Blood cells image segmentation and counting using deep transfer learning

1st Gharbi Aghiles
University M'hamed Bougara
Independence Avenue, 35000
Boumerdes, Algeria
a.gharbi@univ-boumerdes.dz

2nd Neggazi Mohamed Lamine
University M'hamed Bougara
Independence Avenue, 35000
Boumerdes, Algeria
neggazimedlamine@gmail.com

3rd Touazi Faycal
University M'hamed Bougara
Independence Avenue, 35000
Boumerdes, Algeria
f.touazi@univ-boumerdes.dz

4th Yagoubi Mohamed Riad
University M'hamed Bougara
Independence Avenue, 35000
Boumerdes, Algeria
md.yagoubi@univ-boumerdes.dz

5th Gaceb Djamel
University M'hamed Bougara
Independence Avenue, 35000
Boumerdes, Algeria
dj.gaceb@gmail.com

***Abstract*—Blood cell counting is a tedious task that would benefit greatly from automation. Accurate cell counting CBC (Complete Blood Count) provides essential quantitative information and plays a key role in biological research as well as industrial and biomedical applications. Unfortunately, the commonly used manual counting method is time consuming, poorly standardized and not reproducible. The task is made even more difficult by overlapping cells and poor imaging quality. In this paper we compared between two convolutional neural networks used for segmenting cells as a first phase. As a second phase we count the cells present in the output masks of the CNN models using 3 different algorithms (Watershed, Connected Component Labeling and Circle Hough Transform), where we added a loss function for the CHT. We obtained good results compared to other methods to assist pathologists and medical technicians.**

***Index Terms*—Blood cell segmentation, Blood cell counting, U-net, SegNet, deep learning**

I. Introduction

Blood carries out many vital functions as it circulates through the body. It transports oxygen from the lungs to other body tissues and carries away carbon dioxide. It carries nutrients from the digestive system to the cells of the body, and carries away wastes for excretion by the kidneys. Blood helps our body fight off infectious agents and inactivates toxins, stops bleeding through its clotting ability, and regulates our body temperature. Doctors rely on many blood tests to diagnose and monitor diseases. Some tests measure the components of blood itself; others examine substances found in the blood to identify abnormal functioning of various organs. Hence, we here propose a software system which will assist pathologists to detect blood cell count and help to find out the diseases. This information can be very helpful to:
e.g. a physician who is trying to identify the cause of a patient's diseases.

Earlier hematologists were performing microscopic, examination and counting of blood cells manually, which was very time-consuming and tedious process. Also, the accuracy of counting mainly depends on their expertise skill and their physical conditions. Also, because of cells complex nature, it still remains a challenging task to segment cells from their background and count them automatically.

Our work is to automate the task of cell counting, we will try to find the best solution to count red, white blood cells and platelets. Therefore, we proceed in two steps:

- **The segmentation:** where we need to segment the image and remove the noise to get a clear mask on which we are going preform the counting. In this phase we are comparing between two convolutional neural networks U-Net and SegNet.
- **The counting:** in which we will take the output mask (and edge-mask for red blood cells) from the first phase and apply counting algorithms on it. We used 3 algorithms to count the cells: Watershed, Connected Component Labeling, Circle Hough transform.

We can see in fig 1 an image containing all blood cell types (red, white and platelets) :

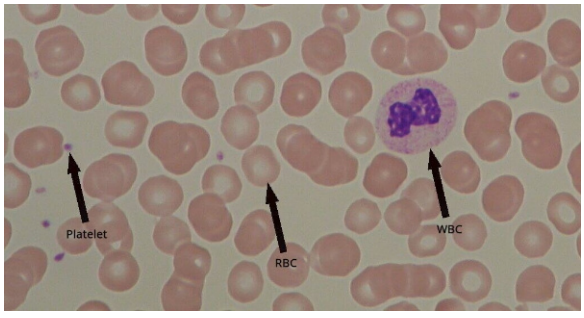

Fig. 1. Red blood cells, white blood cells and platelets

This paper is organized as follows: Related work describes

the research in the area of blood cells segmentation and counting, Data Description describes the dataset used in the proposed approach, Methodology describes the proposed approach. Finally, the last section describes the results of the proposed approach, conclusion and future work to be done.

II. Related Work

Bhavnani et al. [1] have developed a method for segmenting and counting RBC (red blood cells), WBC (White blood cells) and platelets which is also called complete blood count (CBC), by using Otsu's thresholding and morphological operations as a segmentation method, and for counting they are preforming a comparison between two methods: the watershed algorithm and Circular Hough Transform. The CHT method is the best in terms of accuracy with 92.67% but it has some weaknesses with overlapping cells and morphological abnormalities. In the other side the watershed method which is a little bit adapted with overlapping and touching cells achieved an accuracy of 91.07%.

