

# Swift Medical Report Analysis using Computer Vision (SMRA-CV)

Hitesh Hinduja<sup>1</sup>, Rajeev Singh<sup>2</sup>, Gaurav Gaonkar<sup>3</sup>

<sup>1</sup>Artificial Intelligence Researcher, Bangalore, India

<sup>2</sup>BE Student, Department of Computer Engineering, Vidyalkar Institute of Technology, Maharashtra, India

<sup>3</sup>BE Student, Department of Computer Engineering, Vidyalkar Institute of Technology, Maharashtra, India

\*\*\*

**Abstract** - The digitization of documents and their availability over the network demands solutions toward content-based document image analysis, indexing, searching, and retrieval. In today's busy world, tracking the medical reports and handling them becomes very difficult for a person. Using document image analysis will help extract important information from a report in seconds. This research aims to extract the test results from the CBC reports, storing the data of every patient separately and then plotting a graph of their test results. We used different Optical Character Recognition (OCR) engines for the text extraction process by using suitable Image Pre-processing techniques on pixelated reports. After text extraction, we implemented advanced regex patterns for detecting the key metrics along with their result and then stored the result in a CSV file and used the data stored in the file to plot the timeline graph for the patient to help him keep track of his health. Our proposed algorithm takes 2.1 seconds for extracting and storing important data from each report with an accuracy of 95%. This research will play a major role in tracking health in daily life. Hospitals can easily access the patient's medical records in the best possible manner. Patients can easily track the reports graphically and this will make the understanding of reports easy.

**KeyWords** - Optical Character Recognition, EasyOCR, Regular Expression, Tesseract, Binarization, Image Preprocessing

## 1. INTRODUCTION

Understanding a blood report's results for the users has always been a challenge. The actual metrics and range of every metric mentioned in the reports have been identified as ambiguous unless explicitly mentioned in the reports the range of severity of every metric. Majorly, it has been a challenge for the patients to observe their trend of every blood-report metric such as Haemoglobin, R.B.C count, PCV, and others. Especially when it is a continued treatment of performing blood tests every two weeks, it becomes difficult to keep a track of every metric and the improvement in every metric with time. The whole aspect of implementing this technology was to help the users understand their report metrics in visual form and provide

smart recommendations visually. This includes plotting graphs of every metric by automated sorting of reports' basis names and date filter, marking the points on the graph for metrics as critical, normal, or less by taking the expected range of the metrics from the reports. The technology also considers the worst value of every metric which can be reached that is beyond the range of the metric mentioned in the blood reports. This study answers the following questions through visual graphs.

- What is the trend of every metric mentioned in the blood report?
- What are the highs and lows of every metric and how have they changed (improved/deteriorated) with time?

These are the two major questions that are answered through the technology we have built. For this study, we had collected around 120 blood reports where 85 reports (70%) were used in improving our technology and as input to our techniques, and the rest 35(30%) reports were purely considered as test reports with accuracies benchmarked on those reports.

## 1.1 RELATED WORK

A lot of methods and techniques have evolved for document image analysis and retrieval. There has been a lot of research related to the retrieval of data from document images. In 2000, Liu et al. [1], presented an approach to image-based form document retrieval. They developed a prototype form retrieval system using similarity measures for forms that are insensitive to translation, scaling, moderate skew, and image quality fluctuations. Zhu et. al [2], proposed automatic document logo detection and extraction in document images that use boosting strategy across multiple image scales for classifying and localizing the logo. In 2011, Dan Claudiu Ciresan et. al [3] discovered that using simple training data pre-processing gave lesser error than the previous models that used different nets trained on the same data. In 2010, Kokare and Shirdhonkar presented a comprehensive survey on document image retrieval systems that provides an insight into state of art research activities from 1992 to 2009. Badawy et al.[4] in 2012 and

Andrew S. Agbemenu et al.[5] in 2018 discussed the Automatic License Plate Recognition. It refers to extracting the information on the vehicle license plate from an image/group of images. In 2016 Wang et al. [6] presented a fast text detection algorithm for scanned document images. They used low pass filters in their technique to remove high-frequency noise, morphological operations, and edge detection schemes. The Enhancing is done by clipping the L and C channels of the LCH Space. Khan and Puri [7] presented a study of text detection techniques for printed documents. In this paper, technical analysis on textual data identification and extraction from images is given in which various approaches for text detection and extraction are discussed and an evaluation is done on the popularity of the approach. In 2007, H.S. Ackley[8] has discussed various methods for Optical Character Recognition. In 2012, Shrey Dutta et al.[9] used character n-grams in their word recognition approach on degraded Indian language document images with a 15% decrease in error. OCRs give a poor performance for complex documents such as Indian languages. In 2012, Chirag Patel et al.[10] did a comparative study between different OCR tools Transym OCR by taking vehicle number plate as the input. In 2019, Remus Petresuca et al.[11] reviewed the existing papers and proposed two well-known OCR engines and a voting principle based on weights. Aggarwal V. [12] in 2017 investigated the use of image processing

techniques and machine learning algorithms to extract editable text from scanned images. Harraj et al. [13] propose a four-step algorithm to improve Tesseract 3.02's accuracy. The article majorly focuses on using different image pre-processing methods to the images such that the OCR engine receives clearer images to extract. The techniques involved in the article are using brightness and contrast management, grayscale conversion, an unsharp masking approach, and Otsu's binarization method. The authors proved that the image pre-processing techniques can increase the accuracy of the OCR. The accuracy achieved by Harraj et al. [13] was 83.97% which is 06.80% more than the previously achieved accuracy.

