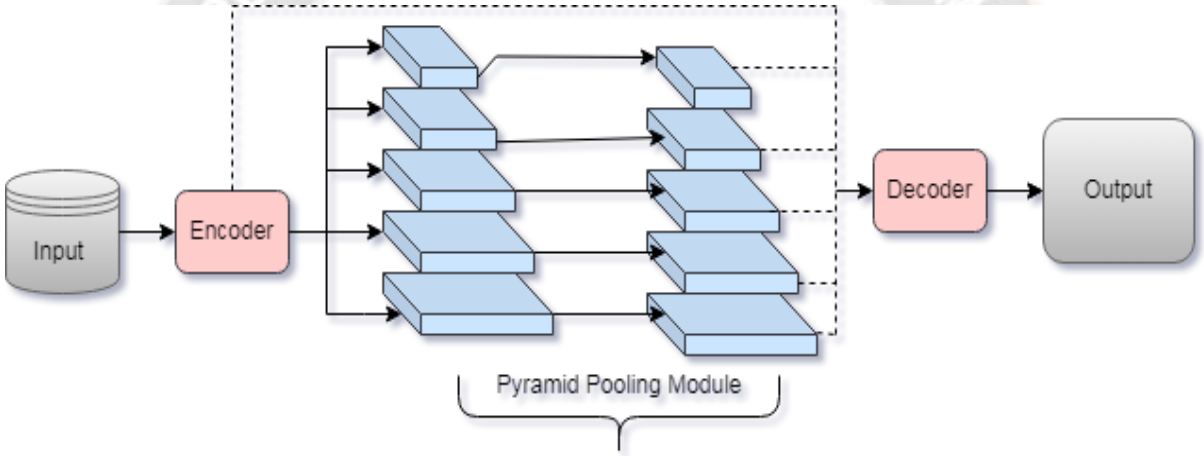

Figure 2: The architecture of the PSPNet model

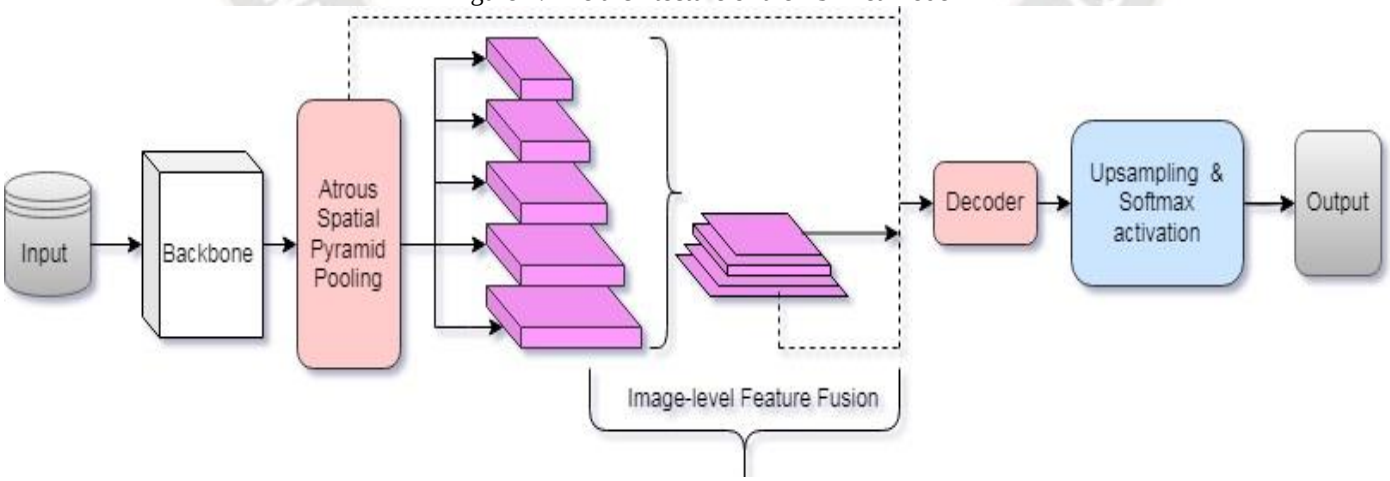

Figure 3: Architecture of the Deep Lab V3+ model.

Discrepancy between the segmentation mask's prediction and the actual data is employed for optimizing the parameters of the network. The parameters are updated using backpropagation and gradient descent during training.

3.2.2 Deep Lab V3 and Deep Lab V3+
A cutting-edge deep learning architecture for semantic picture segmentation is DeepLabv3. It builds upon the DeepLab family of models and incorporates dilated convolutions, ASPP, and a fully connected Conditional Random Field (CRF) for

accurate and detailed segmentation. The input is a set of images of size (W, H). The steps of resizing the image to a fixed size (e.g., 512x512) and normalizing the pixel values have been completed.

Backbone Network uses a pre-trained CNN as the backbone to extract high-level features. Common choices are ResNet, MobileNet, or Xception. Remove the fully connected layers of the backbone network. ASPP is performed by applying parallel dilated convolutions with different rates (e.g., 1, 6, 12, and 18) on the output feature maps from the backbone network.

This captures multi-scale contextual information at different receptive fields. The operation of up-sampling is performed by up -sampling the ASPP output feature maps using bilinear interpolation to match the original input image size.Skip Connections connect feature maps from the backbone network at multiple scales to the upsampled feature maps to recover fine-grained details. Convolutional Layers apply a 1x1 convolutional layer to the combined feature maps to reduce the number of channels. Final upsampling is performed to upsample the output feature maps to the original input image size using bilinear interpolation. Applying the Softmax activation yields the per-pixel probability of each class. The final output is the segmentation mask, which indicates the class label for each pixel. During training, labeled images with corresponding ground truth segmentation masks are required.

The model is trained using a suitable loss function, such as cross-entropy loss or Dice loss, to compare the predicted segmentation mask with the ground truth. Optimization is performed using backpropagation and gradient descent techniques. Figure 3 shows the Architecture of the Deep Lab V3+ model in which the operation of Image-level Feature Fusion performs global average pooling (GAP) on the backbone output feature maps to obtain a global representation of the image. A 1x1 convolutional layer is used to reduce the dimensionality of the global representation. The Upsampling is performed on the reduced representation to match the size of the ASPP output feature maps. The upsampled representation is combined with the ASPP output feature maps through element-wise summation or concatenation to incorporate image-level information. The above-mentioned set of operations is performed for both Deep Lab v3 and Deep Lab v3+. A few additional operations are performed to increase the segmentation accuracy of the model. In Deep Lab V3+, the operation of Image-level Feature Fusion performs GAP operation on the backbone output feature maps to obtain a global representation of the image. A 1x1 convolutional layer is used to reduce the dimensionality of the global representation. The Upsampling is performed on the reduced representation to match the size of the ASPP output feature maps. The upsampled representation is combined with the ASPP output feature maps through element-wise summation or concatenation to incorporate image-level information. The decoder (Segmentation Head) upsamples the fused feature maps using bilinear interpolation to increase the spatial resolution, and to reduce the number of channels, a 1x1 convolutional layer is employed. Optionally, skip connections from the backbone network are used to recover fine-grained details.

3.2.3 Feature Pyramid Network

The FPN (Feature Pyramid Network) is a popular architecture that addresses the challenge of multi-scale object detection and feature representation. It combines high-level and low-level features to achieve accurate and efficient object detection.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

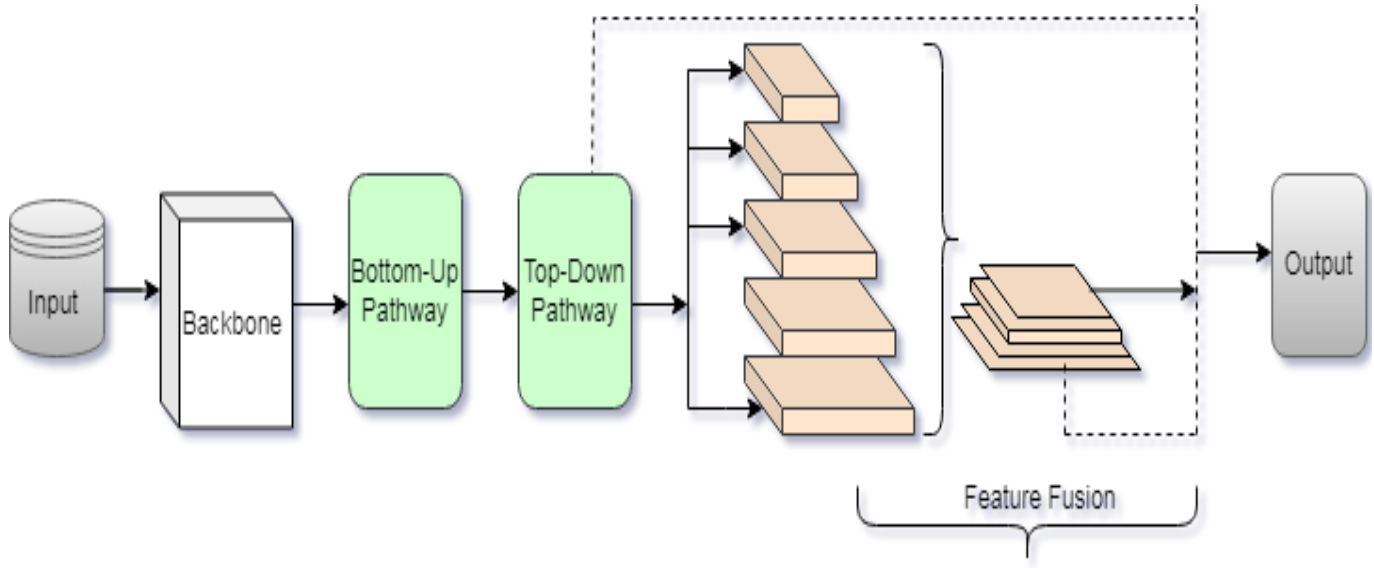

Figure 4 Architecture of the FPN network

As shown in figure 4, the FPN network is composed of the following modules: 1) Backbone networks that use a pre-trained CNN such as ResNet, VGG, or EfficientNet as the backbone. This backbone network extracts features from the input image at multiple spatial resolutions. 2) The process of the bottom-Up Pathway is performed in the backbone network, which typically consists of multiple stages, each producing feature maps at different spatial resolutions. It starts with the lowest-resolution feature map and applies convolutional layers to diminish the number of channels while retaining spatial information. This process is performed for each stage of the backbone network, resulting in a series of feature maps with increasing spatial resolutions. 3) The process of Top-Down Pathway is performed in a feature pyramid that starts from the highest-resolution feature map. To decrease the number of channels, a 1x1 convolutional layer is utilized. Then, upsampling the feature map using the nearest neighbor is done to match the spatial resolution of the corresponding feature map from the lower level.

The upsampled feature map is combined with the lower-level feature map through element-wise addition or concatenation. This process is repeated for each level of the feature pyramid, creating a top-down pathway that recovers spatial details while incorporating high-level semantic information. 4) Feature Fusion is performed to facilitate information flow between different levels of the feature pyramid through lateral connections. A 1x1 convolutional layer is employed to the higher-level feature maps to adjust their channel dimensions to match the lower-level feature maps. Combining the adjusted higher-level feature maps with the corresponding lower-level feature maps through element-wise addition is done. By facilitating the fusion of multi-scale features, this process enhances the representation of objects at different scales. The FPN architecture is applicable to diverse tasks, encompassing object detection and semantic segmentation. For object detection, additional prediction heads are attached to every step of the feature pyramid to generate bounding box coordinates, class probabilities, and other relevant attributes. For semantic segmentation, the fused feature maps can be further processed with convolutional layers to produce pixel-level segmentation masks.