Guiliang, FENG et al. [2] have a developed an algorithm that segments and counts cell images based on image definition, a Discrete Cosine Transform (DCT) is applied, which is proposed by N. Ahmed and Rao in 1974 [3]. Instead of the traditional watershed approach, the DCT method showed better results in comparison. However, there is a drawback to this approach, because this algorithm depends on image definition it relies on well focused images, consequently, when the images are out of focus the segmentation and counting is not reliable. But despite that drawback, it achieved a relatively high accuracy of over 90% which is better that the watershed method.

K. Sudha and P. Geetha [4] have developed a two stage framework which will segment the leukocytes (a type of WBC) with an edge strength-based Grabcut method as a first stage, in the second stage will count the cells using the novel gradient circular hough transform (GCHT) method. In the experiments phase they used ALL-IDB [5] and Cellavision [6] datasets, after resizing the images to 256x256.
After the experiments the proposed method had reached an average segmentation accuracy of 99.32% and a counting accuracy of 97.3%. The new GCHT method can segment touched cells and even overlapped cells.

Kimbahune et al. [7] have developed a method for segmenting and counting red blood cells (RBC) and white blood cells (WBC). segmentation is done using Pulse-Coupled Neural Network (PCNN) and square tracing algorithm for contour tracing after de-noising it with PCNN combined with median filter, the counting is performed by scanning the image and using edge detection methods as square tracing algorithm. this method gave good results compared to state of art methods.

Carlos X. Hernández et al. [8] have used a convolutional neural network (CNN) using a feature pyramid network (FPN) combined with a VGG style neural network for segmenting and counting of cells in a given microscopy image. The dataset they used is BBBC005 [9] from Broad Institute's Bio-image Benchmark Collection. This approach achieved a relatively good accuracy of 81.75% but with some failure cases such as: High cell overlap, Irregular cell shapes and bad focal planes.

Tran, Thanh and Minh et al. [10] have developed a method for segmenting and counting RBC and WBCs by using the SegNet model initialised with weights from a pre-trained VGG-16 model, for the counting they first apply Distance transform with 4 different distance metrics, then they apply binary dilation. At the End, they apply the connected component labeling algorithm to count the number of separated cells in images mask. Their model had a segmentation accuracy of 89% and counting accuracy of 93.3% on RBC and accuracy of 100% on WBC with the testing database which has the cropped images of RBC and WBC.

Yan Kong et al. [11] have developed a two-stage framework using parallel modified U-Nets together with seed guided water-mesh algorithm for automatic segmentation and yeast cells counting which is used to observe the living conditions and survival of yeast cells under experimental conditions. This method achieved a precision of over 99.74% and an average recall rate of 99.35%. however, there is a limitation using this approach, which is the detection of small objects.

Shahzad, Muhammad et al. [12] have developed a custom convolutional encoder-decoder framework along with VGG-16 as the pixel-level feature extraction model to address the problem of whole-slide cell segmentation using the semantic segmentation approach. They used ALL-IDB1 as their baseline dataset and their approach achieved a class-wise accuracy of 97.45%, 93.34%, and 85.11% for RBCs, WBCs, and platelets, respectively, while global and mean accuracy remain 97.18% and 91.96%, respectively.

Overton, Toyah and Tucker, Allan [13] have developed a method which segments and counts IDP (Internally Displaced people) and erythrocytes (red blood cells) using DO-U-Net (Dual Output U-Net) which outputs a segmentation mask and an edge mask then they subtract them to get rid of the overlapping and the touching problem, the model trains on extremely small datasets (10 images) and gives a high segmentation accuracy, They selected 10 images of 108 from ALL-IDB dataset for training the model, the model takes images with a resolution of 188x188 and outputs a segmentation mask and edge mask of lower resolution 100x100, the experiments results have given an accuracy of 98.31% on a 5 randomly selected images from ALL-IDB, for the IDP they had 98.69% for fixed resolution images and

94.66% for scale-invariant satellite images.

Li, Dongming et al. [14] have developed a method for segmenting blood cells by combining neural ordinary differential equations (NODEs) with U-Net networks to improve the accuracy of image segmentation. In order to study the effect of ODE-solve on the speed and accuracy of the network, the ODE-block module was added to the nine convolutional layers in the U-Net network. Using this approach to segment blood cell images in the testing set, it has achieved 95.3% pixel accuracy and 90.61% mean intersection over union.

In this section after we studied these articles we can see that in the segmentation part the deep learning models are giving more accurate results like in articles [13], [10], [11], [8], so we've chosen the two models SegNet and UNet for the segmentation part of our method. For the counting side we took the most used methods that we saw in the studied articles [1], [10], [4] to compare between them.