## 2. Design

For this study, we have used 120 blood sample reports of several users as the data source. On average, nearly 5-6 reports per user were used. This study is done in three parts. In part one, we used Optical Character Recognition (OCR) techniques to get a clean text output across all the reports. In part two, the advanced regular-expressions approach was used to analyse the patterns in the blood reports. In part three, a smart sorting technique was written to sort the reports of the users based on their names and dates on the reports, thus plotting visual representations of blood report metrics for every user.

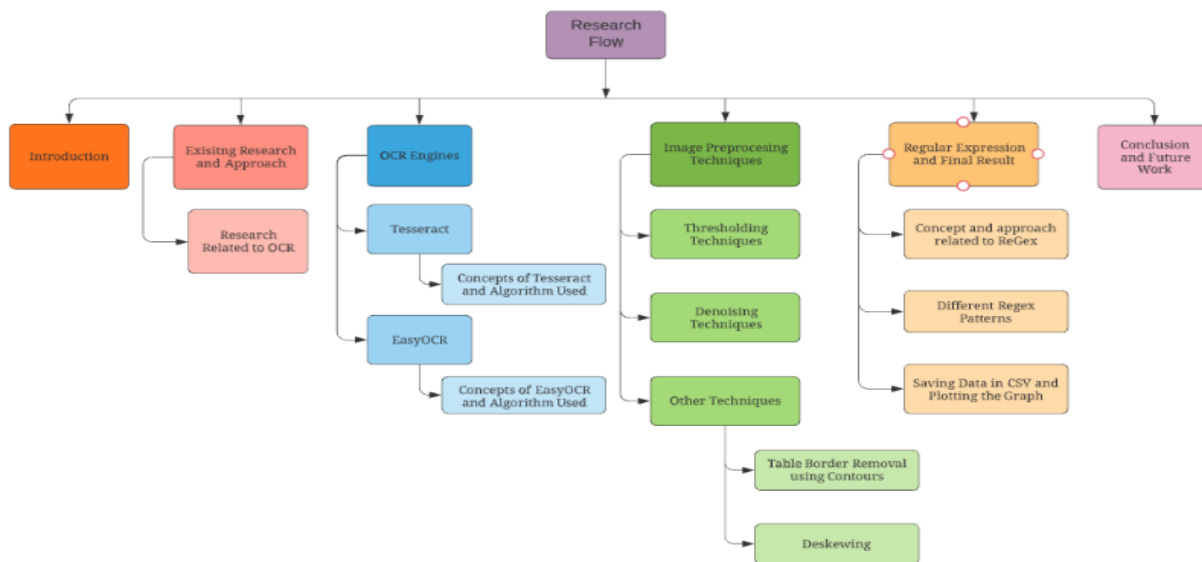


Fig. 1 This figure describes the proposed workflow of the system along with the Image processing techniques mentioned above

### 2.1 Text Extraction Using OCR Engines

Different Optical Character Recognition(OCR) engines with their complex algorithms have achieved huge success in

document image analysis. These OCR engines take the pixelated image of the document and return the text present in the document. Seeing the past performance we

chose to use tesseract SMITH[14] as our OCR engine for extracting text from the medical reports. Tesseract assumes the input images will be in Binarized format. Image preprocessing becomes the most important part when we use tesseract. To get the best output from the tesseract, we used several image-preprocessing techniques that include denoising techniques, image thresholding techniques like Otsu, Sauvola, Niblack, Isauvola for binarizing the image. After testing tesseract with suitable preprocessing on a significant amount of reports we found that tesseract doesn't perform well on images in which text is enclosed in tables as it alters the format of the reports and that disorients the entire text. To encounter this, we removed the borders of the table. This reduced the formatting error tesseract was maintaining the format of the report but still it had a problem it was not extracting numbers accurately. The following figure shows improvement of format after removing table edges.

Name : Rajesh Hinduja	Age/Gender: 51 y / M	Collected: 08/09/2020 01:31
Patient ID : 0010235933	Visit Type :IP	Received : 08/09/2020 09:27
	DOB :20/06/1969	Reported: 08/09/2020 12:01
Accession : 2001023647	Location :1S0613`1SR6181`1SB6181`1PULMMED	

DEPARTMENT OF LABORATORY MEDICINE - HAEMATOLOGY

Test Name	Unit	Reference Range	Low	Normal	High
<b>Complete Blood Count</b>					
Red cell count (EDTA Whole Blood)	x10 <sup>12</sup> /L	4.5-5.5	4.23		
Haemoglobin (EDTA Whole Blood)	g/dL	13.0-17.0	11.9		
Haematocrit (EDTA Whole Blood)	%	40.0-50.0	36.8		
MCV (EDTA Whole Blood)	fl	83-101		87.0	
MCH (EDTA Whole Blood)	pg	27.0-32.0		28.1	
MCHC (EDTA Whole Blood)	g/dL	31.5-34.5		32.3	
RDW (EDTA Whole Blood)	%	11.6-14.0		12.1	
Total Leukocyte Count (EDTA Whole Blood(automated))	x10 <sup>9</sup> /L	4-10			10.75
Neutrophils (EDTA Whole Blood)	%	40-80		72.9	
Lymphocytes (EDTA Whole Blood)	%	20-40	18.0		
Monocytes (EDTA Whole Blood)	%	2-10		8.8	

Fig 2 indicates the sample laboratory report of the patient. It describes the blood report metrics that includes various tests performed in the laboratory. For privacy concerns we have hidden the patient and doctor information.

```
Name __: Rajesh Hinduja Age/Gender: 51 y /M Collected: 08/09/2020 01:31
Patient ID: 0010235933 Visit Type :IP Received : 08/09/2020 09:27
DOB 20/06/1969 Reported : 08/09/2020 12:01
Accession : 2001023647 Location :180613`1SR6181`1SB6181`1PULMMED
( DEPARTMENT OF LABORATORY MEDICINE - HAEMATOLOGY ]

Test Name Unit Reference Range Low | Normal High

Complete Blood Count

Red cell count x10^12/L 45-5.5 423

(EDTA Whole Blot - :

Bacmogiobin g/dL. 13.0:17.0 9

Hi tocrit

(ad Ta Wrote loot! % 40.0-50.0 36.8

MCV

(EDTA Whole Blo)! fl 83-101 87.0

MCH

(EDTA Whole Bloo)! pg 27.0-32.0 28.1

MCHC

(EDTA Whole Bloo)! g/dL 31.5-34.5 32.3

RDW

(EDTA Whole Bloo)t % 1614.0 121
```

Fig 3 indicate the text extracted from the sample report of the patient mentioned in the figure 2. The output shown is from one of the OCR Engine.