4. PROPOSED METHODOLOGY

4.1 System model

The PSPNet is a deep learning model designed for pixel-level scene parsing and semantic segmentation tasks. This network consists of two main components which are the FPN and the Pyramid Pooling Module (PPM).The FPN is tasked with extracting multi-scale features from the input image. It utilizes ResNet to extract hierarchical features at different spatial resolutions. These features are obtained from intermediate layers of the backbone network and are used to construct a feature pyramid. The feature pyramid is built by connecting the feature maps from the top-down and bottom-up pathways, enabling the model to gather both low-level and high-level semantic information. The PP module is built to capture global context data from the feature pyramid. It operates on the feature maps obtained from the FPN and performs spatial pyramid pooling (SPP) at multiple scales. The PPM divides the input feature maps into non-overlapping regions and applies average pooling to each region with different pooling scales. By doing so, the PPM captures contextual information at multiple scales, enabling the model to have a holistic understanding of the scene on a global scale. The pooled features from each scale are then concatenated and fed into subsequent convolutional layers for further processing.

Following the PPM, the model generally incorporates extra convolutional layers and upsampling operations to enhance features and produce predictions at the pixel level.

Finally, the output is usually sent through a softmax or sigmoid activation function to obtain the per-pixel probability map or binary mask indicating the semantic class of each pixel. During training, PSPNet utilizes pixel-wise cross-entropy loss or other suitable loss functions to compare the predicted probability map with the ground truth annotation. During the training process, the model utilizes backpropagation and gradient descent algorithms to decrease the loss value and enhance the network parameters. The mathematical operations involved in PSPNet are described here: Convolutional Operations in PSPNet employs convolutional layers for feature extraction from input images. Mathematically, a convolution operation can be represented as follows: Y [i, j] = sum (W [k, l] * X [i+k, j+l] + b) Where Y is the output feature map, X is the input feature map, W represents the learnable convolutional filters, and b is the bias term. The sum is computed over the filter size and applied at each spatial location (i, j). Pooling Operations are performed in the PPM module; where PSPNet employs pooling operations to gather multi-scale contextual data. Mathematically, pooling operations involve downsampling and aggregating information within a region. Max pooling and average pooling are generally utilized. For example, max pooling can be represented as: Y [i, j] = max(X [i*s:i*s+k, j*s:j*s+k])where Y is the pooled output, X is the input feature map, s is the stride, and k is the pooling size. Upsampling Operations are performed in PSPNet, which utilizes up sampling operations to restore the spatial resolution of feature maps. Mathematically, upsampling can be achieved through techniques like bilinear interpolation or transposed convolutions. Bilinear interpolation involves computing the value of new pixels based on the weighted average of nearby known pixels. The softmax function is commonly used in PSPNet to normalize the output probabilities over different classes. Mathematically, the softmax function is defined as follows for a pixel at location (i, j) with C classes: P[i, j, c] = exp(S[i, j, c]) / sum(exp(S[i, j, k])) for k in 1 to C , where P[i, j, c] represents the probability of class c at location (i, j), and S[i, j, c] is the score for class c at that location. During training, PSPNet employs the cross-entropy loss function to assess the degree of divergence between expected and actual segmentation maps.Mathematically, the cross-entropy loss is given by:L = -sum(GT[i, j, c] * log(P[i, j, c])) ;where GT[i, j, c] is the ground truth label for class c at location (i, j), and P[i, j, c] is the predicted probability.

DeepLab v3 is a CNN architecture utilized for semantic image segmentation. It is made to give each pixel in a picture a semantic label, providing a thorough comprehension of the scene. DeepLab v3 involves neural networks and numerical computations. DeepLab v3 typically starts with a backbone network, such as a CNN, to derive information from the input image. The input image is dealt with by the backbone network, which also creates a number of intermediate feature maps. DeepLab v3 employs atrous convolutions, further referred to as dilated convolutions, to capture multi-scale contextual information. Atrous convolutions introduce controlled holes in the convolutional filters, enabling the network to increase the receptive field without adding more settings. By using distinct dilation rates in various layers, DeepLab v3 derives information at multiple scales and preserves fine-grained details. ASPP is a key component in DeepLab v3 that further enhances the network's ability to capture multi-scale contextual information. ASPP involves parallel atrous convolutions with different dilation rates, followed by global pooling operations. The parallel convolutions at different dilation rates capture information at various scales, while global pooling aggregates information globally. The outputs of the parallel convolutions and global pooling are then concatenated to form a rich representation. DeepLab v3 incorporates a decoder module to refine the segmented output. The decoder module typically consists of upsampling operations to restore the spatial resolution and fusion with the low-level feature maps. Upsampling can be performed using techniques like bilinear interpolation or transposed convolutions. Fusion involves combining the upsampled feature maps with the corresponding low-level feature maps from the backbone network to incorporate fine-grained details. DeepLab v3 includes a final classification layer to give each pixel a meaningful designation. The classification layer is typically implemented using a 1x1 convolutional layer followed by softmax or sigmoid activation. The output is a pixel-wise classification map, where each pixel is assigned a probability distribution over different classes.

DeepLab v3 utilizes convolutional layers to derive features from input images or feature maps. Mathematically, a convolution operation can be represented as follows: Y[i, j] = sum(W[k, l] * X[i+k, j+l] + b) ; where Y is the output feature map, X is the input feature map, W represents the learnable convolutional filters, and b is the bias term. The sum is computed over the filter size and applied at each spatial location (i, j). Atrous convolutions introduce controlled holes in the convolutional filters, allowing for larger receptive fields the amount of parameters without rising. Mathematically, the atrous convolution operation can be represented similarly to a standard convolution operation, but with the added dilation parameter. ASPP in DeepLab v3 involves parallel atrous convolutions with different dilation rates and subsequent pooling operations. The parallel atrous convolutions capture information at various scales, while the pooling operations aggregate information globally. Mathematically, the atrous convolutions and pooling operations can be represented using

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

the standard convolution and pooling equations. Mathematically, upsampling can be done through techniques like bilinear interpolation or transposed convolutions. Bilinear interpolation computes the value of new pixels based on the weighted average of nearby known pixels. DeepLab v3 involves fusion and concatenation operations to combine feature maps from different layers or pathways. Mathematically, fusion can be achieved through element-wise addition or concatenation, which combines the activations of corresponding pixels or channels. DeepLab v3 typically includes a softmax or sigmoid activation function in the final classification layer. Mathematically, softmax activation normalizes the output logits into a probability distribution over classes, while sigmoid activation produces a probability for each pixel independently. These mathematical operations play a crucial role in DeepLab v3 for extracting features, capturing multi-scale information, and assigning semantic labels to each pixel in an image.

DeepLab v3+ is an enhanced version of DeepLab v3, designed to improve the spatial accuracy of semantic image segmentation.

ASPP module align with the lower-level feature maps, facilitating the fusion process. The convolution operation can be defined as follows: For each output channel j, and for each spatial position (i, p) in the output feature map, the convolution operation can be computed as: Y[i, p, j] = Summation over r, s, c_in [X[i+r, p+s, c_in] * W[r, s, c_in, j]] ; here Y[i, p, j] represents the value at spatial position (i, p) in the output feature map for channel j. X[i+r, p+s, c_in] represents the value at spatial position (i+r, p+s) in the input feature map for channel c_in. W[r, s, c_in, j] represents the value of the filter at position (r, s) for input channel c_in and output channel j. The summation is conducted over both the kernel size (k) and all input channels (C_in).The above equation computes the element-wise multiplication between the input feature map and the corresponding filter values, and then sums them up to produce the output feature map. DeepLab v3+ incorporates skip connections to combine features at different scales operations. By fusing information from multiple scales, the model can derive both local and global contextual data, enhancing segmentation accuracy. In the training process, Deep Lab v3+ employs loss functions to assess the dissimilarity between predicted segmentation maps and ground truth labels. The commonly utilized loss functions

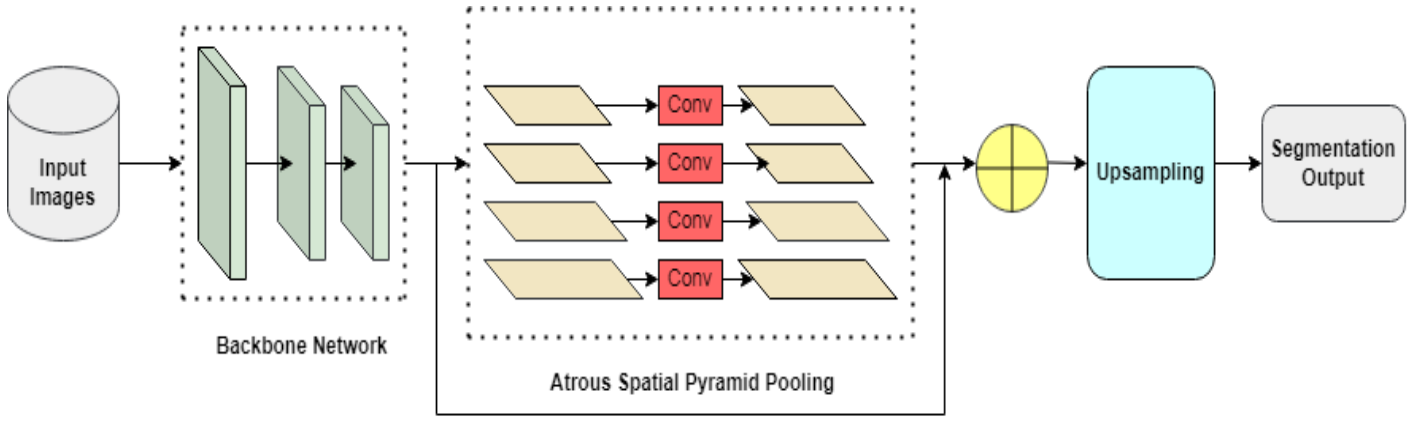

Figure 5: Architecture of the Improved PSPNet model

DeepLab v3+ incorporates improved decoder modules that includes both upsampling and skip connections from earlier layers. The skip connections help fuse low-level spatial details from the backbone network with high-level semantic information from the ASPP module, improving the segmentation accuracy. DeepLab v3+ introduces feature pyramid fusion, where the upsampled feature maps from the ASPP module are added with the corresponding low-level feature maps from the backbone network. This fusion combines the high-level semantic data with the fine-grained spatial details, allowing the network to make accurate predictions at different scales. DeepLab v3+ uses an ASPP alignment approach, which ensures that the outputs of the include cross-entropy loss, dice loss, or focal loss. This is achieved through element-wise addition or concatenation.