III. Methodology

In our work, we focus more on the segmentation task because it directly affects the counting accuracy. where the accuracy of the output mask and edge plays a big role in the detection of overlapping, so our main goal is to segment the overlapped cells for example in red blood cells we added an edge mask then by combining it with the mask we remove 80% of overlapping. after the segmentation we will apply the counting algorithms (Watershed, Circle hough transform, Connected component labeling) which needs some parameter tuning in our case we did a manual tuning. we will devide this section on two: Segmentation phase and counting phase:

A. Cell semantic segmentation using deep leaning

U-Net and SegNet architectures are the most used in cell segmentation tasks [10], [11], [13], [14] (blood cell segmentation in particular).
In the following sections, we will briefly analyze and compare both convolutional neural network (CNN) U-Net and SegNet models with their perspective results. And explain all the post-processing methods we used for the counting of blood cells (red, white and platelets).

1) ***DO-U-Net****:* The network is based on the fully convolutional network and its architecture was modified and extended to work with fewer training images and to yield more precise segmentations. In our case we are using DO-UNet first proposed in [13] which is a modified U-Net to produce dual outputs, which is also known as a contour aware network, first demonstrated by the DCAN architecture [15]. Based on a simple FCN, DCAN was trained to use the outer contours of the areas of interest to guide the training of the segmentation masks. This led to improved semantic and instance segmentation of the model, which in their case, looked at non-overlapping features in biomedical imaging. With the aim of counting closely co-located and overlapping cells, we are predominantly interested in the correct detection of individual objects as opposed to the exact precision of the segmentation mask itself. An examination of the hidden convolutional layers of the classical U-Net showed that the penultimate layer of the network extracts information about the edges of the cells, so the idea is to output the cell mask + edge mask then do a subtraction to break the overlapping cells.

They Started with the classical U-Net then reduced the number of convolutional layers and skip connections in the model. Simultaneously, they minimised the complexity of the model by looking at smaller input regions of the images, thus minimising the memory footprint of the model. They follow the approach of Ronneberger et al. [13] by using unpadded convolutions throughout the network, resulting in a model with smaller output edge and mask (100×100 px) corresponding to a central region of a larger (188×188 px) input image region. DO-U-Net uses two, independently trained, output layers of identical size. Figure 2 shows the DO-U-Net architecture.

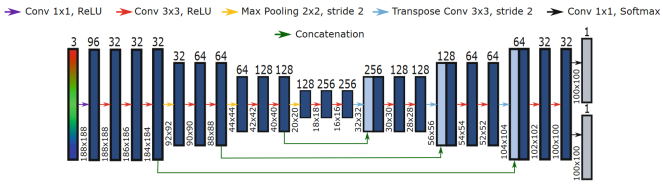

Fig. 2. DO-UNet architecture

2) ***SegNet****:* The SegNet neural network, developed by Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla, all from the University of Cambridge, is a convolutional neural network used for semantic pixel wise labeling. This problem is more commonly called semantic segmentation [16].

SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. This architecture is illustrated in the figure below. The model we used has the 13 encoder layers obtained from the VGG16 network, and 13 decoder layers to match the same number of encoder layers. The final decoder output is fed to a multi-class soft-max classifier to produce class probabilities for each pixel independently (pixelwise).

Each encoder in the encoder network performs convolution with a filter bank to produce a set of feature maps. These are then batch normalized. Then an element-wise rectified- linear non-linearity (ReLU) max(0, x) is applied. Following that, max-pooling with a 2x2 window and stride 2 (non-overlapping window) is performed and the resulting output is sub-sampled by a factor of 2. Max-pooling is used to achieve translation invariance over small spatial shifts in the input image.

The appropriate decoder in the decoder network upsamples its input feature map(s) using the memorized max-pooling indices from the corresponding encoder feature map(s). This step produces sparse feature map(s). This SegNet decoding technique is illustrated in the below figure. These feature maps are then convolved with a trainable decoder filter bank to produce dense feature maps. A batch normalization step is

then applied to each of these maps. Note that the decoder corresponding to the first encoder (closest to the input image) produces a multi-channel feature map, although its encoder input has 3 channels (RGB). This is unlike the other decoders in the network which produces feature maps with the same number of size and channels as their encoder inputs. The high dimensional feature representation at the output of the final decoder is fed to a trainable soft-max classifier. [16]

The input shape we used is (128x128x3) and the output shape is (128x128), there is no loss in resolution because SegNet uses the option 'same' padding on each encoder layer.

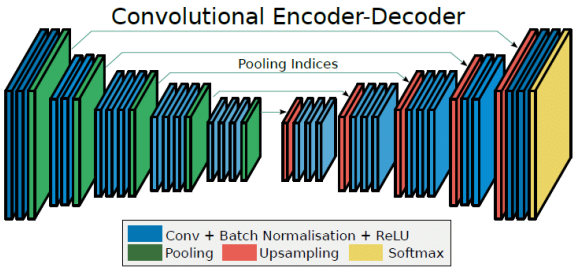

Fig. 3. SegNet architecture

B. Cell counting

After having segmented blood cell images masks for red, white blood cells and platelets, we use multiple post-processing methods to get the number of cells. as we can see in 7 and 8. We can see below all the algorithms we used to get a relatively accurate blood cell count:

1) Circle Hough Transform: Circle Hough Transform (CHT) is machine learning algorithm used to extract features (circles) from imperfect images. We modified its parameters (Minimum distance, Minimum and Maximum radius...) for each type of blood cells (red, white and platelets).

