ED6001: Medical Image Analysis - Term Project Report

White blood cell classification using deep learning models

Mohammad Al Fahim K (EE21S050) Adarsh Singh Niranjan (EE21D023)

Objective

The diagnosis of blood-based diseases often involves identifying and characterizing patient blood samples. Automated methods to detect and classify blood cell subtypes have important medical applications. Thus, the objective of this project is to train various deep learning models on a medical image dataset for classification of WBCs into its subtypes. The dataset is provided by MIT: https://www.kaggle.com/datasets/paultimothymooney/blood-cells.

Introduction

Blood cells are categorized as Red Blood Cells (Erythrocytes), White Blood Cells (Leukocytes), and platelets.

WBCs are a part of the immune system. They fight infection and defend the body against other foreign materials. WBCs recognize intruders, kill harmful bacteria and make antibodies to protect your body against exposure to bacteria and viruses.

There are four main categories of WBCs: Eosinophil, Lymphocyte, Monocyte, and Neutrophil.

- Eosinophils fight bacteria and parasites but also provoke allergy symptoms.
- Lymphocytes are B and T cells that defend against specific invaders.
- Monocytes clean up dead cells.
- Neutrophils are the first responder of immune cells.

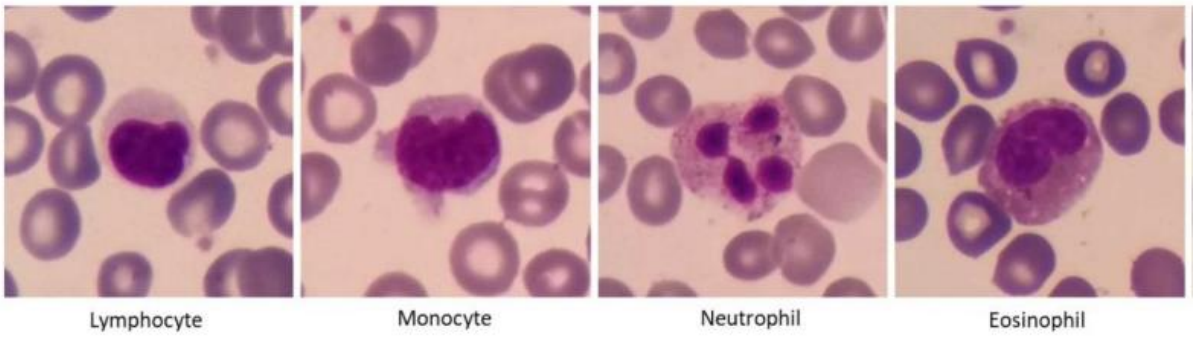

Figure 1. Types of white blood cells in normal peripheral blood [1]

A high white blood cell count is usually a sign of an infection, stress, cancer, inflammation and other illness. Low white blood cell counts can leave vulnerable to serious infections and diseases like aplastic anemia, lupus, Splenic sequestration etc. Thus, determining the correct type and number of white blood cells is very important for diagnosing these diseases.

But manual diagnosis i.e., microscopic examination of blood samples is a time consuming, expensive and error-prone task. Because of these hardships, a computerized approach for detection of WBCs could significantly reduce required time for manual testing, and decrease human errors. Thus, in this report, we are training various deep learning techniques on the available dataset having more than 10k samples of WBCs to classify them into its subtypes.

Dataset

We have used open-source datasets in our work which is hosted on Kaggle licenced by MIT. This dataset contains 12,444 augmented images of White blood cells (JPEG) which is divided into 2 folders – Train and Test. Train folder having 9957 images grouped into 4 different folders (according to cell type). The cell types are Eosinophil, Lymphocyte, Monocyte, and Neutrophil. Similarly, in the Test folder 2487 images can be found belonging to the above mentioned 4 cell types. The Size of the dataset is ≈ 105MB.

Deep Neural Networks

- **ResNet101V2**

"Deep Residual Learning for Image Recognition" was published on Dec 10, 2015 and as of today, it is one of the most cited papers in machine learning [2]. The major roadblocks in building deeper networks are vanishing gradient and degradation. ResNet solves these problems by adding the input of a layer to its output.