4.2 Architecture and working

The proposed DeepSegNet Framework includes four types of models: 1) Improved PSPNet2) Improved FPN 3)Deep Lab V3 4) Improved Deep Lab V3+.As shown in figure 5, the Improved PSPNet with Resnet-101 is designed to capture multi-scale contextual information and achieve enhanced performance in semantic segmentation tasks by effectively capturing local and global context information. PSPNetwith ResNet-101 as the backbone network incorporates various architectural enhancements to achieve better performance in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

semantic segmentation tasks. The Backbone network includes a residual module called ResNet-101, which is used as a popular CNN architecture known for its effectiveness in image classification. The ResNet-101 serves as the feature extraction backbone for the PSPNet.

cover a range of scales, capturing both local and global context. The first branch performs atrous convolution with a dilation rate of 1, which corresponds to the regular convolution. This branch captures fine-grained details and local information. The subsequent branches perform atrous

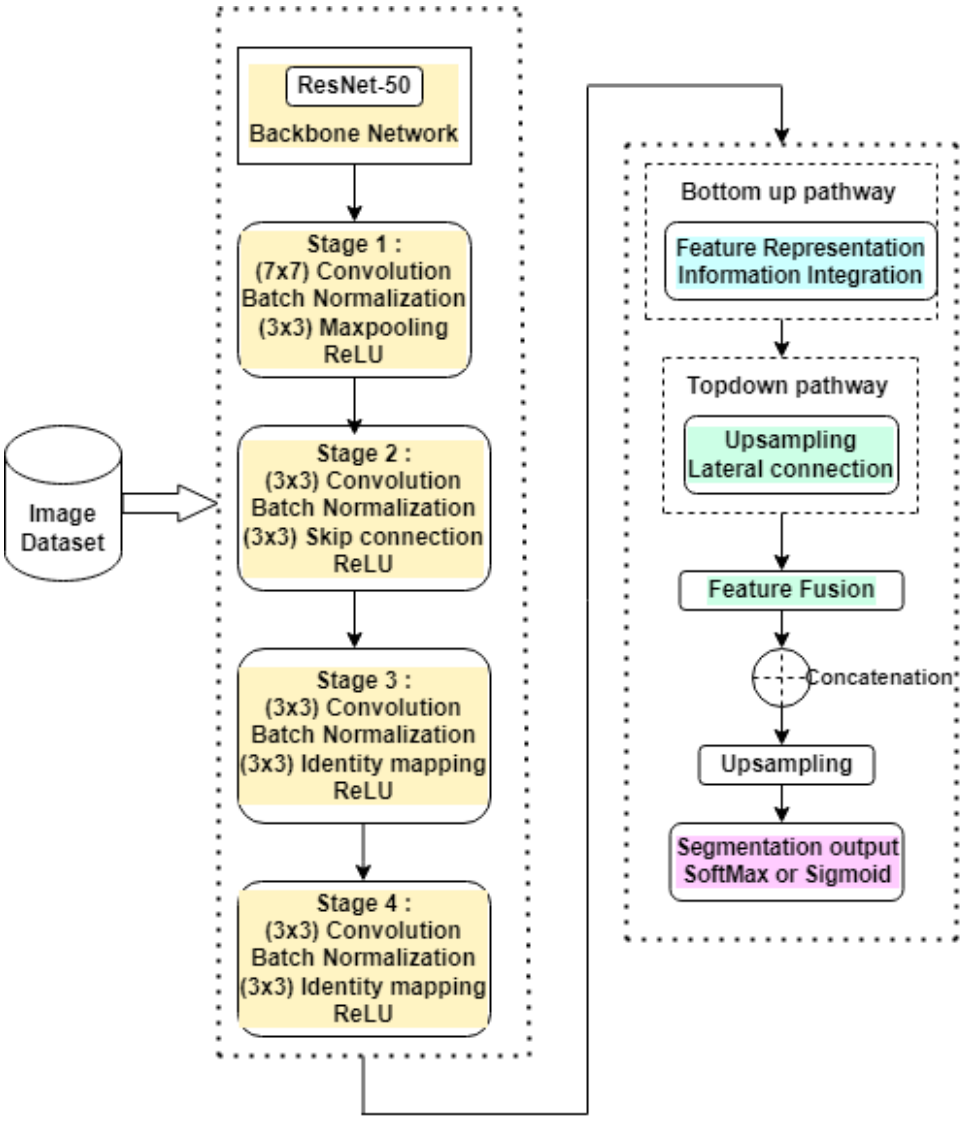

Figure 6: Architecture of the Improved FPN

It consists of multiple residual blocks, which help address the vanishing gradient problem during training. The ASPP module is used to derive multi-scale context information from the input image. It achieves this by applying parallel dilated convolutions at multiple dilation rates to the feature maps obtained from the backbone network. The dilation rates control the receptive field size of the convolutions and enable the network to capture both fine and coarse details. 3) ASPP, which is the core component of the ASPP module. It includes parallel branches, each performing atrous convolutions with various dilation rates. The dilation rates are typically chosen to

convolutions with increasing dilation rates, such as 6, 12, and 18. These branches capture information at progressively larger spatial scales, incorporating more global context. Each branch applies batch normalization and activation functions, such as ReLU, to the output of the atrous convolutions.4) In the upsampling and Concatenation operations, the feature maps obtained from the different branches of the ASPP module are upsampled or downsampled to have the same spatial dimensions. Then, these feature maps are concatenated along the channel dimension, effectively combining the multi-scale contextual information captured by each branch.5) Additional

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Convolutional Layers are utilized to additionally refine the aggregated features; additional convolutional layers may be applied to the concatenated feature maps. These layers can have smaller filter sizes to diminish the dimensionality and complexity of the features.

The ASPP module's final output is a set of feature maps that encode contextual information at multiple scales. The up sampled feature maps from the PP module and the skip connections are concatenated channel-wise to fuse the multi-scale information. This concatenation operation combines the high-level semantics from the PP module with the low-level details from the skip connections. Subsequently, convolutional layers are utilized to the concatenated feature maps to refine and integrate the information.

The refined feature maps from the decoder module undergo a final convolutional layer. This layer diminishes the number of channels and captures more abstract representations. Finally, a pixel-wise prediction layer, typically a 1x1 convolution, maps the feature maps to the desired number of output classes, producing a segmentation map that assigns each pixel to a specific class.

Figure 6 illustrates the Architecture of the Improved FPN network. The FPN with a ResNet-50 pretrained model as the backbone network for image segmentation adds ResNet-50's strong feature extraction skills with the multi-scale feature fusion of FPN. This combination allows for accurate and detailed image segmentation. This model composed of the following components: 1) Backbone Network, in which the FPN network starts with a pretrained ResNet-50 as the backbone network. ResNet-50 is a highly effective deep CNN widely recognized for its outstanding performance in a range of computer vision tasks, particularly image classification. In the context of image segmentation, ResNet-50 is used as a feature extractor. The pretrained ResNet-50 takes an input image and processes it through a series of convolutional layers, max pooling, and residual blocks. These layers progressively downsample the spatial resolution of the input image while enhancing the number of channels, allowing the network to capture features at different levels of abstraction.

At different stages of ResNet-50, intermediate feature maps are obtained. Each feature map represents a various level of abstraction and has a specific spatial resolution. For example, the feature maps from the initial stages retain more fine-grained details, while the feature maps from the later stages capture higher-level semantic information. 2) Bottom-Up Pathway, in which the backbone network processes the input image through a series of convolutional layers, max pooling, and residual blocks. It gradually diminishes the spatial resolution while increasing the number of feature maps' channels. In ResNet-50, this reduction happens in four stages, each with a different spatial resolution.ResNet-50 consists of four stages. Stage 1 of ResNet-50 is the initial stage of the backbone network. It comprises of a single convolutional layer with a stride of 2 and a kernel size of 7x7, then a max pooling operation. The purpose of Stage 1 is to process the input image and down sample it while increasing the number of channels. It creates feature maps with a spatial resolution roughly equal to one-fourth of the resolution of the original image. Stage 2 is the second stage of ResNet-50 and includes a sequence of residual blocks. Specifically, it has three residual blocks, each consisting of multiple convolutional layers. The number of convolutional layers in each residual block is as follows: The first residual block in Stage 2 has two convolutional layers. The subsequent two residual blocks in Stage 2 have three convolutional layers each. These residual blocks in Stage 2 capture features at a relatively low level of abstraction while reducing the visual clarity. They down sample the feature maps further and increase the count of channels. The feature maps produced by Stage 2 have a spatial resolution of approximately 1/8th of the input image's resolution. Stage 3 is the third stage of ResNet-50 and comprises a series of residual blocks. It has four residual blocks, each comprised of multiple convolutional layers.

The feature maps produced by Stage 3 have a spatial resolution of approximately 1/16th of the input image's resolution. Stage 4 is the final stage of ResNet-50 and includes the last set of residual blocks. It has six residual blocks, each comprised of multiple convolutional layers. The count of convolutional layers within each residual block is specified as follows: Each of the six residual blocks in Stage 4 has three convolutional layers. These residual blocks in Stage 4 capture the most abstract and high-level features. They further down sample the feature maps and increase the number of channels. The feature maps produced by Stage 4 have the lowest spatial resolution among all stages, approximately 1/32nd of the input image's resolution, but the highest number of channels. The feature maps generated at different stages of ResNet-50 serve as the input for the subsequent components of the FPN, such as the top-down pathway, lateral connections, and semantic segmentation head.

3) Top-Down Pathway, in which the FPN incorporates a top-down pathway to recover the spatial resolution of the feature maps. It starts by upsampling the feature maps from higher stages of ResNet-50 to correspond with the spatial resolution of the feature maps from lower stages. The upsampling operation can be achieved using techniques like bilinear interpolation or transposed convolutions. 4) Lateral Connections: The FPN establishes lateral connections between the upsampled feature maps from the top-down pathway and the feature maps from the corresponding stages in the bottom-up pathway. These lateral connections involve applying 1x1 convolutions to the upsampled feature maps, reducing their channel dimension to match the number of channels in the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

feature maps from the bottom-up pathway. The resulting feature maps are then element-wise summed, creating fused feature maps that contain information at multiple scales. 5) Feature Fusion: To further improve multi-scale feature fusion, FPN recursively applies lateral connections.

This means that the upsampled feature maps from a higher stage are combined with the associated feature maps from a lower stage, and the process is repeated to create a feature pyramid with multiple levels of fused features. This pyramid captures both local details and global context at different scales. 6) Semantic Segmentation Head, where the fused feature maps from the FPN are passed through a semantic segmentation head. This head typically consists of additional convolutional layers that further process the feature maps and reduces the number of channels. The segmentation head produces a pixel-wise segmentation map, assigning a class label to each pixel in the output.

As shown in Figure 7, an improved version of the DeepLabv3+ architecture for image segmentation is designed. It consists of the following components: 1) Xception Model as Backbone Network, 2) Dilated Convolutions with Multiple Rates, 3) ASPP module, 4) Feature Fusion, and 5) Final Segmentation. The improved DeepLabv3+ architecture takes input images as its initial input. These images can vary in size and dimension. 1) DeepLabv3+'s core network is built on the Xception model. The Xception architecture is a variant of the Inception architecture that focuses on depth-wise separable convolutions. It provides efficient and effective feature extraction capabilities, capturing meaningful patterns and representations from the input images.

The Xception backbone network plays a crucial role in capturing low-level information and extracting high-level semantic elements. The Xception network is organized into three main sections: the Entry Flow, the Middle Flow, and the Exit Flow. The Entry Flow is the initial part of the Xception network. It is comprised of up of several convolutional and pooling layers. To extract low-level information, the input image is sent through a number of convolutional layers with a kernel of a tiny size (for example, 3x3). Batch normalization and ReLU activation functions are employed after every convolutional layer to introduce non-linearity and enhance feature representations. Max pooling operations are utilized to down sample the feature maps and enhance the receptive field gradually. The Middle Flow is the core part of the Xception network. It includes multiple stacked residual modules. The residual modules help to capture and refine more complex and abstract features as the network goes deeper. The residual connection, also known as a skip connection, allows the network to bypass the convolutional layers within each module and directly propagate information from earlier stages. The Middle Flow typically contains several residual modules stacked together to deepen the network and capture increasingly higher-level features. The Exit Flow is the final part of the Xception network. It further refines the feature representations and performs classification or prediction. It includes additional residual modules similar to the Middle Flow, followed by GAP layer to aggregate spatial information.