Name : Rajesh Hinduja	Age/Gender: 51 y / M	Collected: 08/09/2020 01:31
Patient ID : 0010235933	Visit Type :IP	Received : 08/09/2020 09:27
Referred By: [REDACTED]	DOB :20/06/1969	Reported: 08/09/2020 12:01
Accession : 2001023647	Location :1S0613`1SR6181`1SB6181`1PULMMED	

DEPARTMENT OF LABORATORY MEDICINE - HAEMATOLOGY

Test Name	Unit	Reference Range	Low	Normal	High
<b>Complete Blood Count</b>					
Red cell count (EDTA Whole Blood)	x10 <sup>12</sup> /L	4.5-5.5	4.23		
Haemoglobin (EDTA Whole Blood)	g/dL	13.0-17.0	11.9		
Haematocrit (EDTA Whole Blood)	%	40.0-50.0	36.8		
MCV (EDTA Whole Blood)	fl	83-101		87.0	
MCH (EDTA Whole Blood)	pg	27.0-32.0		28.1	
MCHC (EDTA Whole Blood)	g/dL	31.5-34.5		32.3	
RDW (EDTA Whole Blood)	%	11.6-14.0		12.1	
Total Leukocyte Count (EDTA Whole Blood(automated))	x10 <sup>9</sup> /L	4-10			10.75
Neutrophils (EDTA Whole Blood)	%	40-80		72.9	
Lymphocytes (EDTA Whole Blood)	%	20-40	18.0		
Monocytes (EDTA Whole Blood)	%	2-10		8.8	

Fig 4 indicates the laboratory report without the table border as mentioned in the section 2.1. Removing borders led to a significant improvement in the extraction as shown in the fig 5 below.

```
Name : Rajesh Hinduja Age/Gendei: 51 y /M Collected: 08/09/2020 01:31
Patient ID: 0010235933 Visit Type :IP Received : 08/09/2020 09:27
DOB :20/06/1969 Reported : 08/09/2020 12:01
Accession : 2001023647 Location :1$0613°1$R6181°1SB6181°1PULMMED

XN
[ DEPARTMENT OF LABORATORY MEDICINE - HAEMATOLOGY

Test Name | Unit | Reference Range Low Normal High

Complete Blood Count

Red cell .

Be eeWhoeweal x10*12/L 4555 423

eee Res g/dl. wore n9

Haematocrit

(EDTA Whole Blo % 40.0-50.0 8

MCV

(EDTA Whole Blood! fl S311 87.0

MCH

(EDTA Whole Bloo)t pg 27.0-32.0 28.1
```

Fig 5 indicates output of the laboratory report without border in the tables

Preserving the format and extracting numbers become most important when you want to detect lab metrics and their results accurately. This motivated us to use EasyOCR, an OCR engine built by Jaidedai. It is known to work well with numbers. The perks of using EasyOCR is that it takes a normal RGB image as input and not a binary image, it returns every word in the images separately with the bounding box of the word and the confidence on the extracted word. After using EasyOCR on a significant number of images we found out that it completely outperformed tesseract when extracting numbers and text from the table. Although its results were better than tesseract in most of the reports it had an anomaly as well. It was not performing well on skewed images although it was accurate in extracting the text, the bounding box coordinates were not accurate resulting in the disorientation of text. To tackle this problem we created a function that deskews the image. This change improved the coordinates significantly. A detailed explanation of each of these engines along with the image processing techniques we used and some custom functions we created to improve the results is given below.

### 2.1.1. Breaking down Tesseract Engine:-

Connected Component Analysis is the first step in maintaining object frames. Frames are gathered together, based on the nest, in the Blobs. Blobs are organized into

lines of text, and lines and circuits are analyzed to obtain a consistent tone or equal text. The lines are separated by words differently depending on the type of character space. Fixed pitch text is quickly cut into character cells Text that is measured is divided into words using specific spaces or complex spaces. Recognition continues as a two-world process. During the first pass, an effort was made to identify each word. Adequately identified words are transferred to the variable variant as training data. As a result, the dynamic segmentation gets a chance to improve the results between the text at the bottom of the page. Apply flexible separation training in the text next to the page as done in the second world, where poorly recognized words are also separated. The final section solves complex spaces and explores some of the x-height assumptions to find the smallest text.