ResNet101V2 is a modified version of ResNet101 that performs better than ResNet50 and ResNet101 on the ImageNet dataset. In ResNet50V2, a modification was made in the propagation formulation of the connections between blocks. There are a total of 101 layers.

Layer name	**Output size**	**101-layers**
Conv 1	112 x 112	7 x 7, 64, stride 2
Conv2_x	56 x 56	3 x 3 max pool, stride 2
		$[1x1,64\ 3x3,64\ 1x1,256\ ]$ x 3
Conv3_x	28 x 28	$[1x1,128\ 3x3,128\ 1x1,512\ ]x4$
Conv4_x	14 x 14	$[1x1,256\ 3x3,256\ 1x1,1024\ ]x23$
Conv5_x	7 x 7	$[1x1,512\ 3x3,512\ 1x1,2048\ ]x3$
	1x1	Average pool, 100-d fc, softmax
FLOPs		7.6×10^9

Figure 2. ResNet101V2 architecture

❖ Xception

Xception is an extreme version of Inception. It is even better than Inception-v3. In the Xception model, depthwise separable convolution (i.e., depthwise convolution followed by a pointwise convolution) is used [3]. Compared with conventional convolution, we do not need to perform convolution across all channels. That means the number of connections is fewer and the model is lighter. The modified depthwise separable convolution is the pointwise convolution followed by a depthwise convolution**.** This modification is motivated by the inception module in Inception-v3.

The Xception architecture: the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow.