Note that we do not rely on this approach to count white blood cells, because most white blood cells have different shapes. Therefore, this method is useless when it comes to white blood cells counting.

We Modified this method by adding a loss function which will help us to eliminate False Positives circles by calculating the percentage of the intersection between the circle and the cell mask. we improved the counting accuracy by more than 20% with a threshold intersection percentage of 60%.

As we can see in fig 4 the steps of the counting with the CHT method:

- we first take the mask/edge from the model.
- we apply a threshold on the mask to binarize it.
- we apply our surface filter algorithm to filter object that are not in the size range of the cell that we are counting.
- apply the circle hough transform to detect the circles in the cleaned image.
- feed the binary mask and the obtained circles from CHT to calculate loss of each circle (percentage of intersection).
- return the final number of circles that meet the threshold condition which is the circle count.

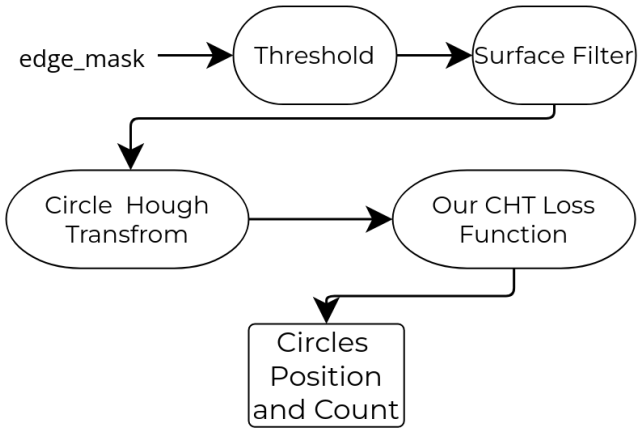

Fig. 4. CHT schema applied to count blood cells

2) Connected Component Labeling: Connected Component Labeling (CCL) is machine learning algorithm used to detect connected regions in a binary image. Before applying the connected component labeling, we convert the images to gray-scale. Then, a binary threshold is applied to the images to get binary values. Finally, we apply the connected component labeling to get the labels and map component labels to the resulting image, and the number of labels is the cell count accordingly.

3) Watershed: Watershed algorithms (also called drainage divide) are used in image processing primarily for object segmentation purposes, that is, for separating different objects in an image. the main purpose of using watershed in this phase is to segment the touching and overlapping cells, the watershed takes two inputs, first i takes an image with different intensity levels in our-case the distance transform of our mask where the intensity levels represents reliefs. the second input is the water sources in our-case we extracted local maxima from the distance transform image. we can see below the steps we used to count the cells.

1) **Compute the Euclidean distance**: we compute euclidean distance from every binary pixel to the nearest zero pixel, this map will be used as our relief map in the watershed algorithm.
2) **We find peaks in the distance map**: we search for peaks in our euclidean distance map which is the local maxima in each region, which are the highest points in the map (higher intensity levels), which we will use as water sources in the watershed algorithm.
3) **Apply connected component labeling on the peak map**: we apply CCL algorithm which is also called 8-connectivity algorithm to label the peaks (label each water source).
4) **Apply the Watershed algorithm on the reversed distance map using the labeled peaks**: at the end we feed the reversed distance map and the water sources map (local maxima) to the watershed algorithm to get the segmented image.

Here is a schema presenting the previously mentioned steps:

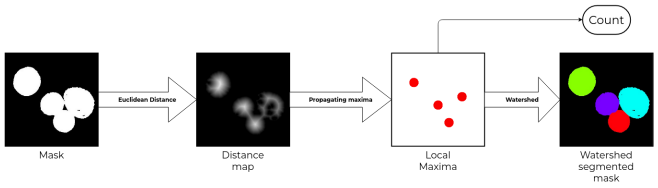

Fig. 5. Watershed schema applied to count white blood cells

IV. Experimentations and results

A. Dataset used in this work

1) Description: Dataset ALL-IDB1 is a version of ALL-IDB dataset, it is composed of 108 images (see figure 6) collected during September, 2005. It contains about 39000 blood elements, where the lymphocytes have been labeled by expert oncologists [5]. All images are in JPG format with 24 bit color depth, and a native resolution equal to 2592 × 1944, captured with a PowerShot G5 camera. The images are related to different magnifications of the microscope (ranging from 300 to 500). It can be freely downloaded from [17].

The ALL-IDB1 can be used for segmentation or classification with image processing methods or artificial intelligence models.