### 2.1.2 Working of Tesseract engine on Medical Reports:-

As stated in the paper by SMITH[14], the Tesseract works excellent when we have images with dpi greater than 300. In our research, we made every image to a dpi of 300. Then we grayscale the image and apply the Sauvola Threshold to binarize the image. After binarization, we applied the Fast Non Local denoising method to denoise the image and then using the Tesseract OCR to extract the text from the image. On trying tesseract on several reports we found out that tesseract returned the text in distorted order which made it difficult for regex to find a pattern for metrics and to find its value. Tesseract didn't work on 30 images out of 85 and gave 65% or even less accuracy on them. This motivated us to look for a better OCR engine to deal with problems and our research found out EasyOCR, one of the engines that deal with the above-mentioned problems.

```
Name : Rajesh Hinduja Age/Gendei: 51 y /M Collected: 08/09/2020 01:31
Patient ID: 0010235933 Visit Type :IP Received : 08/09/2020 09:27
DOB :20/06/1969 Reported : 08/09/2020 12:01
Accession : 2001023647 Location :1$0613°1$R6181°1SB6181°1PULMMED

XN
[ DEPARTMENT OF LABORATORY MEDICINE - HAEMATOLOGY

Test Name | Unit | Reference Range Low Normal High

Complete Blood Count

Red cell .

Be eeWhoeweal x10*12/L 4555 423

eee Res g/dl. wore n9

Haematocrit

(EDTA Whole Blo % 40.0-50.0 8

MCV

(EDTA Whole Blood! fl S311 87.0

MCH

(EDTA Whole Bloo)t pg 27.0-32.0 28.1
```

Fig 6 indicates the output of the tesseract engine for the sample report mentioned in the Figure 2.

The benefit of using EasyOCR is that it gives the bounding box of the text extracted and gives the extracted text in the form of a list that makes it easy for manipulation.

### 2.1.3. EasyOCR:

EasyOCR is a python based OCR library that extracts the text from the image. It is based on research/codes from

several papers/open-source repositories, It's a ready-to-use OCR with 80+ languages supported including Chinese, Japanese, Korean Latin, Thai, etc. It's an open-source project licensed under Apache 2.0. Its deep learning part was built using the Pytorch library.

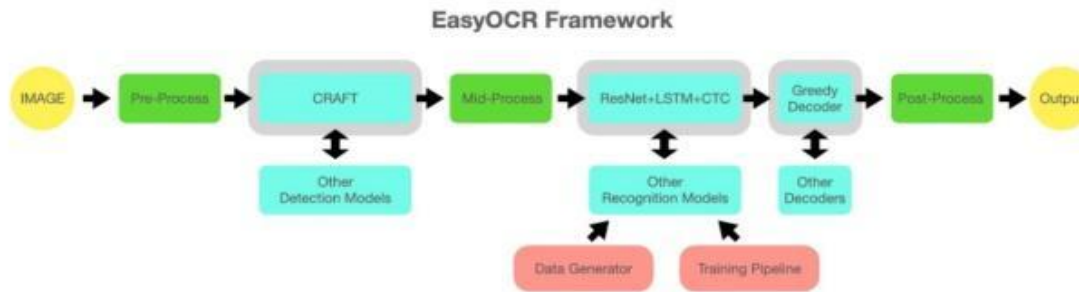


Image Source: [Github](#)

Fig 7 indicates the EasyOCR framework

It performs required preprocessing text for document image analysis within its library and extracts the text. It uses the CRAFT(Character Region Awareness for Text Detection)algorithm for text detection. CRAFT is a deep learning-based scene text detection method. It detects text area by exploring each character and affinity between the characters. The recognition model uses CRNN (Convolutional Recurrent Neural Network). It is composed of 3 main components, feature extraction, sequence labeling, and decoding. The feature extraction is performed using Residual Neural Network(Resnet), the sequencing labeling is performed by LSTM and CTC (Connectionist Temporal Classification), here the CTC is meant for labeling the unsegmented sequence data with RNN

### 2.1.4. Working of EasyOCR engine on Medical Reports:-

RGB images were passed to the EasyOCR engine; it returned the extracted text in the form of a list where the elements of the list were words. Using the EasyOCR output list we created another list in which elements of the list were lines of the report. Each line of the report was a different element of the list. We did this using the bounding boxes given by EasyOCR.The following figure shows the output given by EasyOCR and our created output. The purpose of doing this was to reduce the uncertainty in the detection of text for regex. After using it on a significant number of images we found out that EasyOCR is not accurate when the images are skewed. We leveled up this anomaly by deskewing the image in the

preprocessing step. Even though it performed well on images with tables, we removed the edges from the table to reduce the content and make EasyOCR focus entirely on text. These changes improved the results significantly and the accuracy increased by 13% from its previous accuracy. For a more detailed discussion on deskewing and table removal, visit section 2.1.8 and section 2.1.7 respectively.

```

Name Rajesh Hinduja Age/Gender: 51y / M Collected: 31/08/2020 01:42
Patient ID 0010235933 Visit Type IP Received : 31/08/2020 09.25
Referred By: DOB :20/06/1969 Reported : 31/08/2020 12.53
Accession 2001017500 Location :1S0613 1SR6181 1SB618] ` IINTMED
DEPARTMENT OF LABORATORY MEDICINE HAEMATOLOGY
Test Name Unit Reference Range Low Normal High
Complete Blood Count Red cell count x10^12/L 45-5.5 4.18
(EDTA Whole Bloo1l
Haemoglobin 13.0-17.0 12.2
(EDTA Whole
Haematocrit % 40.0-50.0 35.6
(EDTA Whole Bloo1l MCV fl 83-101 85.2
(EDTA Whole Bloo1l MCH pg 27.0-32.0 29.2
(EDTA Whole Bloo}l
MCHC 31.5-34.5 34.3
(EDTA Whole Bloo1l
RDW % 11.6-14.0 12.5
(EDTA Whole Bloo1l
Total Leukocyte Count x10^9/L 410 8.50
(EDTA Whole Bloo1lglutamated)
Neutrophils % 40-80 87.3
(EDTA Whole Bloo1l
Lymphocytes 0 20-40 7.2
(EDTA Whole Bloo1l
Monocytes % 2-10 5.3
(EDTA Whole Bloo1l ti
  
```

Fig 8 indicates the output of the EasyOCR engine for the sample report mentioned in the figure 2.