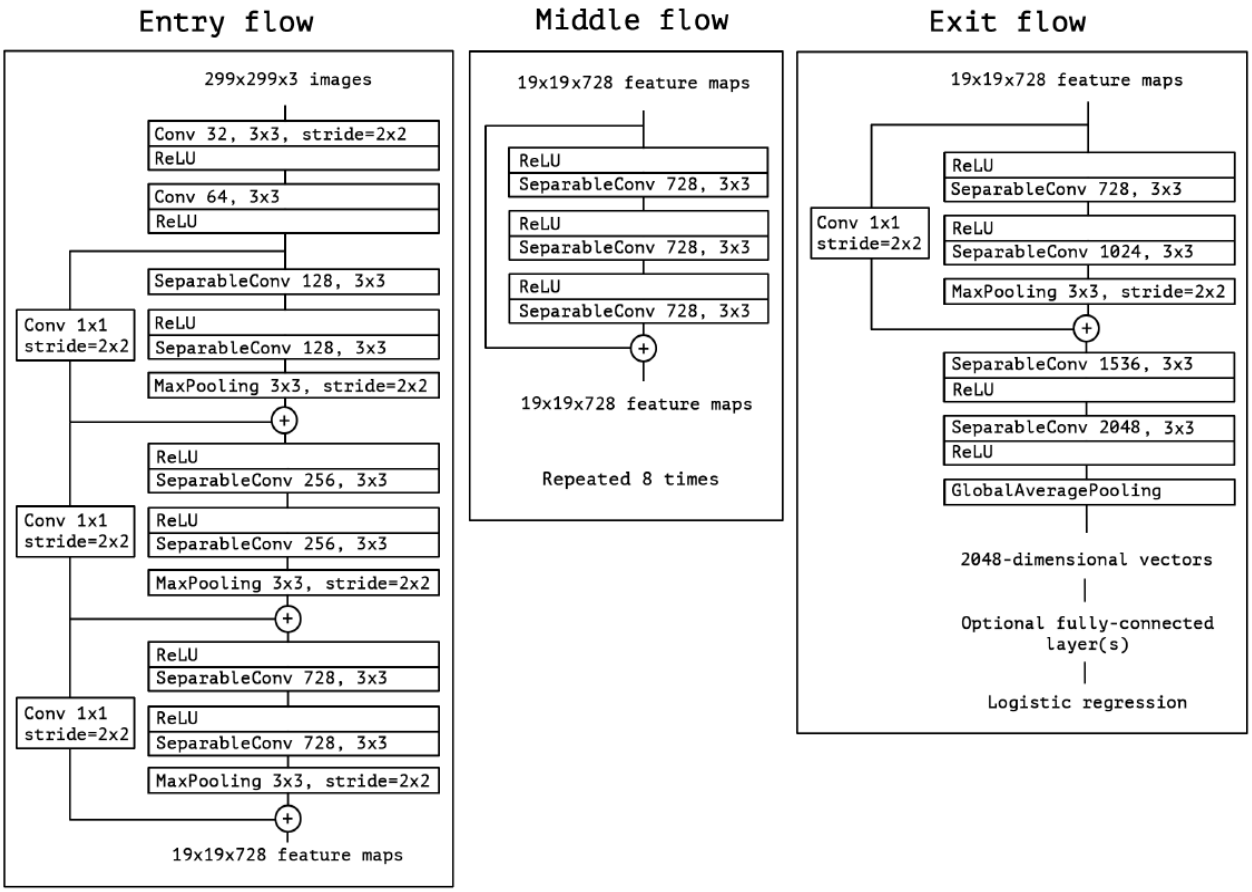

Figure 3. Xception Architecture

❖ InceptionResNetV2

InceptionResNet combines the Inception architecture with residual connections. InceptionResNetV2 is a variation of InceptionV3 model which borrows some ideas from ResNet [4]. In the case of InceptionResNet, we used batch-normalization only on top of the traditional layers, but not on top of the summations. In the Inception-Resnet block, multiple sized convolutional filters are combined with residual connections. The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time.

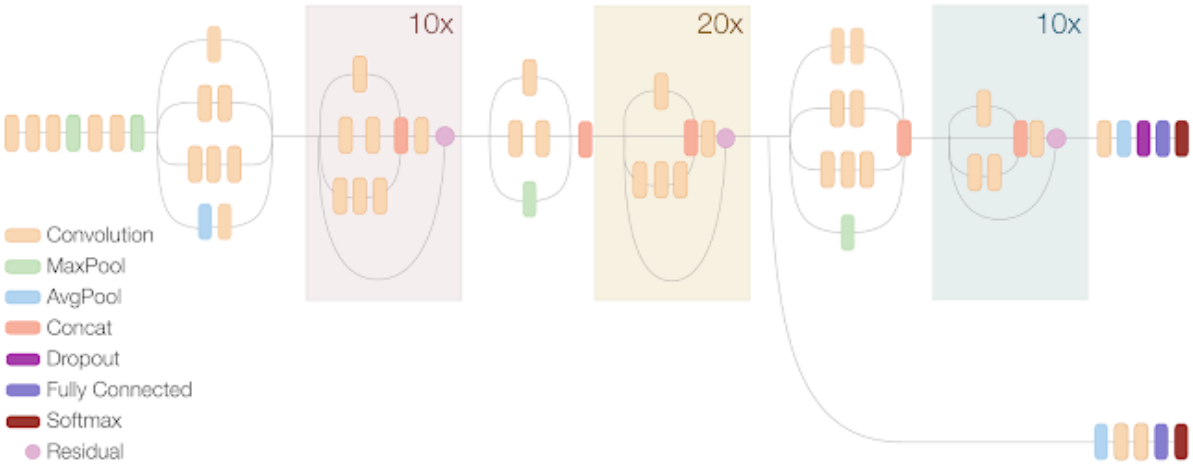

Figure 4. Schematic diagram of Inception-ResNet-v2

Segmentation

❖ Graph-cut Method

The Graph Cut technique applies graph theory to image processing to achieve fast segmentation. In the graph-cut method, each image is represented as a graph of nodes. Each node corresponds to an image pixel and links connecting the nodes are called edges. A pathway is constructed connecting all the edges to travel across the graph. To travel across the graph from a start node to an end node, the preferred path is the route in which the total weight sum is at a minimum. Therefore, in order to create path preferences, weights are assigned to individual edges. In a nutshell, this algorithm involves calculating the weights of the original image and then initializing the path endpoints. Finally, the shortest path is found by limiting the search region [5].

The technique creates a graph of the image where each pixel is a node connected by weighted edges. The higher the probability that pixels are related the higher the weight. The algorithm cuts along weak edges, achieving the segmentation of objects in the image. The algorithm cuts along weak edges, achieving the segmentation of objects in the image.