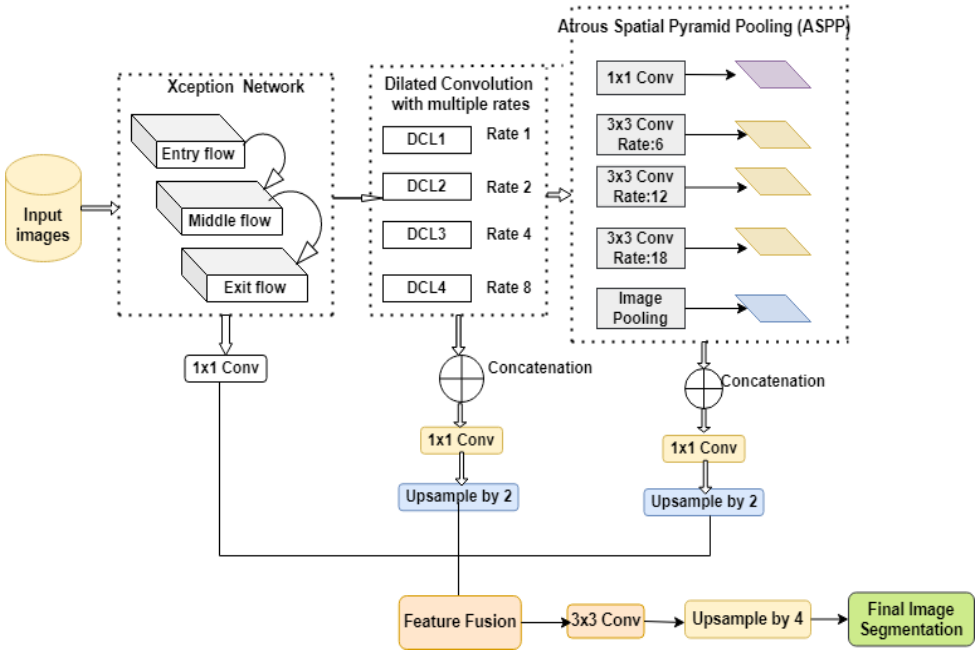

Figure 7: Architecture of the Improved DeepLab V3+

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

The feature maps are then passed through fully connected layers for classification or prediction tasks. Finally, appropriate activation functions (such as softmax) are applied to produce the final output probabilities or predictions. 2) DeepLabv3+ incorporates dilated convolutions with multiple rates to capture multi-scale information. Dilated convolutions, also known as atrous convolutions, expand the receptive field of each neuron without augmenting the parameter count. By applying dilations at different rates (1, 2, 4, and 8), the model can capture contextual information at multiple scales. This allows the network to effectively handle objects of distinguished sizes and capture both local details and global context.3) ASPP incorporates atrous (or dilated) convolutions with different rates.

Atrous convolutions, also known as dilated convolutions, involve introducing gaps or dilations between kernel elements to extent the receptive field without enhancing the number of parameters. ASPP utilizes atrous convolutions at multiple rates, such as rates 1, 6, 12, and 18, to derive context at different scales. By employing dilated convolutions with different rates, ASPP can incorporate information from various receptive fields and gather multi-scale context. GAP layer is a pooling operation that calculates the average value of each feature map across all spatial locations. In ASPP, GAP operation is performed to aggregate spatial information and captures the holistic context of the entire image. It allows the network to obtain a global view of the input and make predictions based on the overall context. ASPP includes atrous convolutions at multiple rates: 1, 6, 12, and 18. Each atrous convolution layer is responsible for capturing context at a specific scale, or receptive field size. By using multiple rates, the network can gather information at different scales simultaneously, enabling it to handle objects of varying sizes. Image pooling, also known as image-level pooling, is an additional step in ASPP that helps capture global context and incorporate information from the entire image. It involves downsampling the entire feature map to a single value by applying pooling operations (e.g., max pooling) across the entire spatial extent. The resulting pooled representation provides a high-level summary of the entire image, which can be useful for making global predictions or incorporating global context into the segmentation task. By combining atrous convolutions at different rates (1, 6, 12, and 18), global average pooling, and image pooling, The ASPP module strengthens the network's capacity to capture context at various sizes and combine both local and global data. 4) After the ASPP module, DeepLabv3+ performs feature fusion to combine features from various scales and steps of abstraction. This fusion allows the network to leverage the complementary information captured at various scales and create a more comprehensive feature representation. Fine-grained data and high-level semantics are combined to create fused features, which produce more detailed and meaningful visualizations of features. 5) In the final segmentation step, the fused features are upsampled to the original input image resolution. Upsampling is performed using techniques like transpose convolutions or interpolation, which recover the spatial information lost during the down sampling process.

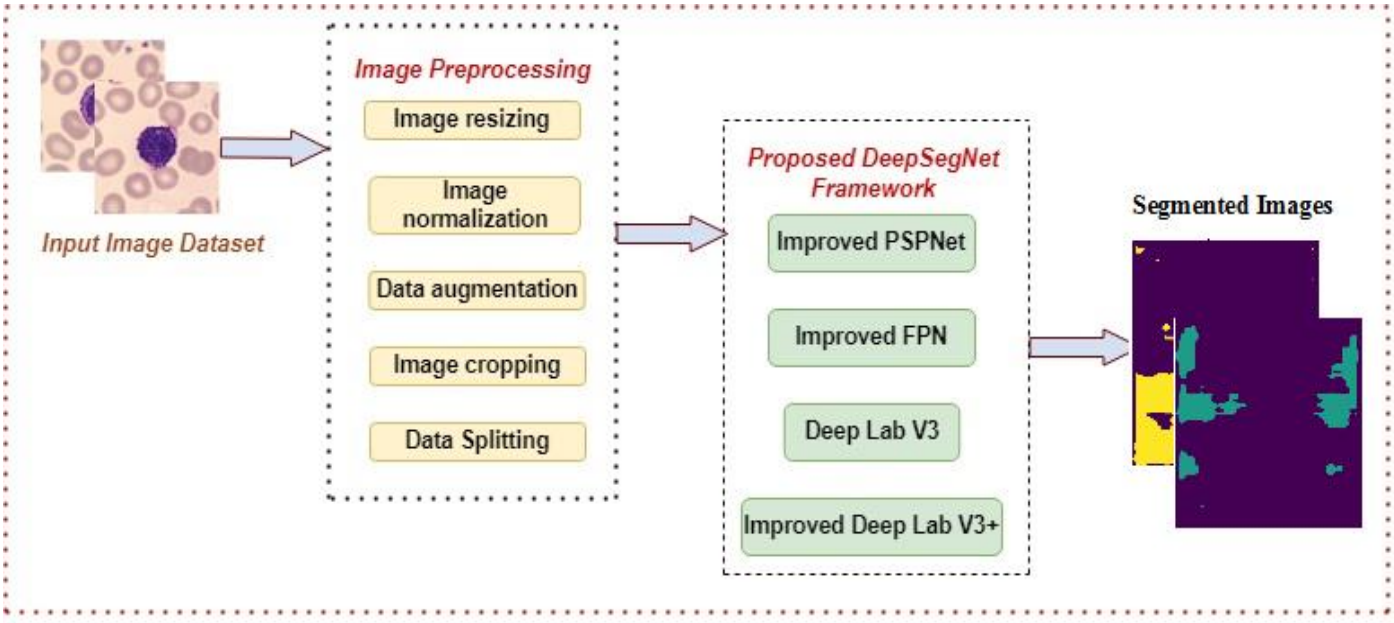

Figure 8: Overall flow diagram of the proposed DeepSegNet framework

The upsampled feature maps are passed through a 1x1 convolution layer to get the final pixel-wise segmentation predictions. Each pixel in the segmentation map represents the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

expected class label for that particular pixel in the input image. Figure 8 shows the overall flow diagram of the proposed DeepSegNet framework. Batches of images of blood cells are provided as input, and the images are preprocessed as explained in section 3.1.2. After preprocessing, images are passed into the DeepSegNet Framework, which includes four types of models, such as 1) Improved PSPNet 2) Improved FPN 3) Deep Lab V3 4) Improved Deep Lab V3+. The following Preprocessing Steps are carried out: a) Image Resizing is used to resize the input images to a consistent resolution to ensure compatibility with the segmentation models and to reduce computational complexity. b) Image normalization normalizes the pixel values of the images to remove variations in lighting conditions and improve model performance. Typical approaches include dividing by the image dataset's standard deviation and deducting the mean. c) To expand the diversity of training samples and enhance the model's capacity to handle multiple variations of object orientations, data augmentation uses various transformations, such as rotations, flips, translations, and scaling. d) Image Cropping is applied to the images. If the images contain large background areas or irrelevant regions, cropping the images to focus on the regions of interest can improve computational efficiency and model performance.

After preprocessing, the images are passed into the DeepSegNet Framework, in which the first network is An improved PSPNet model with pre-trained weights obtained from a big-scale dataset, such as ImageNet, is initialized. Deep CNN architecture is utilized to derive hierarchical features from the input image. SPP is applied to gather multi-scale contextual information by dividing the feature maps into different regions and performing pooling operations. The multi-scale features using skip connections and dilated convolutions are aggregated to enhance the representation power and contextual understanding. The aggregated features are used to perform convolutional operations to obtain pixel-wise predictions, and then the loss value is calculated between the predicted segmentation and ground truth masks utilizing an appropriate loss function. The second segmentation model, an improved FPN with pre-trained weights obtained from a large-scale dataset, is initialized. The features are derived from the input image using a backbone network. A feature pyramid is constructed by applying lateral connections to propagate high-resolution information from lower-level feature sets to higher-level feature sets. A fusion model enhances the quality of representation and captures multi-scale contextual information. The fused features are upsampled, convolutional operations are performed to generate pixel-wise predictions, and then loss value and accuracy values are calculated. As a third image segmentation model, the DeepLabv3 model with pre-trained weights is initialized. The features are extracted from the input image through the backbone network atrous convolutions at various rates gather multi-scale data while preserving spatial resolution. The ASPP module aggregates context at multiple scales and captures rich contextual information. After upsampling the feature maps, convolutional operations are performed to obtain pixel-wise predictions. The Improved DeepLabv3+ is setup with pre-trained weights before being used. The feature extraction is done from the Xception network, then the dilated convolution with multiple rates captures the features, and then the ASPP modules with multiple rates capture the multi-scale contextual information. The characteristics derived from the Xception network's feature extraction, dilated convolution module, and ASPP module are fused and upsampled by 4 to get the final segmentation output.

5. EXPERIMENTATION, RESULTS AND ANALYSIS

5.1 Experimental Setup

The experimentation with the proposed DeepSegNet Framework is conducted using the following system setup: a Core i7-10750H processor, renowned for high performance within Intel's Core i7 series, and 16 GB of DDR4 RAM, signifying enhanced system memory capacity for improved speed and efficiency. The entire experiment is executed on Google Colab, leveraging its practical and efficient environment tailored for deep learning.The torch.nn module in PyTorch that Google Colab supports is imported, which provides classes and functions for building neural networks. For creating neural network designs, it provides a variety of prebuilt layers, activation functions, loss functions, and tools. For image segmentation models, torch.nn enables the creation of custom architectures by defining modules such as convolutional layers, pooling layers, transposed convolutions, and skip connections. It also provides tools for weight initialization, gradient optimization, and parameter management. In addition, Google Colab provides users with access to various pre-installed libraries and tools for data analysis and visualization, such as NumPy and Matplotlib. The library torchviz is installed. tqdm is a popular Python library that is imported for the following reasons: a fast, extensible progress bar for loops and other iterable objects. It adds a progress bar to your loops to monitor the progress of iterations. It displays the current iteration count, estimated time remaining, and other useful information. CV2 (OpenCV) is a potent library for problems involving computer vision. It offers a variety of functions and algorithms for image and video processing, object detection, feature extraction, and more. The random library, a built-in Python module used in this research work, provides functionalities related to random number generation and randomization. The PIL Library is imported to perform image loading and Saving, Image

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Manipulation, Image filtering and enhancement, and image transformation operations. The GPU runtime was built up.