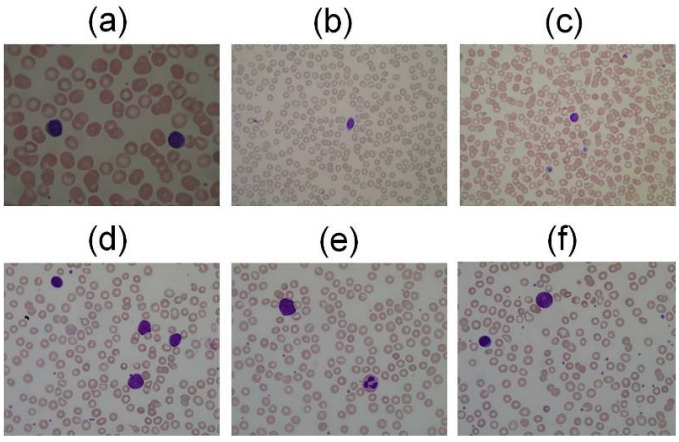

Fig. 6. Examples of the images contained in ALL-IDB1: healthy cells from non-ALL patients (a,b,c), probable lymphoblasts from ALL patients (d,e,f).

For both models, we decided to work with the updated ALL-IDB1 dataset which has 13 edge-masks and 108 masks of Red Blood Cells (RBCs), 108 masks of White Blood Cells (WBCs) and Platelets. For the count information we have 13 RBC images which have the count information, but for WBCs and Platelets we had to use manual count and algorithms to find the count information because they didn't provide the count information. For the WBC we used manual count, for the Platelets we used the connected component labeling on the ground truth mask.
10 images with their perspective masks and edge masks were chosen for red blood cell training and 3 as a test dataset, for white blood cells we used 73 images with their masks as a train dataset, and 33 as a test dataset. For platelets, we used 71 for training and 31 as a test dataset. Only red blood cells have edge masks, because we need to get rid of overlapped cells, white blood cells and platelets don't need the edge masks, we can retrieve all the necessary features using masks only, because both white blood cells and platelets rarely overlap. The images will be sliced to the input size of the according model to match the input shape of the models, we also add a padding when we slice to recover the edges loss in convolution. The resulting train dataset will be 3916 image, mask, and edge tiles (a total of 11748 tiles) see table I. For the test dataset 1072 image, mask, and edge tiles (a total of 3216 tiles) for red blood cells. As for white blood cells, 28126 image and mask tiles were used for training (a total of 56252 tiles), and 15892 image and mask tiles were used as a test dataset for white blood cells (a total of 31784 tiles). Finally, for platelets we used 27650 image and mask tiles were used for training (a total of 55300), and 14410 image and mask tiles as a test dataset for platelets (a total of 28820).
Here is a table which contains the resulting sliced images for red blood cells, white blood cells and platelets:

Dataset		Train images	Test images	Train Tiles	Test Tiles	Total images	Total tiles
Red Blood Cells	**Image**	10	3	3916	1072	13	**4988**
	Mask	10	3	3916	1072	13	**4988**
	Edge	10	3	3916	1072	13	**4988**
White Blood Cells	**Image**	73	33	28126	15892	106	**44018**
	Mask	73	33	28126	15892	106	**44018**
Platelets	**Image**	71	31	27650	14410	102	**42060**
	Mask	71	31	27650	14410	102	**42060**

TABLE I
DATASET USED FOR BOTH MODELS

2) Data augmentation: We used the same dataset augmentation on all cells (red, white blood cells, and platelets) in both models UNet and SegNet. The augmentation we used was custom which involves the following steps:

1) Pick a random image from the train dataset.
2) Get the x and y coordinates randomly from the chosen image.
3) Rescale the image randomly to a smaller size then scale it back to the original size to reduce quality.
4) Take a slice of the image and mask accordingly and also edge if available.
5) Skip the image if it doesn't contain out object of interest
6) Resize the image and mask to the model input.
7) Randomly rotate and flip the image chip.
8) Randomly augment the colors (luminosity and saturation).