❖ K-means clustering

The K-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. K-means is one of the oldest and most

approachable algorithms and guarantees convergence [6]. Implementation of *k*-means clustering is reasonably straightforward.

Algorithm:

1. Specify the number of clusters K
2. Randomly initialize K centroids
3. Repeat
4. expectation: Assign each point to its closest centroid
5. maximization: Compute the new centroid (mean) of each cluster
6. until the centroid positions do not change.

Training:

The models were written and trained in Keras and Tensorflow, in python. The categorical cross entropy was used as loss function to train the models and an Adam optimizer with a learning rate of 0.0001. All the models were trained for 30 epochs and the models with the best validation Area under ROC curve (AUC) scores were used for testing.

Results

Classification

Model	AUC	Accuracy	Sensitivity	Specificity	Precision
Resnet101V2	99.43	95.81	86.67	98.87	96.25
Xception	98.62	94.49	83.3	98.23	94
Inception-ResNet-V2	98.2	92.84	91.49	93.3	82

Table 1: Results of Eosinophil classification

Model	AUC	Accuracy	Sensitivity	Specificity	Precision
Resnet101V2	99.99	98.99	96.13	99.94	99.83
Xception	99.99	99.43	97.9	99.94	99.83
Inception-ResNet-V2	99.99	99.19	99.83	98.98	97.02

Table 2: Results of Lymphocyte classification

Model	AUC	Accuracy	Sensitivity	Specificity	Precision

Resnet101v2	97.14	92.8	75	98.71	95.09
Xception	98.23	94.41	79.35	99.41	97.81
Inception-ResNet-v2	99.67	97.58	93.7	98.87	96.51

Table 3: Results of Monocyte classification

Model	AUC	Accuracy	Sensitivity	Specificity	Precision
Resnet101v2	97.38	89.22	95.19	87.22	71.39
Xception	97.92	89.46	94.55	87.76	72.12
Inception-ResNet-v2	96.16	78.04	97.69	91.88	92.76

Table 4: Results of Neutrophil classification

Model	AUC	Accuracy	Loss
Resnet101v2	98.49	88.42	0.3307
Xception	98.69	88.9	0.3470
Inception-ResNet-v2	98.51	91.15	0.3085

Table 5: Average of classification results

Segmentation

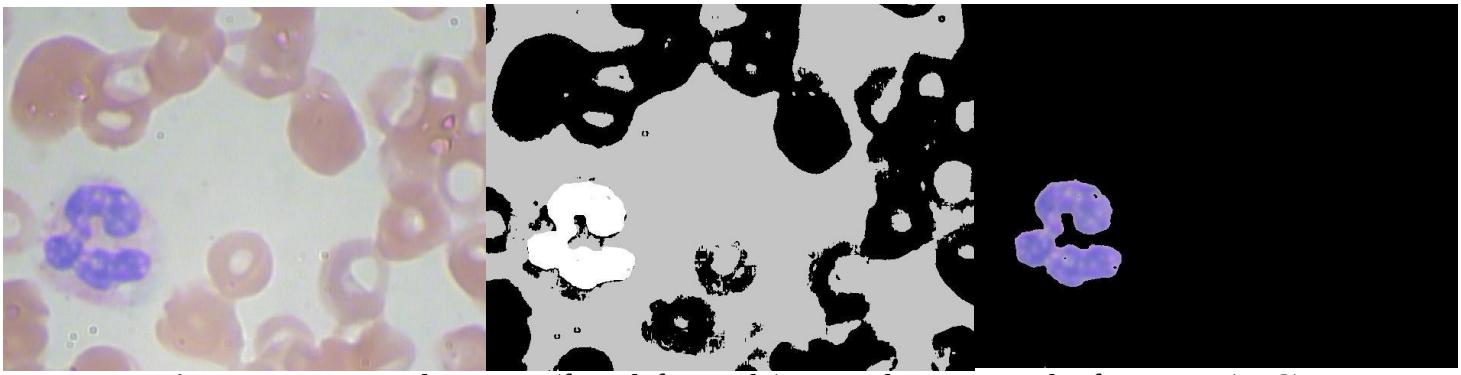

Figure 5. K means clustering: (from left to right) original image, result of K-means(K=3), segmented image