## 2.1.5 Thresholding Techniques:-

### 2.1.5.1 Otsu:-

The algorithm presumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their intra-class variance is minimal, or equivalently (because the sum of pairwise squared distances is constant) the inter-class variance should be maximal. The reason OTSU doesn't work well on images with non-uniform intensity is that it uses a global thresholding algorithm. Around 50 out of 85 images were binarized perfectly well by OTSU whereas 35 images didn't turn out to binarize well, making the binary image partially or completely black (refer Fig [9] and Fig [10a]). This motivated us to use local thresholding methods since they perform best when the images have non-uniform intensity.

### 2.1.5.2 Niblack:-

The thresholding value for each window is determined based on mean,  $m$ , and standard deviation,  $\sigma$  values of pixels in that window as the following

$$T = m + k \times \sigma \tag{1}$$

where  $k$  is  $-0.2$  as suggested by Niblack [5], and the window size is pre-determined by the user. Based on research done on this thresholding technique, this method can strongly identify the text body. But it also generates black noises or patches in empty windows. Since Niblack's algorithm determines a threshold value pixel-wise by sliding a rectangular window over the gray level image, it caused black noise to the image (refer Fig [9] and Fig [10b]). Using the Niblack threshold technique caused the noise to the image that didn't go completely even after using denoising techniques.

### 2.1.5.3 Savuola:-

This method was proposed by Sauvola et al. in 1997 [16]. This approach is inherited from the Niblack method. It has solved the issue of black noise which occurs in Niblack. The thresholding formula is as following:-

$$T = m \times \left[ 1 + k \times \frac{(1 - \sigma)}{R} \right] \tag{2}$$

where  $k$  is a controlling factor in the range of  $[0.2, 0.5]$ ,  $R$  is a predetermined image gray level value. The author suggested  $k=0.2$ ,  $R= 125$ . Sauvola worked perfectly on

around 83 images out of 85 (refer to Fig [9] and Fig [10c]). In the remaining 2 images, the image contrast between the text and background was extremely small. Sauvola doesn't work well when the contrast between background and foreground is small.

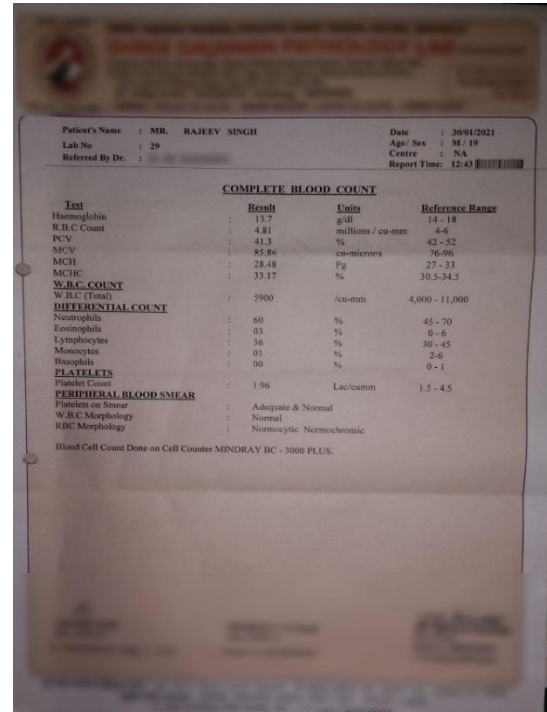


Fig 9 indicates the input laboratory report for the Otsu, Niblack and Sauvola Binarization techniques

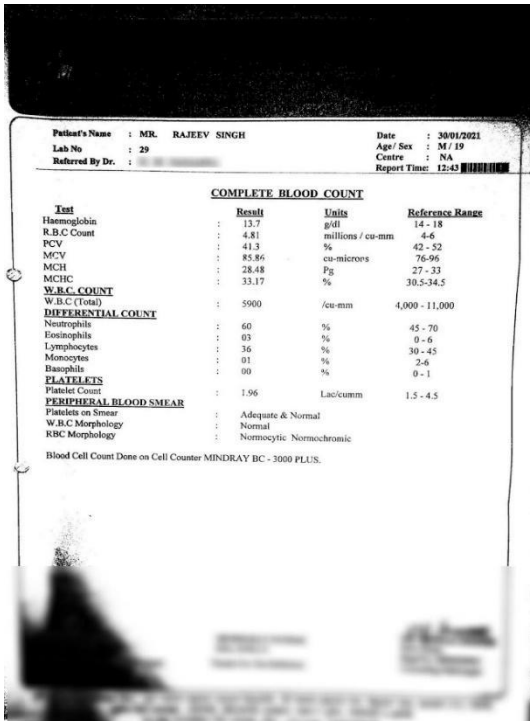


Fig. 10 a) indicates the output by Otsu Binarization on sample input report mentioned in the fig 9

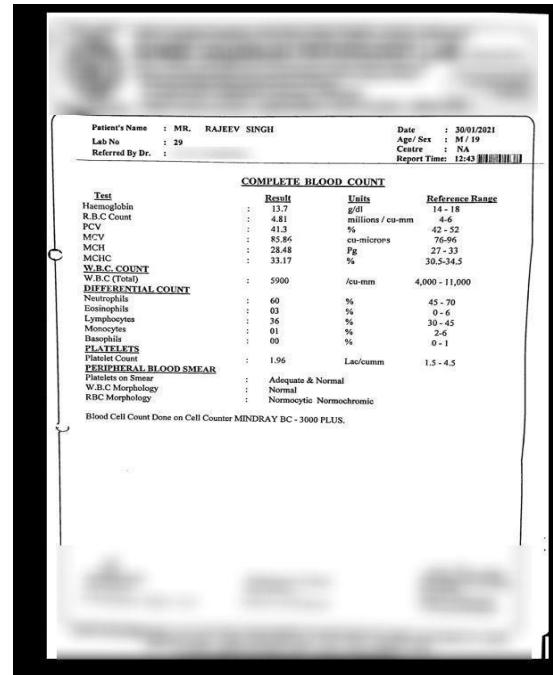


Fig. 10 c) indicates the output by Savuola Binarization on sample input report mentioned in the fig 9