5.2 Performance Metrics

Accuracy measures the overall correctness of the segmentation predictions. It measures how many pixels in the image were correctly identified (including true positives and true negatives) as shown in equation (1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Recall determines how well a segmentation model can identify every pixel that is positive. It denotes the proportion of true positive (TP) pixels relative to the sum of true positive and false negative (FN) pixels, as shown in equation (2).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

The segmentation model's precision is how well it can distinguish positive events. It is the ratio of true positive (TP)

pixels to the total of pixels that are true positive and false positive (FP), as shown in equation (3).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

The ground truth mask and expected segmentation mask are compared using the Dice coefficient to determine how similar or overlapped they are. The masks must perfectly match for the Dice coefficient to be 1, which range from 0 to 1.Equation (4) illustrates the formula for computing the Dice coefficient.

$$\text{Dice coefficient} = \frac{(2 * |A \cap B|)}{(|A| + |B|)} \quad (4)$$

Where A denote the predicted segmentation mask (as a set of pixels), B represents the ground truth segmentation mask (a collection of pixels), |A ∩ B| demonstrates the cardinality (number of elements) of the intersection between A and B (similar pixels between the expected and ground truth masks), |A| represents the cardinality of A (total number of pixels in the predicted mask), and |B| represents the cardinality of B (The ground truth mask's overall pixel count). The numerator (2 * |A ∩ B|) represents the predicted and ground truth masks both share twice as many pixels. The denominator (|A| + |B|) represents the sum of pixels in both the predicted and ground truth masks. The anticipated and ground truth masks must match exactly for the Dice coefficient to be 1; otherwise, there is no overlap or similarity.

The Dice coefficient is frequently employed in the segmentation of medical images since it gives an indication of how well and closely the segmented regions match the ground truth annotations. It is frequently employed in additional picture segmentation tasks to assess the accuracy of the results.

5.3 Results

This section presents the results of the proposed DeepSegNet Framework. In the context of training and evaluating a segmentation model, the terms "train accuracy", "validation accuracy" "train loss" "validation loss" and "Dice Coefficient" are considered. Train accuracy would indicate how well the model predicts the correct segmentation labels for the training samples. Test accuracy provides an estimate of how well the

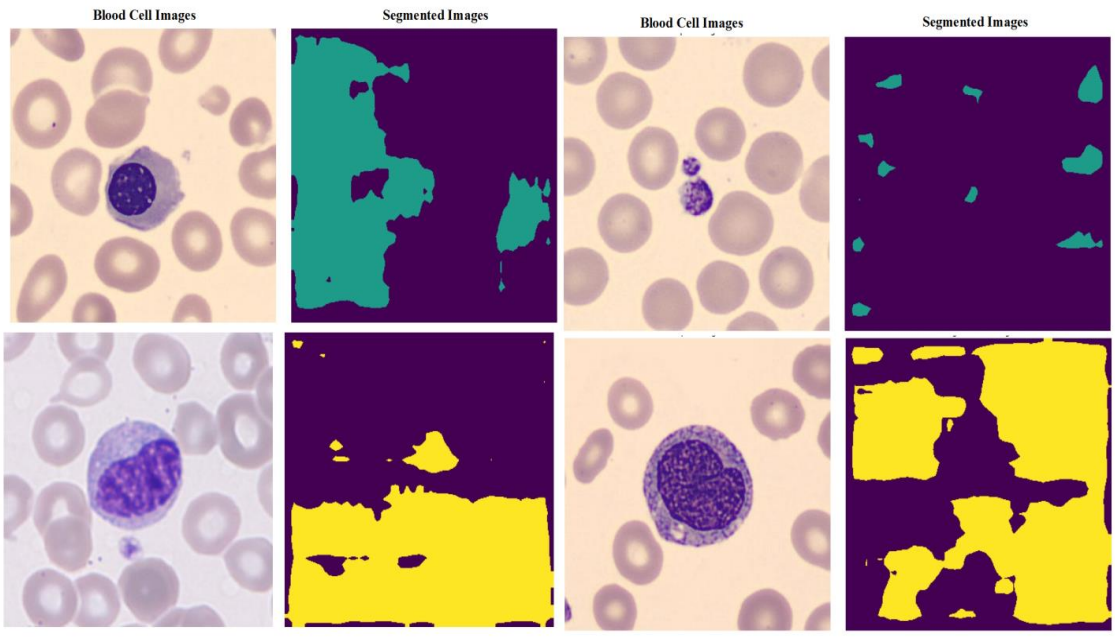

Figure: 9 Sample Blood cell images and segmented images by the DeepSegNet Framework

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

segmentation model generalizes and performs on new data. It measures the model's ability to correctly predict the segmentation labels for the test samples. Higher test accuracy shows that the model is better at precisely segmenting items or identifying areas of interest in fresh, untainted data. "Training loss" pertains to the computed loss or error during the training phase of the model. Loss is a measure of how well the model's predictions align with the ground-truth segmentation labels for the training samples. "Test loss" refers to the loss or error calculated on a separate test dataset. The test loss measures the model's performance and how well it generalizes to new, unseen data in terms of segmentation accuracy. The Dice coefficient is typically utilized as an evaluation metric rather than training metric. The Dice coefficient quantifies the concurrence between the predicted segmentation and the ground-truth segmentation.

Figure 9 shows sample blood cell images and segmented images by the DeepSegNet Framework, the Blood cell images are divided into distinct regions based on their characteristics, such as shape, color, texture, or intensity. The purpose of segmentation is to isolate individual cells or specific regions of interest within the image for further analysis or diagnosis. Segmented blood cell images can aid in diagnosing various diseases and medical conditions. By counting and analyzing different blood cell types, healthcare professionals can detect abnormalities or specific patterns indicative of certain diseases, such as anaemia, leukaemia, infections, and immune disorders. As shown in figure 10, the improved PSPNet model produced train loss values of 0.1526, 0.1531, 0.1514, 0.1505, 0.153, 0.1513, 0.1498, 0.1506, 0.1504, 0.151, 0.1496, 0.1508, 0.1497, and test loss values of 0.1506, 0.1495, 0.1498, 0.1496, 0.1487, 0.1484, 0.1482, 0.1481, 0.148, 0.1479, 0.1477, 0.1478, and 0.1476.

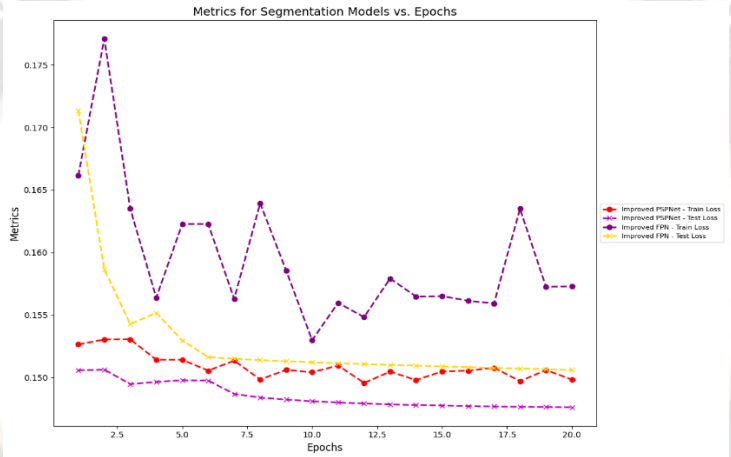

Figure 10: Performance of the improved PSPNet and FPN models in terms of train and test losses for Epochs 20

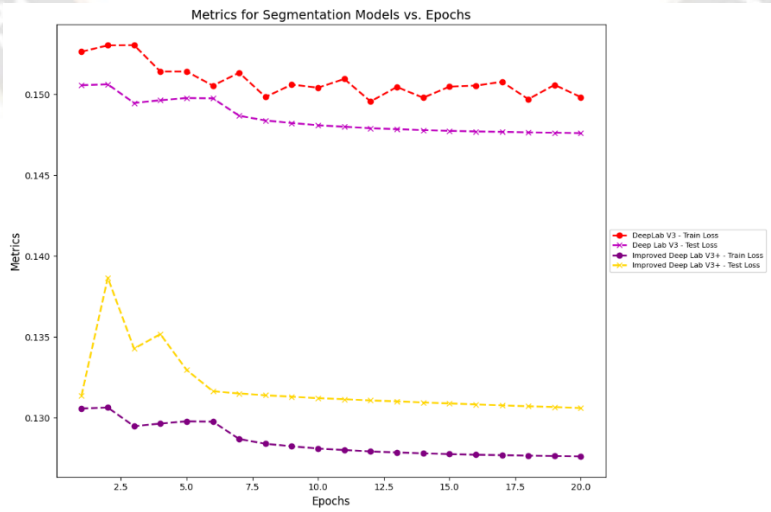

Figure 11: Performance of the improved DeepLabV3 and improved DeepLabV3+ models in terms of train and test losses for Epochs 20

The lowest test loss value is 1476, and the highest test loss value is 1506. The improved FPN model produced loss values of 0.1661, 0.1771, 0.1635, 0.1564, 0.1623, 0.1563, 0.1639, 0.1586, 0.153, 0.156, 0.1548, 0.1579, 0.1565, 0.1561, 0.1559,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

0.1635, 0.1573, and test loss values of 0.1714, 0.1543, 0.1552, 0.1587, 0.153, 0.1516, 0.1515, 0.1514, 0.1513, and 0.1512, 0.1511, 0.1508, 0.1506, 0.151, 0.1507, and the lowest test loss value are 0.1505, and the highest test loss value is 0.1713. As shown in figure 11, the improved PSPNet model produced train loss values of 0.1516, 0.1521, 0.1504, 0.1515, 0.1523, 0.1543, 0.1508, 0.1516, and 0.1503, and test loss values of 0.1507, 0.1525, 0.1507, 0.1506, 0.1527, 0.1514, 0.1502, and 0.1476.

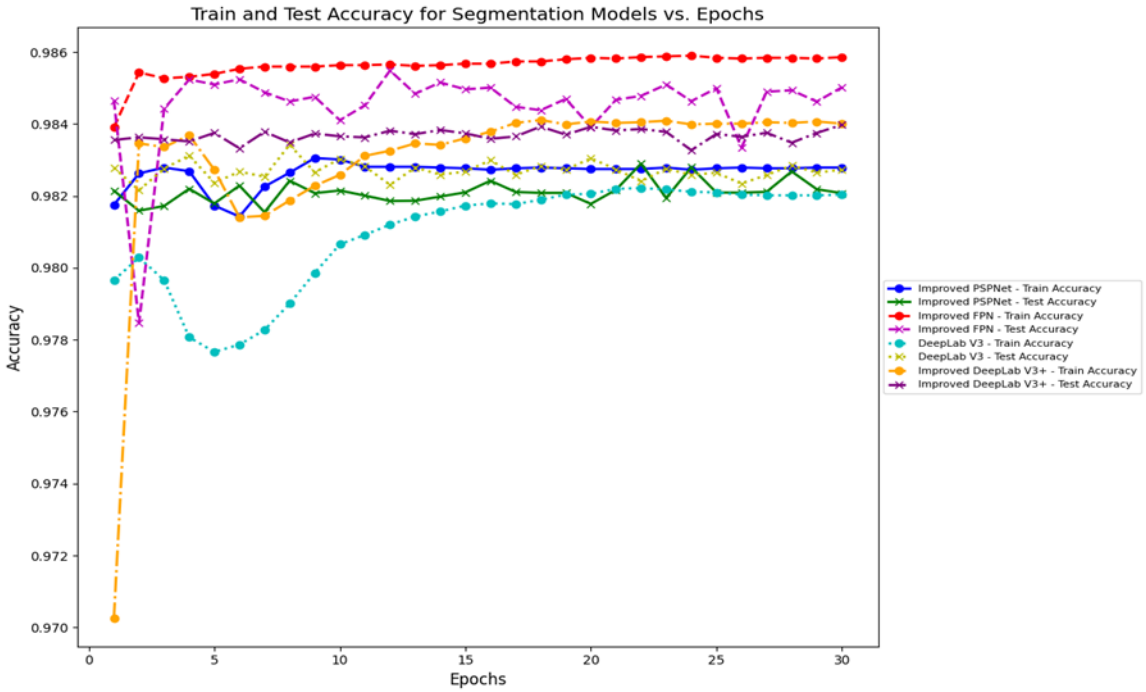

Figure14: DeepSegNet Framework's training, valid accuracy on the Blood cell Image Dataset for Epochs 30