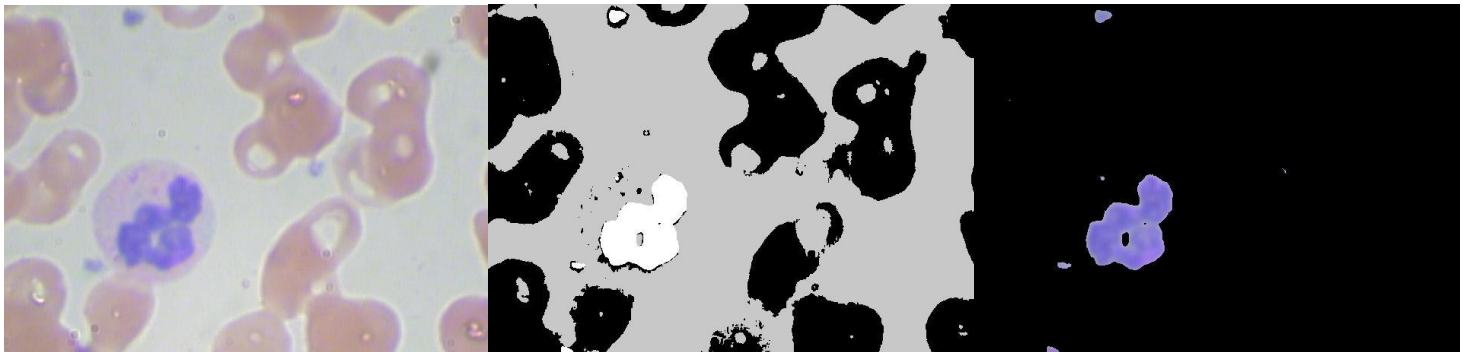

Figure 6. K means clustering: (from left to right) original image, result of K-means(K=3), segmented image

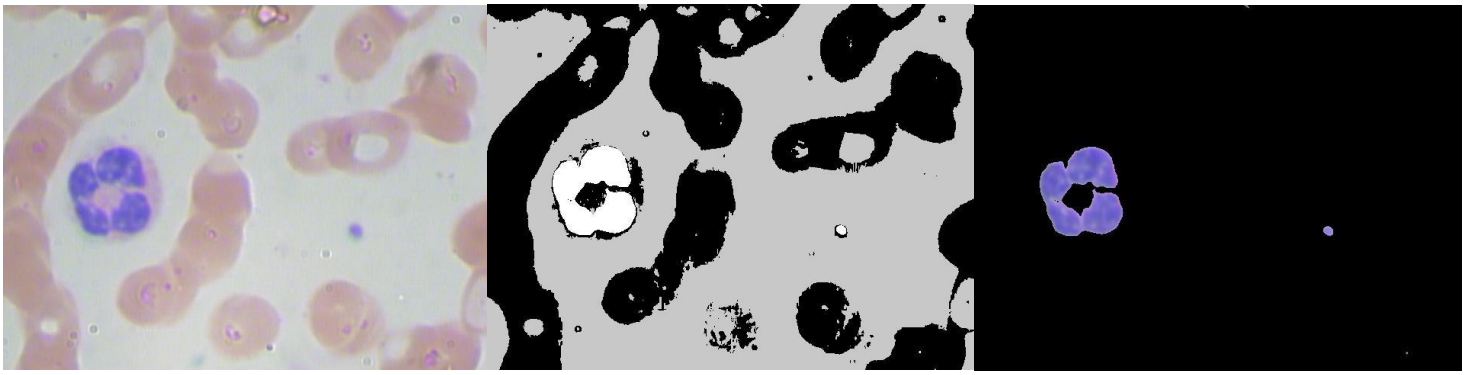

Figure 7. K means clustering: (from left to right) original image, result of K-means(K=3), segmented image