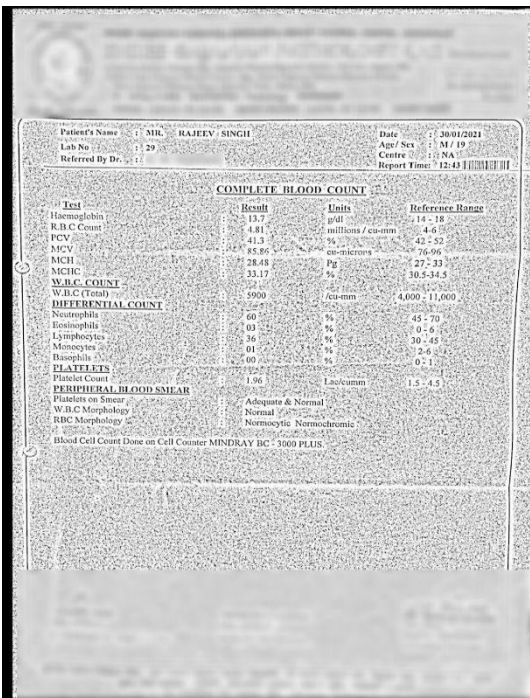


Fig. 10 b) indicates the output by Niblack Binarization on sample input report mentioned in the fig 9

### 2.1.5.4 Isauvola:-

This method has been proposed by Zineb Hadjadj[17]. This approach is inherited from the Sauvola method. It can successfully overcome the black noise problem and the thresholding in low contrast images problem. The thresholding formula is as following:-

$$T = m_w - \frac{m_w^2 - \sigma_w}{(mg + \sigma_w) \times (\sigma_{adaptive} + \sigma_w)} \quad (3)$$

where T is the thresholding value, mW is the mean value of the widow's pixels, σW is the standard deviation of the widow's pixels, mg is the mean value of all pixels in the image and σAdaptive is the adaptive standard deviation of the window. The local standard deviation method for each window is given by equation.

$$\sigma_{adaptive} = \frac{(\sigma_w - \sigma_{min})}{(\sigma_{max} - \sigma_{min})} \quad (4)$$

While using the Isauvola method, the text had become much thicker and thus we used morphology transformation techniques such as erosion and dilation. But the iteration for the erosion and dilation wasn't constant for all the images that resulted in images either

with bolder text or with faded text. Isavuola worked on just 20 Images out of 85(refer to Fig [11] and Fig [12]).

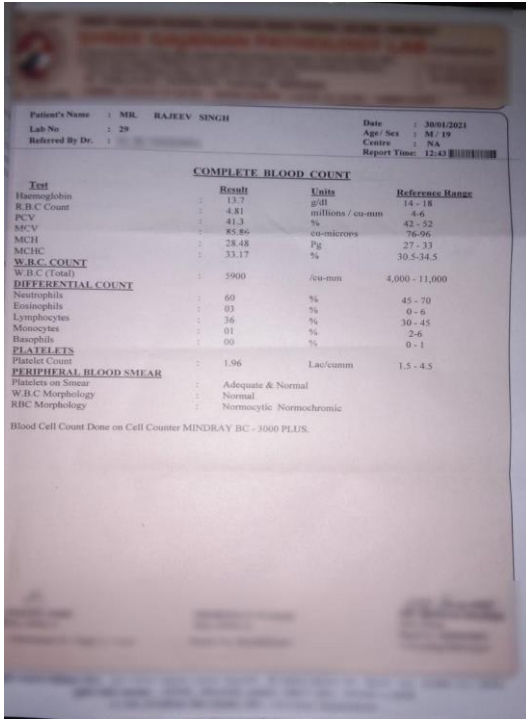


Fig. 11 indicates the input laboratory report for the ISavuola Binarization technique

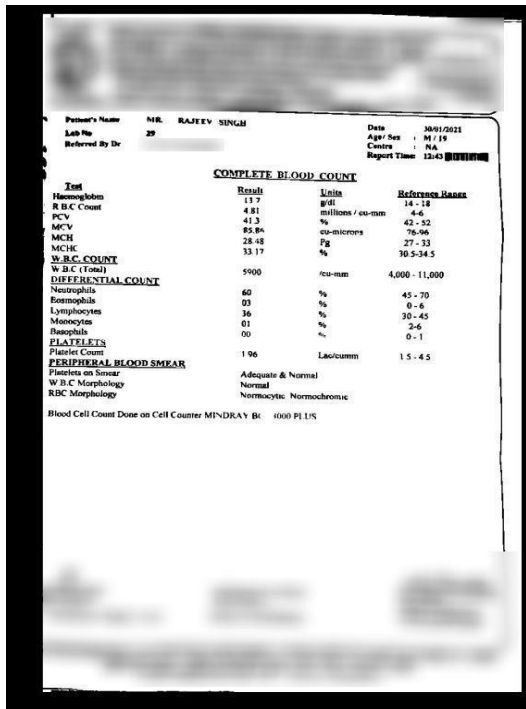


Fig 12 indicates the output by ISavuola Binarization on sample input report mentioned in the fig 11

After trying different thresholding algorithms we found out each method has its pros and cons. Sauvola was most accurate and gave the best results with the OCR engine. Even after binarizing the image in the best possible manner, these OCR engines make some alphanumerical errors due to background noise. The main objective of the *Noise removal* stage is to smoothen the image by removing small noises that are normally called dots or patches which comparatively have a higher intensity than the rest of the image. Enhancing the edges of an image can help the model to detect the features of an image more accurately.