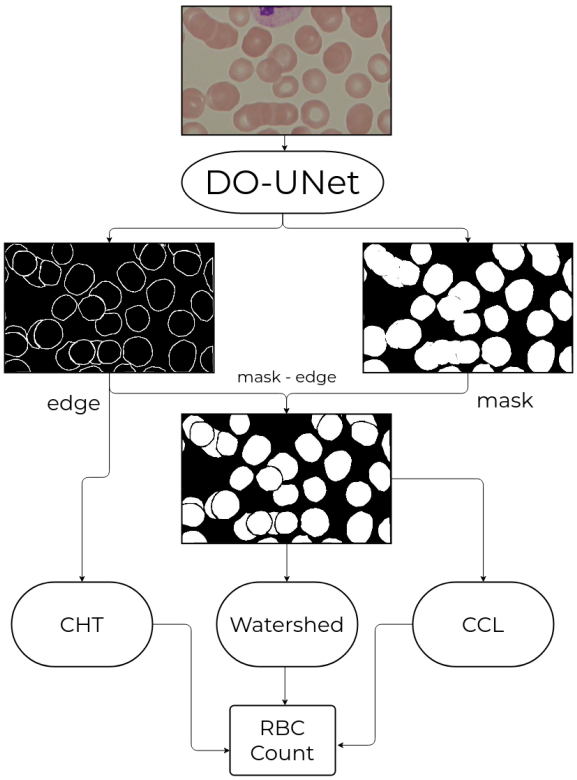

Fig. 7. Schema of the segmentation and counting steps of the RBC's

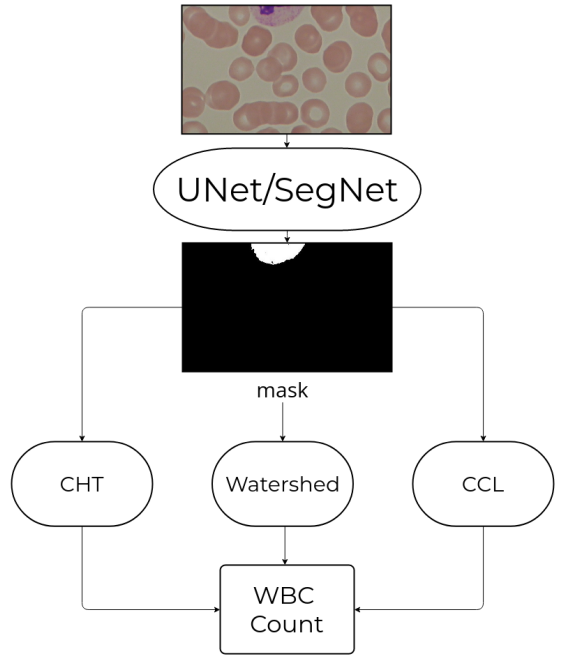

Fig. 8. Schema of the segmentation and counting steps of the WBC's and Platelets

B. Metrics

- **Accuracy** Pixel accuracy It is the percent of pixels in the input image that are classified correctly as background or foreground.
- **IOU** The Jaccard Index or Intersection Over Union [18] citejaccard1912distribution

$$J(A,B) = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

- **Dice** aka. The Sørensen–Dice coefficient [19]:

$$DSC = \frac{2TP}{2TP + FN + FP} \quad (2)$$

- **Tversky** The Tversky index, named after Amos Tversky [20]:

$$S\_LOSS(X,Y) = 1 - \frac{|X \cap Y|}{|X \cap Y| + \alpha|XY| + \beta|YX|} \quad (3)$$

C. Results Discussion

References	RBC		WBC		Platelets	
	Segmentation	Counting	Segmentation	Counting	Segmentation	Counting
Guiliang, FENG et al. (2016)	90%					
Bhavnani et al. (2016)	CHT- 92.67% Watershed- 91.07%				N/A	
Carlos et al. (2018)	95%					
Tran, Thanh and Minh et al.(2019)	93%	93.30%	93%	N/A	N/A	
K. Sudha and P. Geetha (2020)	N/A		99.32%	97.3%	N/A	
Yan Kong et al. (2020)	96%					
Shahzad et al. (2020)	97.45%	N/A	93.34%	N/A	85.11%	N/A
Overton, Toyah and Tucker, Allan (2020)	98.31%	N/A				
Li, Dongming et al. (2021)	N/A		95.3%	N/A		
Our Method	97.1%	95.36%	99.76%	97.94%	99.46%	98.58%

TABLE II
COMPARATIVE TABLE FOR SEGMENTATION AND COUNTING ACCURACY'S BETWEEN STUDIED METHODS

As we can see in table II that most of the methods didn't focus on the counting side, the stop at the segmentation stage. In our method we did a full benchmark on the 3 blood components with 2 segmentation models and 3 counting algorithms. For the segmentation we can see in the outputs that the SegNet gives better edge masks with sharp edges that helps removing the overlapping by subtraction. but i generates a lot of noise compared to U-Net which has no noise at all.

We have also a problem that affects both models which is the color space, if we change the colors a little bit in the input images we will get bad results because the models depends a lot on the RGB layers.

In the counting phase: we had a problem the CHT algorithm where we was getting bad circles which doesn't cover any cell, we had to create a circle loss function to solve this problem with CHT. in the watershed algorithm we get a lot of over-segmentation cases because of the shape and transparency of the WBCs. for the 3 counting algorithms we had a parameter tuning problem, in our case we tuned the parameters manually.

D. DO-U-Net Results

1) Red Blood Cells: They are the most difficult to detect because of their nature of overlapping, where in some samples we can't see the overlapping by the naked eye, in this experiment we are testing the DO-U-Net from [13]. For the DO-U-Net we updated the data augmentation phase, and applied Transfer Learning to get a better edge-mask. we can see in the dataset that we only have 13 images out of 108 from ALL-IDB1 that contains edge-masks and 108 masks. Consequently, there is a lack of the edge-mask labels. We ended up with the method below as the best fit to our problem:

1) Train the DO-U-Net outputs on the large dataset (108 masks) which will output two identical masks.
2) Continue Training the DO-U-Net with the small dataset (13 masks, 13 edges), and Freeze the Mask Output.