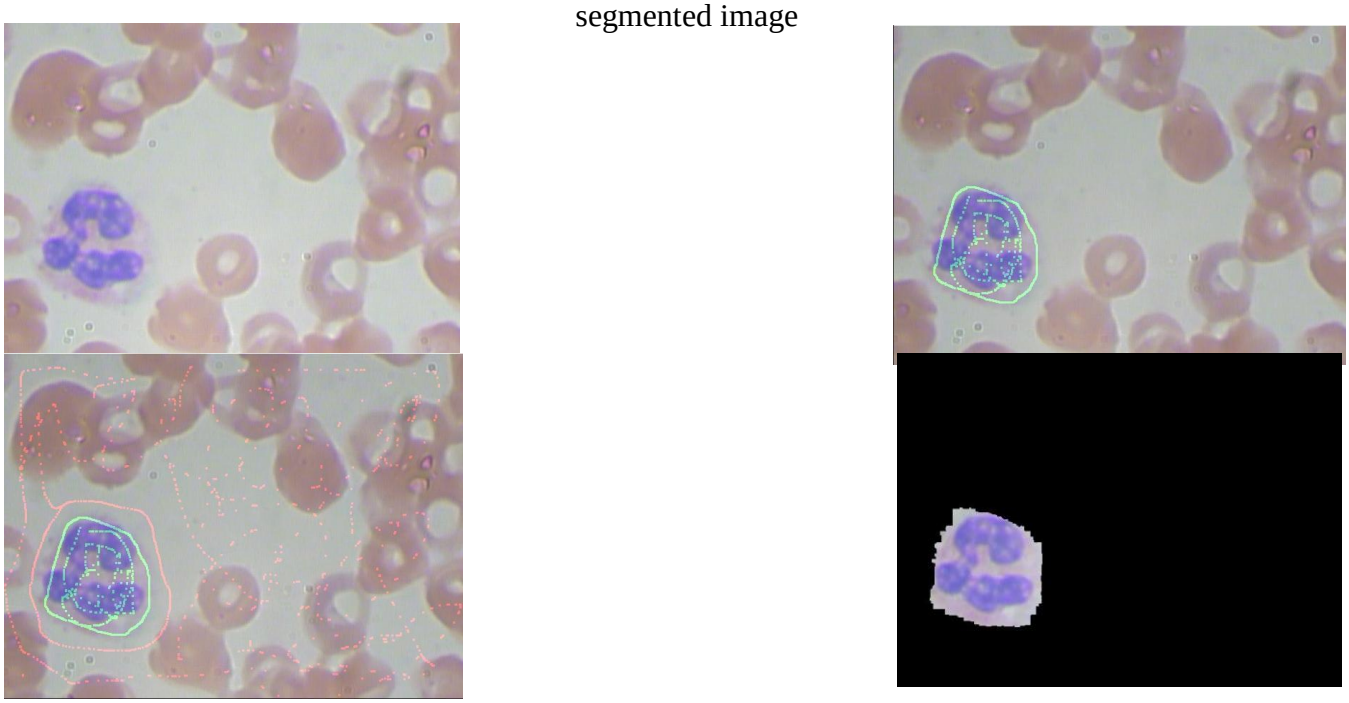

Figure 8 Graph Cuts: (from left to right and top to bottom) original image, (green) foreground seed points, (red) background seed points, segmented image

Figure 9. Graph Cuts: (from left to right and top to bottom) original image, (green) foreground seed points, (red) background seed points, segmented image