The lowest test loss value is 1503, and the highest test loss value is 0.1543. The improved Deep Lab V3+ model produced loss values of 0.1305, 0.1306, 0.1294, 0.1297, and0.1283 and test loss values of 0.1313, 0.1386, 0.1342, 0.1587, 0.1351, 0.1329, and 0.1316. The lowest test loss value is 0.13070, and the highest test loss value is 0.1386. As shown in the figure 12, the Improved PSPNet model achieved train accuracy of 97.953966% and test accuracy of 97.902425%, and the Improved FPN model achieved train accuracy of 97.959391% and test accuracy of 97.941081%.The Deep Lab V3 model produced train accuracy of 97.637939% and test accuracy of 96.946208%, and the Improved Deep Lab V3+ model achieved train accuracy of 97.229004% and test accuracy of 97.062853% for 20 Epochs. As shown in Figure 13, the Improved FPN model produced the lowest train loss of 0.1354; the lowest test loss of 0.1334, and the highest train loss and test loss values of 0.1466 and 0.1384, respectively. The Improved FPN model produced train loss values of 0.1438, 0.1430, 0.1432, 0.1434, 0.1435, 0.14311, 0.14311, 0.1430, 0.1429, and test loss values of 0.1384, 0.1336, 0.1346, 0.1341, and 0.1343. The lowest train loss was 0.1434, the lowest test loss was 0.1337, and the highest train loss and test loss values were 0.1455 and 0.1339 for 30 epochs. The Deep Lab V3 model produced values of 0.116201, 0.123999, 0.152224, 0.188287, 0.190309, 0.175009, 0.168977, 0.165626, 0.158659, 0.144845, 0.134652, 0.064655, 0.064189, 0.063741, and test loss values of 0.1656, 0.1655, 0.1651, 0.1644, 0.1642, 0.16403, 0.16354, 0.16315, and 0.1622, respectively.

The Improved Deep Lab V3+ model produced train loss values of 0.083868, 0.084204, 0.08243, 0.082299, 0.081325, 0.080378, 0.079592, 0.079544, and test loss values of 0.116874, 0.128583, 0.131476, 0.139951, 0.148738, 0.14596, 0.143375, 0.13522, 0.129151, 0.119143, 0.112273, 0.102552, 0.096634, 0.068848, and 0.067934. The Improved Deep Lab V3+ model achieved the lowest train loss of 0.0795, the lowest test loss of 0.068355, and the highest train loss and test loss values of 0.083868 and 0.148738, respectively.

As shown in figure 14, the Improved PSPNet model achieved train accuracy of 98.3052% and test accuracy of 98.2794%, and the Improved FPN model achieved train accuracy of 98.5819% and test accuracy of 98.5243%.The Deep Lab V3 model produced train accuracy of 98.2116% and test accuracy of 98.3045%, and the Improved Deep Lab V3+ model achieved train accuracy of 98.4069% and test accuracy of 98.3920% for 30 Epochs. As shown in Figure 15, the Improved FPN model produced the lowest train loss of 0.1424; the lowest test loss of 0.1417; and the highest train loss and test loss values of 0.14309 and 0.1420, respectively.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

The Improved FPN model produced train loss values of 0.099253, 0.09938, 0.097974, 0.097807, 0.096537, 0.096898, 0.096079, 0.097132, 0.096061, 0.097142, 0.095454, 0.0968, 0.095906, 0.09637, 0.09687, 0.09544, and 0.1343. The lowest train loss was 0.096, the lowest test loss was 0.124, and the highest train loss and test loss values were 0.099 and 0.1263 for 30 epochs. The Deep Lab V3 model produced values of 0.116201, 0.123999, 0.152224, 0.188287, 0.190309, 0.175009, 0.168977, 0.165626, 0.158659, 0.144845, 0.134652, 0.064655, 0.064189, 0.063741, and test loss values of 0.139225, 0.148018, 0.148454, 0.149173, 0.152419, 0.159772, 0.141621, 0.150529, 0.156062, 0.133358, and 0.1229, respectively. The Improved Deep Lab V3+ model produced train loss values of 0.085667, 0.086042, 0.085728, 0.085696, 0.085732, 0.085588, 0.085557, 0.065244, 0.065327, 0.075212, and test loss values of 0.097177, 0.098164, 0.10056, 0.101527, 0.0965, 0.096066, and 0.08533. The Improved Deep Lab V3+ model achieved the lowest train loss of 0.0652, the lowest test loss of 0.0652, and the highest train loss and test loss values of 0.0853 and 0.10812, respectively, for epochs 40.As shown in figure 16, the Improved PSPNet model achieved train accuracy of 98.3425% and test accuracy of 98.2520%, and the Improved FPN model achieved train accuracy of 99.2453% and test accuracy of 99.0406%. The Deep Lab V3 model produced train accuracy of 98.3527% and test accuracy of 98.2299%, and the Improved Deep Lab V3+ model achieved train accuracy of 99.4745% and test accuracy of 99.3057% for 40 Epochs. The Improved Deep Lab V3+ model produced train loss values of 0.085667, 0.086042, 0.085728, 0.085696, 0.085732, 0.085588, 0.085557, 0.065244, 0.065327, 0.075212, and test loss values of 0.097177, 0.098164, 0.10056, 0.101527, 0.0965, 0.096066, and 0.08533.The Improved Deep Lab V3+ model achieved the lowest train loss of 0.0652, the lowest test loss of 0.0652, and the highest train loss and test loss values of 0.0853 and 0.10812, respectively, for epochs 40.

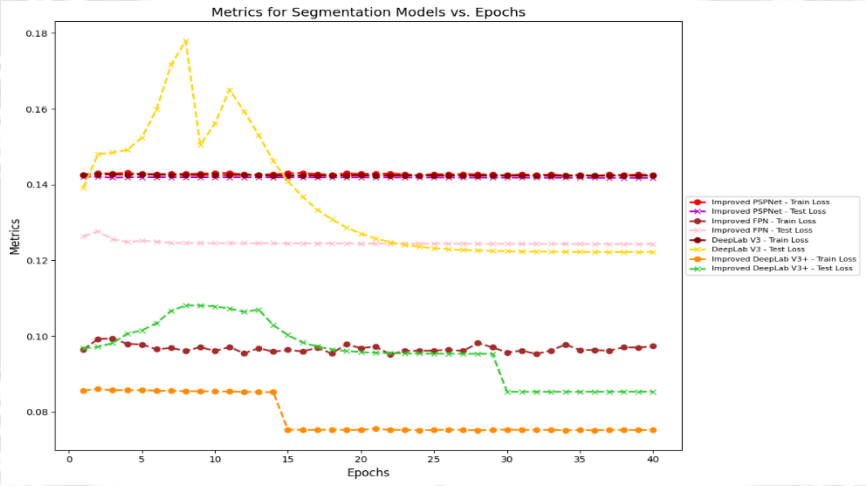

Figure 15: Performance of the improved DeepLabV3 and improved DeepLabV3+ models in terms of train and test losses for Epochs 40

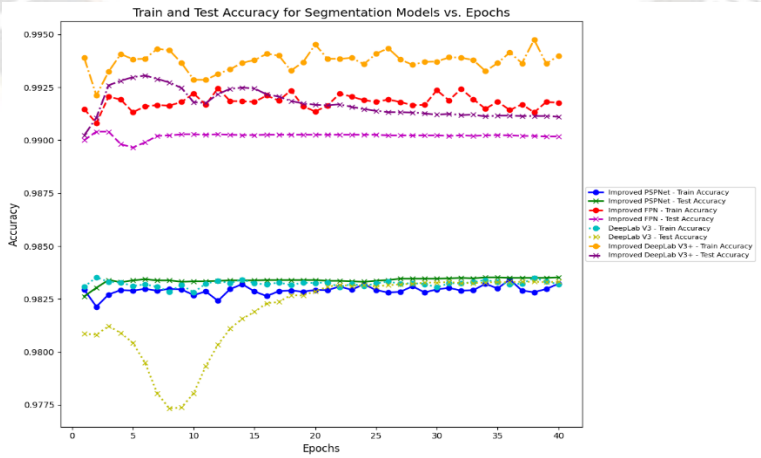

Figure 16: DeepSegNet Framework's training, valid accuracy on the Blood cell Image Dataset for Epochs 40

As shown in figure 16, the Improved PSPNet model achieved train accuracy of 98.3425% and test accuracy of 98.2520%, and the Improved FPN model achieved train accuracy of 99.2453% and test accuracy of 99.0406%. The Deep Lab V3 model produced train accuracy of 98.3527% and test accuracy of 98.2299%, and the Improved Deep Lab V3+ model achieved train accuracy of 99.4745% and test accuracy of 99.3057% for 40 Epochs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

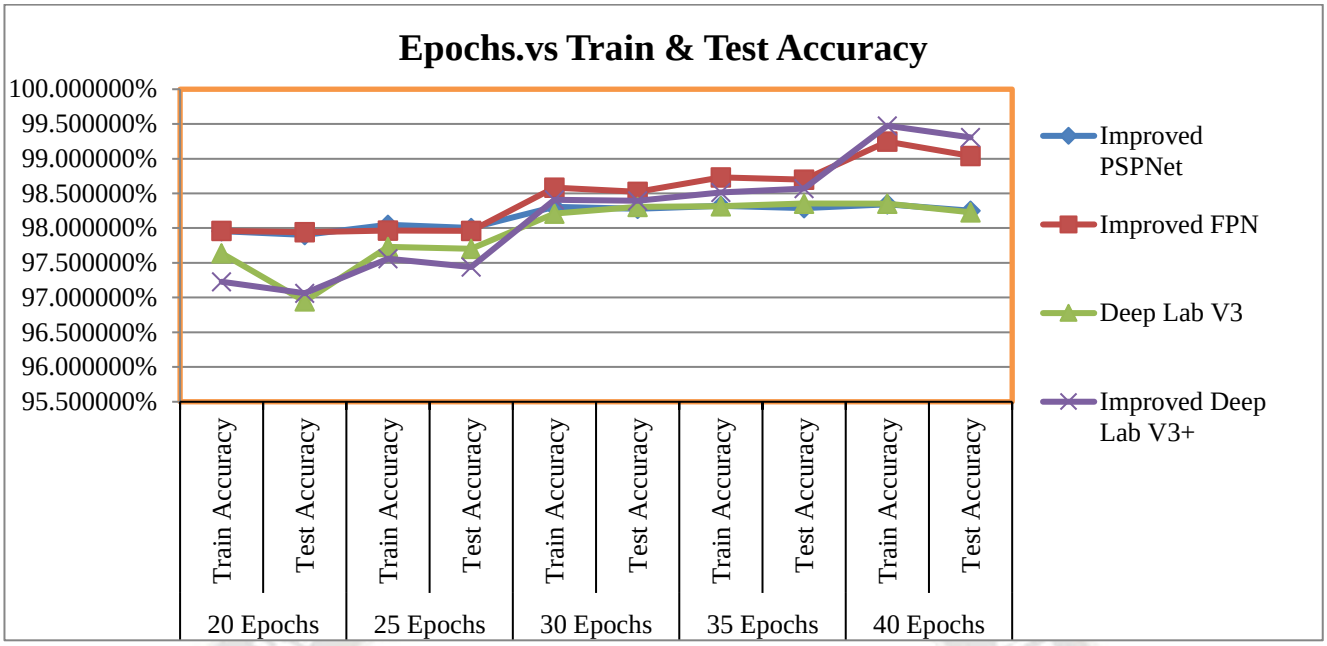

Figure 17: Performance of the proposed frame work based on training & Validation accuracy

5.4 Analysis

This section explains the analysis of the DeepSegNet Framework models in terms of various metrics and epochs and the comparison of the proposed framework with existing techniques. As shown in figure 17, the Improved PSPNet model achieved train accuracy of 98.050265% and test accuracy of
98.002% and the Improved FPN model achieved train accuracy of 97.963460% and test accuracy of 97.961171%. The Deep Lab V3 model produced train accuracy of 97.7294% and test accuracy of 97.70372%, and the Improved Deep Lab V3+ model achieved train accuracy of 97.56062% and test accuracy of 97.438557% for 25 Epochs. The Improved PSPNet model achieved train accuracy of 98.319498% and test accuracy of 98.287624%, and the Improved FPN model achieved train accuracy of 98.732503% and test accuracy of 98.699951%.