### 2.1.6 Denoising and Blurring Techniques:-

Binarization of the image resulted in adding noise to the image. Noise is a random variation of pixel intensities or color information in images. Noise in Images is caused by reasons such as compression of images, the transmission of images, and many others. To remove the noise we tried several denoising techniques. There are many types of Noises such as Gaussian noise, Poisson noise, and salt and pepper noise. In this paper, we tried Gaussian blur, median blur, and Fast Non-local Algorithm for Image Denoising.

#### 2.1.6.1 Gaussian Blur:-

Gaussian Blur or Filtering takes place by convolving every point in the input array with a Gaussian Kernel. It is then summed up together to produce an output array. Since the text on the report is very small in size compared to the background, Gaussian Blur didn't work well and it reduced the details of the binarized images that made decimals(,), commas(,) completely invisible for the OCR engine.

#### 2.1.6.2 Median Blur:-

Median Blur is an offline digital filtering method, often used to remove sound from an image or signal. Such noise reduction is a common pre-processing step to improve the results of later processing (e.g., edge detection in the image). The median filter runs through each element of the signal (in this case the image) and replaces each pixel with the median of its neighboring pixels (located in a square neighborhood around the evaluated pixel). As median blur finds the median of neighboring pixels, it blurred some of the characters, especially when there was a line it faded the text near the line. As this method removed important features from the image, we decided not to use this method for denoising.

#### 2.1.6.3 Fast Non Local Means Denoising:-

Fast Non Local Means Denoising is an improvement towards the Non-local Means Denoising. Using a pre-classification process, only weights for the most meaningful pixels are computed. This pre-classification is a fast way to exclude dissimilar windows, which eventually



results in a smaller computation time and even in a better overall denoising quality. This fast approach is further optimized by taking advantage of the symmetry in the weights and by using a lookup table to speed up the weight computations. Refer Fig[9] and Fig[10c] for Sauvola thresholding with Fast Non Local Means Denoising.

**2.1.7 Using Contours to Remove Table border:-**

One of the major drawbacks of the OCR engines is that they struggle to detect tables from the documents. Some of our laboratory reports contained tables that made the extraction poor. So to overcome this limitation we decided to remove table edges from the image(refer Fig [13] and Fig [14]). We approached this problem by creating a custom function that detects vertical and horizontal edges separately using contours and then eliminates undesired contours with few conditions

**Steps by step implementation of the function are as follows:**

- 1) It first binarized the image using Sauvola thresholding,
- 2) And then finds the vertical contours(edges) using a vertical\_kernel of size (1,40) and then removes all vertical contours whose length is less than 1/3 of the height of the image.
- 3) Similarly, it finds the horizontal contours(edges) using a horizontal\_kernel of size(40,1) and then removes all the contours whose length is less than 1/2 of the width of the image.

Test Name	Unit	Reference Range	Low	Normal	High
Complete Blood Count					
Red cell count (EDTA Whole Blood)	x10 <sup>12</sup> /L	4.5-5.5	4.18		
Haemoglobin (EDTA Whole Blood)	g/dL	13.0-17.0	12.2		
Haematocrit (EDTA Whole Blood)	%	40.0-50.0	35.6		
MCV (EDTA Whole Blood)	fl	83-101		85.2	
MCH (EDTA Whole Blood)	pg	27.0-32.0		29.2	
MCHC (EDTA Whole Blood)	g/dL	31.5-34.5		34.3	
RDW (EDTA Whole Blood)	%	11.6-14.0		12.5	
Total Leukocyte Count (EDTA Whole Blood/automated)	x10 <sup>9</sup> /L	4-10		8.50	
Neutrophils (EDTA Whole Blood)	%	40-80			87.3
Lymphocytes (EDTA Whole Blood)	%	20-40	7.2		
Monocytes (EDTA Whole Blood)	%	2-10		5.3	

Fig 13 indicates the table with borders on which steps mentioned in 2.1.7 are performed

Test Name	Unit	Reference Range	Low	Normal	High
Complete Blood Count					
Red cell count (EDTA Whole Blood)	x10 <sup>12</sup> /L	4.5-5.5	4.18		
Haemoglobin (EDTA Whole Blood)	g/dL	13.0-17.0	12.2		
Haematocrit (EDTA Whole Blood)	%	40.0-50.0	35.6		
MCV (EDTA Whole Blood)	fl	83-101		85.2	
MCH (EDTA Whole Blood)	pg	27.0-32.0		29.2	
MCHC (EDTA Whole Blood)	g/dL	31.5-34.5		34.3	
RDW (EDTA Whole Blood)	%	11.6-14.0		12.5	
Total Leukocyte Count (EDTA Whole Blood/automated)	x10 <sup>9</sup> /L	4-10		8.50	
Neutrophils (EDTA Whole Blood)	%	40-80			87.3
Lymphocytes (EDTA Whole Blood)	%	20-40	7.2		
Monocytes (EDTA Whole Blood)	%	2-10		5.3	

Fig 14 indicates the output after removing contours mentioned in the section 2.1.7 on fig. 13

**2.1.8 Deskewing:-**

Scanned documents often become misaligned(slanted) during scanning because of misfeeds or other alignment errors. Skew is the amount of rotation necessary to return an image to horizontal and vertical alignment. Skew is measured in degrees. Deskewing is a process whereby a skew is removed by rotating an image by the same angle as its skew but in the opposite direction. This results in a horizontally and vertically aligned image where the text runs across the page rather than at an angle. When an image is not aligned correctly, optical character recognition (OCR) is more difficult and becomes slower and less accurate. Deskewing the documents in advance can make the OCR process faster and more accurate. We created a custom function that recognizes the skew in the image and deskews the image.