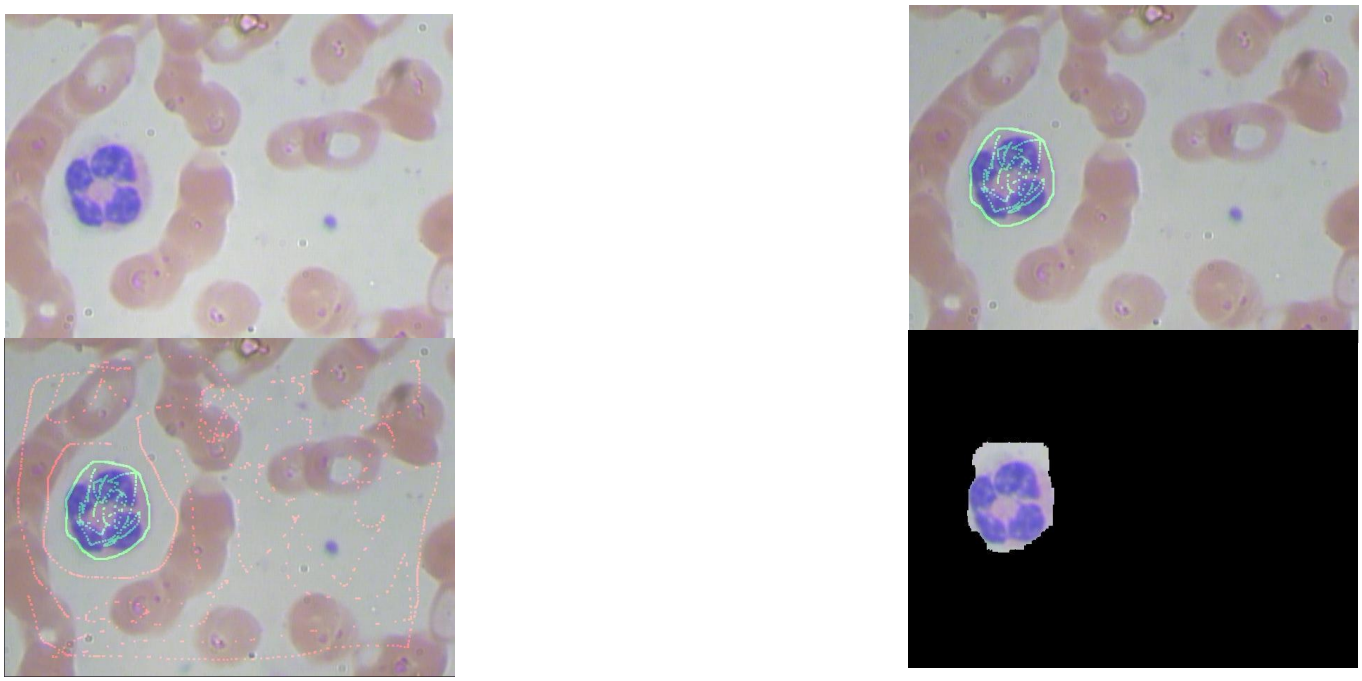

Figure 10. Graph Cuts: (from left to right and top to bottom) original image, (green) foreground seed points, (red) background seed points, segmented image

Conclusion

In this project, we have trained Resnet101V2, Xception and InceptionResnetV2 CNN models for classification of WBCs into its subtypes (Eosinophil, Lymphocyte, Monocyte, and Neutrophil). We obtained the AUC, accuracy, sensitivity, specificity and precision scores for all three models with respect to each subtype. Then after evaluating the average AUC,

sensitivity and loss of all three models, we can conclude InceptionResnetV2 performed better than the rest models.

We also performed k-means clustering and Graph-cut for segmenting WBCs from blood samples. The performance of both the methods is satisfactory, but these methods are manual and interactive, thus a time-consuming and rigorous process.

References

1. Z. M. Kouzehkanan, S. Saghari, "Raabin-WBC: a large free access dataset of white blood cells from normal peripheral blood", bio-Rxiv 2021.05.02.442287; doi: https://doi.org/10.1101/2021.05.02.442287
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
3. F. Chollet, **"**Xception: Deep Learning with Depthwise Separable Convolutions"**,** Oct. 2016. https://arxiv.org/abs/1610.02357
4. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning", *Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17),* pp. 4278–4284, 2017.
5. Stephanie J. Chiu, Yuliya Lokhnygina, Adam M. Dubis, Alfredo Dubra, Joseph Carroll, Joseph A. Izatt, and Sina Farsiu, "Automatic cone photoreceptor segmentation using graph theory and dynamic programming," Biomed. Opt. Express 4, pp. 924-937, 2013.
6. MacQueen, J. B.*, "*Some Methods for classification and Analysis of Multivariate Observations*", Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press., pp. 281-297, 1967.