The Deep Lab V3 model produced train accuracy of 98.31746% and test accuracy of 98.35476%, and the Improved Deep Lab V3+ model achieved train accuracy of 98.5148% and test accuracy of 98.57063% for 35 Epochs.

Table 1 shows the training and testing Accuracy Results of the Proposed DeepSegNet framework for various Epochs on the Blood Cell image dataset. DeepSegNet framework models produced above 98% train-and-test accuracy from 30 epochs onward. The performance of the improved PSPNet and Deep Lab V3 models is more similar. The Improved FPN model produced the second-highest train accuracy of 99.245388% and test accuracy of 99.0406413, and the Improved Deep Lab V3+ model produced the highest training and test accuracy of 99.4745497% and 99.3057454%, respectively. These results demonstrate that the proposed framework does not suffer from overfitting or under fitting problems.

Table 2 shows the performance of the proposed DeepSegNet framework on the blood cell image dataset. The Improved PSPNet model produced a recall of 98.35%, Precision of 98.24%, accuracy of 98.252%, and a Dice Coefficient of 98.3%. The Improved FPN model produced a recall of 98.99%, Precision of 99.15%, accuracy of 99.04%, and a Dice Coefficient of 99.07%. The Deep Lab V3 model produced a recall of 98.30%, Precision of 98.24%, accuracy of 98.23%, and a Dice Coefficient of 98.27%. The Deep Lab V3 model produced a recall of 99.27%, Precision of 99.38%, accuracy of 99.31%, and a Dice Coefficient of 99.32%. As shown in figure 18, the Improved Deep Lab V3+ model produced the highest Recall of 99.27%, precision of 99.38%, accuracy of 99.31%, and Dice Coefficient of 99.32% among all the models in the proposed framework. The improved FPN model produced the second highest Recall of 98.99%, precision of 99.15%, accuracy of 99.04%, and Dice Coefficient of 99.07%. The Improved PSPNet received slightly higher results than the Deep Lab V3 model, but the result is lower than the other models such as Deep Lab V3+ and the Improved FPN model. Table 3 shows the comparison analysis of the proposed DeepSegNet framework with the existing Segmentation model based on accuracy. The improved DeepLab V3+ model has produced higher values of 1.741%, 4.01%, 2.39%, 5.21%, 4.91%, 10.01%, and 3.61% than the existing models such as WBC nucleus segmentation [25], Neural ordinary differential [27], circle Hough Transform (CHT)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

TABLE 1:

TRAIN AND TEST ACCURACY RESULTS OF THE PROPOSED DEEPSEGNET FRAMEWORK FOR VARIOUS EPOCHS

		Improved PSPNet	Improved FPN	Deep Lab V3	Improved Deep Lab V3+
20 Epochs	Train Accuracy	97.953965929%	97.959391276%	97.637939453%	97.229003906%
	Test Accuracy	97.902425130%	97.941080729%	96.946207682%	97.062852648%
25 Epochs	Train Accuracy	98.050265842%	97.963460286%	97.729492188%	97.560628255%
	Test Accuracy	98.002115885%	97.961171468%	97.703721788%	97.438557943%
30 Epochs	Train Accuracy	98.305257161%	98.581949870%	98.211669922%	98.406982422%
	Test Accuracy	98.279486762%	98.524305556%	98.304578993%	98.392062717%
35 Epochs	Train Accuracy	98.319498698%	98.732503255%	98.317464193%	98.514811198%
	Test Accuracy	98.287624783%	98.699951172%	98.354763455%	98.570632935%
40 Epochs	Train Accuracy	98.342556424%	99.245388500%	98.352728950%	99.474549700%
	Test Accuracy	98.252050781%	99.040641300%	98.229980469%	99.305745400%

TABLE 2:

PERFORMANCE OF THE MODELS ON THE BLOOD CELL IMAGE DATASET

Performance metrics	Improved PSPNet	Improved FPN	Deep Lab V3	Improved Deep Lab V3+
Recall	98.35%	98.99%	98.30%	99.27%
Precision	98.24%	99.15%	98.24%	99.38%
Accuracy	98.25%	99.04%	98.23%	99.31%
Dice Coefficient	98.30%	99.07%	98.27%	99.32%

[30], semi-supervised method [34], TransFuse [48], TransFuse [48], Eres-UNet++ [50], and PSPNet [45]. All the proposed models have achieved higher accuracy than all the existing models. Table 4 shows the comparison analysis of the proposed DeepSegNet framework with the existing Segmentation model based on the dice coefficient. The improved DeepLab V3+ model has produced higher values of 8.91%, 3.72%, 17.73%, 2.83%, 7.96%, 9.61%, 17.36%, 6.22%, 13.32%, and 14.32% than the existing models such as the UNeXt method [38], DRINet model [32], segmentation with a diffusion-probabilistic model [35], DoubleU-Net [36], UNet++ architecture [40], TransUNet architecture [41], MISSFormer network [43], hybrid Transformer [39], semi-supervised method [34], and CNN [46]. All the proposed models have achieved a higher Dice Coefficient than all the existing models. Table 5 shows the comparison analysis of the proposed DeepSegNet framework with the existing Segmentation model based on Precision and over segmentation rates. The improved DeepLab V3+ model has produced higher values of 11.75%, 0.9%, 7.59%, and 30.38% than the existing models such as WBC nucleus segmentation [25], WBC-Net [26], adaptive Thresholding [29], and Self-supervised learning [28].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

TABLE 3
COMPARISON OF THE PROPOSED FRAMEWORK WITH EXISTING MODELS BASED ON ACCURACY

S. No	Segmentation models & Methods	Performance metric(Accuracy)
1	WBC nucleus segmentation[25]	97.57%
2	Neural ordinary differential equations [27]	95.30%
3	Circle Hough Transform (CHT) [30]	96.92%
4	Semi supervised method[34]	94.10%
5	TransFuse[48]	94.40%
6	Eres-UNet++ [50]	89.30%
7	PSPNet[45]	95.70%
Proposed DeepSegNet Framework		
8	Improved PSPNet	98.25%
9	Improved FPN	99.04%
10	Deep Lab V3	98.23%
11	Improved Deep Lab V3+	99.31%

.

TABLE 4
COMPARISON OF THE PROPOSED FRAMEWORK WITH EXISTING MODELS BASED ON DICE COEFFICIENT

S. No	Segmentation models & Methods	Performance metric (Dice Coefficient)
1	UNeXt method [38]	90.41%
2	DRINet model [32]	95.60%
3	Segmentation with diffusion-probabilistic model[35]	81.59%
4	DoubleU-Net [36]	76.49%
5	UNet++ architecture [40]	91.36%
6	TransUNet architecture [41]	89.71%
7	MISSFormer network [43]	81.96%.
8	Hybrid Transformer[39]	93.10%
9	Semi supervised method[34]	86%
10	CNN[46]	85%
Proposed DeepSegNet Framework		
11	Improved PSPNet	98.30%
12	Improved FPN	99.07%
13	Deep Lab V3	98.27%
14	Improved Deep Lab V3+	99.32%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

TABLE 5

COMPARISON OF THE PROPOSED FRAMEWORK WITH EXISTING MODELS

S. No	Segmentation models & Methods	Performance metric(Others)
1	WBC nucleus segmentation[25]	87.63%(precision)
2	WBC-Net[26]	98.48%(precision)
3	Adaptive Thresholding [29]	91.79%(Precision)
4	Self-supervised learning [28]	69% (Over segmentation-rate)
Proposed DeepSegNet Framework		
5	Improved PSPNet	98.24%(Precision)
6	Improved FPN	99.15%(Precision)
7	Deep Lab V3	98.24%(Precision)
8	Improved Deep Lab V3+	99.38%(Precision)

All the proposed models have achieved higher precision than all the existing models. The proposed DeepSegNet framework demonstrated superior performance compared to all existing techniques in various aspects, including accuracy, Dice coefficient, precision, and more. Notably, the proposed framework effectively addressed the gaps identified in the existing techniques, such as lack of accuracy and other metrics, Long Training Time, inefficiency in variations and Artifacts , interpretability, and lack of generalization to new data, by improving the existing segmentation models, especially concentrating on capturing multi-scale information and extracting local and global features accurately by adding different dilated convolutions, and adding the module of feature fusion and other operations to the proposed segmentation models to get the essential and effective features and improve the quality of the features. The diversity of the data has been improved to help prevent overfitting and improve the model's ability to handle different orientations, scales, and positions of blood cells. Additionally, DeepSegNet exhibited efficient computation, leading to reduced processing time while consistently yielding high-quality segmentation results. The incorporation of proper preprocessing techniques further enhanced the overall effectiveness of the proposed approach.

6. CONCLUSION AND FUTURE WORK

This paper proposes an innovative DeepSegNet framework for Accurate Blood Cell Image segmentation. Improved PSPNet, Improved FPN, Deep Lab V3, and Improved Deep Lab V3+ are improved to fulfil the gaps identified in the literature survey. The model's generalization ability, the diversity of the data, and stability during training are increased using preprocessing techniques. The proposed framework has been improved to capture multi-scale information and extract local and global features accurately by adding different dilated convolutions, and the module of feature fusion has been added to the proposed segmentation models to get the essential and effective features and improve the quality of the features. The DeepSegNet framework saves computation time by reducing the number of epochs and has high accuracy, dice coefficient, and precision. The blood cell image dataset is applied to the proposed DeepSegNet framework, which consists of models such as the Improved PSPNet, Improved FPN, Deep Lab V3, and Improved Deep Lab V3+, which performed well and produced 98.25%, 99.04%, 98.23%, and 99.31% accuracy, respectively. Among the proposed segmentation models, the Improved Deep Lab V3+ model outperformed well and produced a Dice Coefficient of 99.32% and Precision of 99.38%, but the proposed framework is moderately complex in design, and it will be addressed in the future by reducing the number of layers and changing the model architecture. In our future work, clinical data, genetic data, and Blood cell data validated by medical experts will be incorporated for evaluation, and the proposed framework will be experimented with other medical image datasets and other data sets. Leveraging expert annotations in the form of additional supervision will be considered during model training. Graph-based representations or incorporating scene context to improve segmentation in complex scenes will be investigated. New methods will be designed for uncertainty estimation in segmentation predictions, particularly important in medical

applications, to identify cases where the model may be uncertain about its segmentation results.