The table III compares between the normal trained model A on the small dataset which contains 13 masks and edge-masks and the model B which is trained on two phases, first using the large dataset (108 masks without edge-masks) for 60 epochs, and in the second phase we continued training using the small dataset (13 masks + edges) for 400 epochs. We can see that this method pushed the edge accuracy higher, which is really important to get rid of the overlapping.

2) White Blood Cells: The White Blood Cells are also difficult to detect because of the non stable shape and in some cases they overlap, in this experiment we are comparing single output U-Net from [13] and the single output SegNet model. In the U-Net we removed the edge output because we don't have the edge annotation. then trained the model for 15 epochs with the binaryCrossEntropy Loss function on (74 + 34) images. We ended up with a very high accuracy and IOU score as we can see in table **??**.

3) platelets: The platelets are easy to count because of the rare overlapping but they are a bit difficult to segment because of their small size.

In this experiment we are testing single output U-Net from [13]. In the U-Net we removed the edge output because we don't have the edge annotation. then trained the model for 50 epochs with the BinaryCrossEntropy Loss function on (74 + 34) images. We ended up with a very high accuracy and IOU score as we can see in table **??**.

E. SegNet Results

SegNet segmentation results were pretty accurate for white blood cells and platelets, as for red blood cells, the segmentation was done using dual output (mask and edge-mask) to get rid of overlapped cells.

Here are the results of the Mean Squared Error (MSE) loss function on each type of cell:

1) Red Blood Cells: For the dual-output SegNet model, the resulting segmented images were very good, sometimes better than the do-U-Net, though it is not as optimized when training and also predicting images, but it gets the job done with 95.86% mask and 93.99% edge accuracies. The segmented output images also had some noise which affected Connected Component Labeling (CCL) when counting. As for Circle Hough Transform (CHT), the noise did not affect the result. Red Blood Cells detection and counting is by far the hardest, because it is the only cell that overlaps and that makes it hard for counting. The segmented output of do-SegNet is thresholded using a binary threshold, and then sent to 3 algorithms:

- **Circle Hough Transform**: CHT was our best result for red blood cells counting, which achieved an accuracy of 94.03% on the same dataset used for training the model (13 images with their respective masks and edge-masks).
- **Connected Component Labeling**: CCL was applied directly on the thresholded output edge, this method was far from accurate because the do-SegNet output had some noise (even when removing most of it), and also the overlapped nature of red blood cells which makes it very hard for this algorithm to count correctly. CCL achieved an accuracy of 76.49% counting red blood cells.
- **Euclidean Distance Transform**: EDT is used to get rid of the overlapped cells, also peak local max was applied on the EDT output for finding local maxima(s), the result of this approach is 84.64% accuracy.

2) White Blood Cells: The results of white blood cells segmentation and counting using the SegNet model was very accurate achieving 99.72% when segmenting. White blood cells are the easiest out of the three, and the most accurate results. However, white blood cells are different from the other cells because they come in different shapes and sizes, which made it hard to adapt each counting algorithm to every cell. The same counting methods are applied CHT, CCL and EDT. And each method had some drawbacks.

Here are the results:

- **Circle Hough Transform**: Due to the different shapes of white blood cells, CHT achieved the lowest result which is 79.9% counting accuracy, because some of the cells don't even look like circles and also their different size which made it harder to count.
- **Connected Component Labeling**: CCL is similar to CHT when it comes to white blood cells. And, because of the noisy outputs of SegNet the binary threshold can only do so much (it thresholds some of the noise generated when predicting).
  CCl achieved an average counting accuracy of 82.89%.
- **Euclidean Distance Transform**: EDT is the contender of white blood cells counting, because the distance transform gets rid of the noise completely and peak local max was very helpful in eliminating that noise and getting an accurate count.
  This method achieved a counting accuracy of 96.43%.

3) Platelets: The platelets segmentation result we achieved is not the best compared to the do-U-Net model. SegNet extracts the platelets but with some noise which made it very hard to count accurately. It achieved a segmentation accuracy of 99.89% and the highest counting accuracy is

RBC_Model	Dataset	Epochs	Output	Loss	Mean IOU	Dice	Tversky	Accuracy
A	small Dataset (mask + edge) * (10 + 3)	800	Mask	0.3615	0.6365	0.8304	0.8232	0.8754
			Edge	0.1816	0.0663	0.3582	0.3475	0.9343
B	Phase 1: large dataset mask*108 Phase 2: small Dataset (mask + edge)*13	Phase 1: 120 Phase 2: 400	Mask	0.0713	0.7751	0.9528	0.9567	0.9716
			Edge	0.1465	0.0759	0.4127	0.4015	0.9385

TABLE III
NORMAL TRAINED MODES COMPARED TO TRANSFER LEARNING MODEL

71.56% which is not very good.
Here are all the counting accuracies for each approach:

V. CONCLUSION

In the present work, we have mainly presented two different models of segmentation based on CNNs and three counting algorithms to perform a complete blood count. We focused more on the segmentation task where we tested two models U-Net and SegNet to get better masks.