REFERENCES

[1] A. P. Dhawan, "MEDICAL IMAGE ANALYSIS."
[2] M. Adnan and Q. Muhammad, "Medical Image Analysis using Convolutional Neural Networks: A Review."
[3] G. Lee, *Deep Learning in Medical Image Analysis*. .
[4] J. B. Freund, "Numerical Simulation of Flowing Blood Cells," 2013, doi: 10.1146/annurev-fluid-010313-141349.
[5] S. Alférez and A. Acevedo, "Image processing and machine learning in the morphological analysis of blood cells," vol. 40, no. February, pp. 46–53, 2018, doi: 10.1111/ijlh.12818.
[6] B. Marieb and N. Elaine, "Essen ti al s of Hum an A n atom y & Ph ysi ol ogy (10th Edi ti on)."
[7] "amjpathol00256-0002.pdf." .
[8] M. Petrou, "Image Processing :"
[9] "REVIEW THE ORGANIZATION OF PROTEINS IN THE A Review," vol. 62, pp. 1–19, 1974.
[10] R. Tomari *et al.*, "Computer Aided System for Red Blood Cell Classification in Blood Smear Image," *Procedia - Procedia Comput. Sci.*, vol. 42, pp. 206–213, 2014, doi: 10.1016/j.procs.2014.11.053.
[11] M. Taherisadr, M. Nasirzonouzi, B. Baradaran, and A. Mehdizade, "New Approch to Red Blood Cell Classification Using Morphological Image Processing," vol. 14, no. 1, 2013.
[12] J. A. T and K. J. Friston, "Unified segmentation," vol. 26, pp. 839–851, 2005, doi: 10.1016/j.neuroimage.2005.02.018.
[13] J. P. Gowda and S. C. P. Kumar, "Segmentation of White Blood Cell using K-Means and Gram-Schmidt Orthogonalization," vol. 10, no. February, 2017, doi: 10.17485/ijst/2017/v10i6/111137.
[14] H. T. Madhloom and S. A. Kareem, "An Image Processing Application for the Localization and Segmentation of Lymphoblast Cell Using Peripheral Blood Images," pp. 2149–2158, 2012, doi: 10.1007/s10916-011-9679-0.
[15] A. C. Sparavigna, A. C. Sparavigna, and P. Torino, "Measuring the blood cells by means of an image segmentation To cite this version : HAL Id : hal-01654006 Measuring the blood cells by means of an image segmentation," 2017.
[16] N. Theera-Umpon, "White blood cell segmentation and classification in microscopic bone marrow images," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3614 LNAI, no. January, pp. 787–796, 2006, doi: 10.1007/11540007.
[17] F. Sadeghian, Z. Seman, A. R. Ramli, B. Hisham, A. Kahar, and M. Saripan, "A Framework for White Blood Cell Segmentation in Microscopic Blood Images Using Digital Image Processing," vol. 11, no. 1, pp. 196–206, doi: 10.1007/s12575-009-9011-2.
[18] J. M. Sharif, M. F. Miswan, M. A. Ngadi, M. S. H. Salam, and M. M. B. A. Jamil, "Red blood cell segmentation using masking and watershed algorithm: A preliminary study," *2012 Int. Conf. Biomed. Eng. ICoBE 2012*, no. February, pp. 258–262, 2012, doi: 10.1109/ICoBE.2012.6179016.
[19] 1–7. Dorini, Leyza BaldoDorini, L. B., Minetto, R., & Jerˆ Based on Multiscale Analysis, N. (2011). Based on Multiscale Analysis, 6(1), R. Minetto, and N. Jerˆ, "Based on Multiscale Analysis," vol. 6, no. 1, pp. 1–7, 2011.
[20] C. Fatichah, M. L. Tangel, M. R. Widyanto, F. Dong, and K. Hirota, "Interest-based ordering for fuzzy morphology on white blood cell image segmentation," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 16, no. 1, pp. 76–86, 2012, doi: 10.20965/jaciii.2012.p0076.
[21] S. S. Savkare and S. P. Narote, "Blood cell segmentation from microscopic blood images," *Proc. - IEEE Int. Conf. Inf. Process. ICIP 2015*, pp. 502–505, 2016, doi: 10.1109/INFOP.2015.7489435.
[22] J. Ma *et al.*, "Loss odyssey in medical image segmentation," *Med. Image Anal.*, vol. 71, 2021, doi: 10.1016/j.media.2021.102035.
[23] A. S. A. Nasir and M. Y. Mashor, "Unsupervised Colour Segmentation of White Blood Cell for Acute Leukaemia Images," pp. 1–4, 2011.
[24] B. C. Ko, J. Gim, and J. Nam, "Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake," *Micron*, vol. 42, no. 7, pp. 695–705, 2011, doi: 10.1016/j.micron.2011.03.009.
[25] P. P. Banik, R. Saha, and K. D. Kim, "An Automatic Nucleus Segmentation and CNN Model based Classification Method of White Blood Cell," *Expert Syst. Appl.*, vol. 149, p. 113211, 2020, doi: 10.1016/j.eswa.2020.113211.
[26] Y. Lu, X. Qin, H. Fan, T. Lai, and Z. Li, "WBC-Net: A white blood cell segmentation network based on UNet++ and ResNet," *Appl. Soft Comput.*, vol. 101, p. 107006, 2021, doi: 10.1016/j.asoc.2020.107006.
[27] D. Li *et al.*, "Robust Blood Cell Image Segmentation Method Based on Neural Ordinary Differential Equations," *Comput. Math. Methods Med.*, vol. 2021, 2021, doi: 10.1155/2021/5590180.
[28] X. Zheng, Y. Wang, G. Wang, and J. Liu, "Fast and robust segmentation of white blood cell images by self-supervised learning," *Micron*, vol. 107, pp. 55–71,

2018, doi: 10.1016/j.micron.2018.01.010.

[29] E. P. Mandyartha, F. T. Anggraeny, F. Muttaqin, and F. A. Akbar, "Global and Adaptive Thresholding Technique for White Blood Cell Image Segmentation," *J. Phys. Conf. Ser.*, vol. 1569, no. 2, 2020, doi: 10.1088/1742-6596/1569/2/022054.

[30] S. N. Mohd Safuan, M. R. Md Tomari, and W. N. Wan Zakaria, "White blood cell (WBC) counting analysis in blood smear images using various color segmentation methods," *Meas. J. Int. Meas. Confed.*, vol. 116, pp. 543–555, 2018, doi: 10.1016/j.measurement.2017.11.002.

[31] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-Net: Towards Unified Image Segmentation," *Adv. Neural Inf. Process. Syst.*, vol. 13, no. NeurIPS, pp. 10326–10338, 2021.

[32] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "DRINet for Medical Image Segmentation," *IEEE Trans. Med. Imaging*, vol. 37, no. 11, pp. 2453–2462, 2018, doi: 10.1109/TMI.2018.2835303.

[33] J. Gao, B. Wang, Z. Wang, Y. Wang, and F. Kong, "A wavelet transform-based image segmentation method," *Optik (Stuttg).*, vol. 208, p. 164123, 2020, doi: 10.1016/j.ijleo.2019.164123.

[34] X. Li, L. Yu, H. Chen, C. W. Fu, L. Xing, and P. A. Heng, "Transformation-Consistent Self-Ensembling Model for Semisupervised Medical Image Segmentation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 2, pp. 523–534, 2021, doi: 10.1109/TNNLS.2020.2995319.

[35] T. Amit, T. Shaharbany, E. Nachmani, and L. Wolf, "SegDiff: Image Segmentation with Diffusion Probabilistic Models," 2021, [Online]. Available: http://arxiv.org/abs/2112.00390.

[36] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "DoubleU-Net: A deep convolutional neural network for medical image segmentation," *Proc. - IEEE Symp. Comput. Med. Syst.*, vol. 2020-July, no. 1, pp. 558–564, 2020, doi: 10.1109/CBMS49503.2020.00111.

[37] Y. Liu *et al.*, "PaddleSeg: A High-Efficient Development Toolkit for Image Segmentation," 2021, [Online]. Available: http://arxiv.org/abs/2101.06175.

[38] J. M. J. Valanarasu and V. M. Patel, "UNeXt: MLP-Based Rapid Medical Image Segmentation Network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13435 LNCS, pp. 23–33, 2022, doi: 10.1007/978-3-031-16443-9_3.

[39] Y. Gao, M. Zhou, and D. N. Metaxas, "UTNet: A Hybrid Transformer Architecture for Medical Image Segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12903 LNCS, pp. 61–71, 2021, doi: 10.1007/978-3-030-87199-4_6.

[40] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020, doi: 10.1109/TMI.2019.2959609.

[41] J. Chen *et al.*, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," pp. 1–13, 2021, [Online]. Available: http://arxiv.org/abs/2102.04306.

[42] C. F. Baumgartner *et al.*, "PHiSeg: Capturing Uncertainty in Medical Image Segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11765 LNCS, pp. 119–127, 2019, doi: 10.1007/978-3-030-32245-8_14.

[43] X. Huang, Z. Deng, D. Li, and X. Yuan, "MISSFormer: An Effective Transformer," 2021, [Online]. Available: http://arxiv.org/abs/2109.07162.

[44] M. Li *et al.*, "Image Projection Network: 3D to 2D Image Segmentation in OCTA Images," *IEEE Trans. Med. Imaging*, vol. 39, no. 11, pp. 3343–3354, 2020, doi: 10.1109/TMI.2020.2992244.

[45] X. Zhu, Z. Cheng, S. Wang, X. Chen, and G. Lu, "Coronary angiography image segmentation based on PSPNet," *Comput. Methods Programs Biomed.*, vol. 200, p. 105897, 2021, doi: 10.1016/j.cmpb.2020.105897.

[46] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep Learning-Based Image Segmentation on Multimodal Medical Imaging," vol. 3, no. 2, pp. 162–169, 2019.

[47] Y. Xie, J. Zhang, and C. Shen, "CoTr : Efficiently Bridging CNN and," pp. 23–25.

[48] Y. Zhang, H. Liu, and Q. Hu, "TransFuse: Fusing Transformers and CNNs for Medical Image Segmentation," vol. 2, pp. 1–11.

[49] C. V Jun, "Input Augmentation with SAM : Boosting Foundation Model."

[50] J. Li *et al.*, "Eres-UNet ++ : Liver CT image segmentation based on high-effi- ciency channel attention and Res-UNet ++," no. May, 2023, doi: 10.1016/j.compbiomed.2022.106501.

[51] O. Sharif, M. M. Hoque, A. S. M. Kayes, R. Nowrozy, and I. H. Sarker, "applied sciences Learning Techniques," pp. 1–23, 2020.

[52] S. Bhadula*, S. Sharma, P. Juyal, and C. Kulshrestha, "Machine Learning Algorithms based Skin Disease Detection," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 2, pp. 4044–4049, 2019, doi:

10.35940/ijitee.b7686.129219.

[53] A. K. Verma, S. Pal, and S. Kumar, "Comparison of skin disease prediction by feature selection using ensemble data mining techniques," *Informatics Med. Unlocked*, vol. 16, 2019, doi: 10.1016/j.imu.2019.100202.

[54] P. R. Kshirsagar, H. Manoharan, S. Shitharth, A. M. Alshareef, N. Albishry, and P. K. Balachandran, "Deep Learning Approaches for Prognosis of Automated Skin Disease," *Life*, vol. 12, no. 3, 2022, doi: 10.3390/life12030426.

[55] M. Ahammed, M. Al Mamun, and M. S. Uddin, "A machine learning approach for skin disease detection and classification using image segmentation," *Healthc. Anal.*, vol. 2, no. April, p. 100122, 2022, doi: 10.1016/j.health.2022.100122.

[56] A. Jain, A. C. S. Rao, P. K. Jain, and A. Abraham, "Multi-type skin diseases classification using OP-DNN based feature extraction approach," *Multimed. Tools Appl.*, vol. 81, no. 5, pp. 6451–6476, 2022, doi: 10.1007/s11042-021-11823-x.

[57] S. Jinnai, N. Yamazaki, Y. Hirano, Y. Sugawara, Y. Ohe, and R. Hamamoto, "The development of a skin cancer classification system for pigmented skin lesions using deep learning," *Biomolecules*, vol. 10, no. 8, pp. 1–13, 2020, doi: 10.3390/biom10081123.

[58] K. Aljohani and T. Turki, "Automatic Classification of Melanoma Skin Cancer with Deep Convolutional Neural Networks," *Ai*, vol. 3, no. 2, pp. 512–525, 2022, doi: 10.3390/ai3020029.

[59] G. Arora, A. K. Dubey, Z. A. Jaffery, and A. Rocha, "Bag of feature and support vector machine based early diagnosis of skin cancer," *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8385–8392, 2022, doi: 10.1007/s00521-020-05212-y.