We can see that the U-Net gave better result with less noisy mask, but the two models has some weaknesses with the images color space, because both models takes rgb images therefore they depend a lot on the color features.
e.g. if we slightly change the colors of the input image we get a big difference in the output mask. In the counting task, we had a small time window where we couldn't tune the 3 three algorithm parameters to get the best results. but we got acceptable results in each blood cell type, especially with platelets.

As a future work, we can combine between the two models to benefit the strength points of both models, we can also find more combinations between edge and mask. We can Also create a model that predicts the algorithm parameters for watershed and ccl which in our case are tuned manually.

Blood Cells / Model	Output	DO-U-Net				SegNet			
		Segmetation Accuracy %	Counting Accuracy %			Segmetation Accuracy %	Counting Accuracy %		
			CHT	CCL	Watershed		CHT	CCL	Watershed
Red Blood Cells	**Mask**	**97.16**	**95.36**	78.66	90.43	95.86	94.03	76.49	84.64
	Edge	93.85				**93.99**			
White Blood Cells	**Mask**	**99.76**	88.7	58.74	**97.94**	99.72	79.9	82.89	96.43
Platelets	**Mask**	99.46	X	**98.58**	X	**99.89**	65.65	71.56	65.65

TABLE IV
COMPARISON OF DO-U-NET AND SEGNET RESULTS

REFERENCES

[1] L. A. Bhavnani, U. K. Jaliya, and M. J. Joshi, "Segmentation and counting of wbcs and rbcs from microscopic blood sample images," *International Journal of Image, Graphics and Signal Processing*, vol. 8, no. 11, p. 32, 2016.

[2] F. Guiliang, L. Yiping, and P. Wei, "Microscopic cell image segmentation and counting algorithm based on image definition." *International Journal of Simulation–Systems, Science & Technology*, vol. 17, no. 38, 2016.

[3] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[4] K. Sudha and P. Geetha, "A novel approach for segmentation and counting of overlapped leukocytes in microscopic blood images," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 2, pp. 639–648, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0208521620300267

[5] R. D. Labati, V. Piuri, and F. Scotti, "All-idb: The acute lymphoblastic leukemia image database for image processing," in *2011 18th IEEE international conference on image processing*. IEEE, 2011, pp. 2045–2048.

[6] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968432817303037

[7] V. V. Kimbahune and N. Uke, "Blood cell image segmentation and counting," *Inter J Engineer Sci Tech*, vol. 3, no. 3, pp. 2448–2453, 2011.

[8] C. X. Hernández, M. M. Sultan, and V. S. Pande, "Using deep learning for segmentation and counting within microscopy data," *CoRR*, vol. abs/1802.10548, 2018. [Online]. Available: http://arxiv.org/abs/1802.10548

[9] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation." *Nature methods*, vol. 9, no. 7, pp. 637–637, 2012.

[10] T. Tran, L. B. Minh, S.-H. Lee, and K.-R. Kwon, "Blood cell count using deep learning semantic segmentation," 2019.

[11] Y. Kong, H. Li, Y. Ren, G. Z. Genchev, X. Wang, H. Zhao, Z. Xie, and H. Lu, "Automated yeast cells segmentation and counting using a parallel u-net based two-stage framework," *OSA Continuum*, vol. 3, no. 4, pp. 982–992, Apr 2020. [Online]. Available: http://opg.optica.org/osac/abstract.cfm?URI=osac-3-4-982

[12] M. Shahzad, A. I. Umar, M. A. Khan, S. H. Shirazi, Z. Khan, and W. Yousaf, "Robust method for semantic segmentation of whole-slide blood cell microscopic images," *Computational and Mathematical Methods in Medicine*, vol. 2020, 2020.

[13] T. Overton and A. Tucker, "Do-u-net for segmentation and counting," in *Advances in Intelligent Data Analysis XVIII*, M. R. Berthold, A. Feelders, and G. Krempl, Eds. Cham: Springer International Publishing, 2020, pp. 391–403.

[14] D. Li, P. Tang, R. Zhang, C. Sun, Y. Li, J. Qian, Y. Liang, J. Yang, and L. Zhang, "Robust blood cell image segmentation method based on neural ordinary differential equations," *Computational and Mathematical Methods in Medicine*, vol. 2021, 2021.

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

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[17] "All-idb web site, r. donida labati, v.piuri, and f.scotti, universitàdeglistudi di milano, departement of informationtechnology,http://www.dti.unimi.it/fscotti/all."

[18] A. H. Murphy, "The finley affair: A signal event in the history of forecast verification," *Weather and forecasting*, vol. 11, no. 1, pp. 3–20, 1996.

[19] T. A. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Biol. Skar.*, vol. 5, pp. 1–34, 1948.

[20] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, 1977.