**Steps by step implementation of the function are as follows:**

- 1) It first inverse binarize the image (i.e text becomes white, and background becomes black).
- 2) After inverse binarizing, it finds text blocks present in the image. For finding the text blocks it merges all neighbouring characters of the document. It achieves it via dilation on the inverse binarized image (expansion of white pixels) with a larger kernel on the X-axis to get rid of all spaces between words, and a smaller kernel on the Y-axis to blend in lines of one block between each other but keeping the larger spaces between text blocks intact.
- 3) Next, it does contour detection on the dilated image with minimum area rectangle enclosing. It considers the block with the largest area and uses its angle to determine how skewed our image is and accordingly it creates a rotation matrix using the block angle to deskew the image. (Refer Fig [15] and Fig [16] for the result

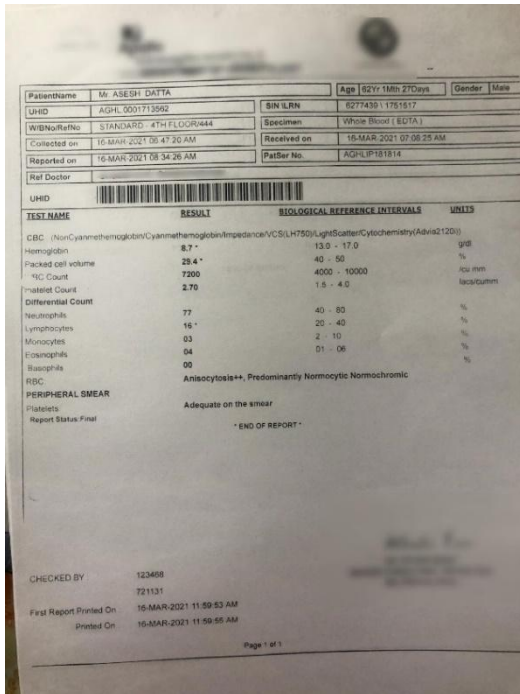


Fig 15 indicates the skewed laboratory reports on which steps mentioned in 2.1.8 are performed

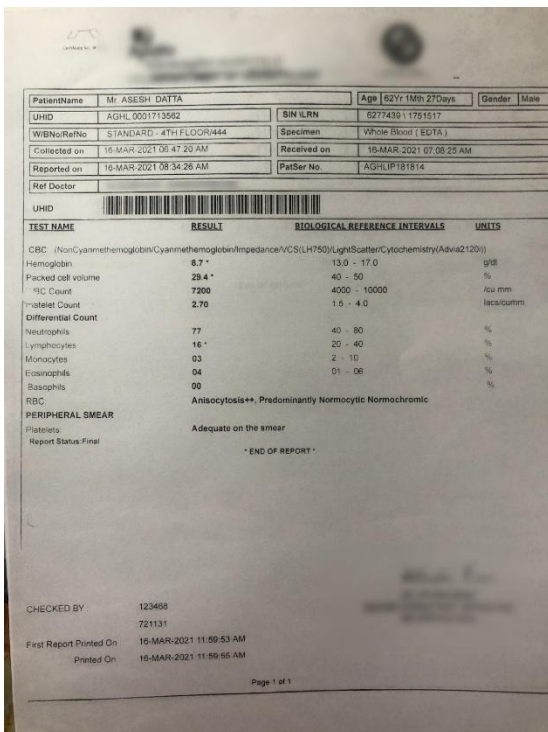


Fig 16 indicates the output after deskewing fig 15 with the steps mentioned in the section 2.1.8

After the text is extracted from the image successfully, the next step will be extracting the useful pieces of information

of the patients and storing them in the database(in our case it is in CSV format)

### 2.2 Regular Expression(ReGex):-

Every blood report consists of patient information, test names, results, and units. After successful extraction of the report text without altering its format, our next task was to detect this information from the extracted text. This brought our attention to Regular expression (regexes) a library built-in python often referred to as re. Regular expressions (regexes) are an abstraction of a keyword search that enables the identification of text using a pattern instead of an exact string. Due to the increasing popularity of ReGex, researchers have created many tools that support their creation, validation, and use. Common uses of regexes include locating content within a file, capturing parts of strings, and parsing user input. Although regexes are powerful and versatile, being a rule-based system it can be hard to understand, maintain, and debug, resulting in tens of thousands of bug reports. Regex is a rule-based system. Stronger the rule the better the regex pattern. An example of regular expression utilization is given below:

```
function      pattern      flags
r1 = re.compile("(0|-?[1-9][0-9]*)$", re.MULTILINE)
```

Fig 17. indicates Utilization of Regex.

A pattern is extracted from a utilization, as in Figure[17]. In essence, it is a string, but more formally it is an ordered series of regular expression language feature tokens. The pattern in Figure[17] will match if it finds a zero at the end of a line, or a (possibly negative) integer at the end of a line (i.e., due to the symbol “ -?” it means sequence which denotes zero or one instance of the -).

We found out that every CBC report has ‘Patient Name’, ‘Date’ and lab\_metrics such as *RBC*, *MCV*, *MCHC*, *MCH*, *PCV*, *Platelets*, *Hemoglobin*, *Hematocrit*, *Monocytes*, *Lymphocytes*, etc and this information can be easily retrieved from the text using regex patterns. The problem is not as simple as it appears; these lab\_metrics have their acronym and sometimes in reports these acronyms are used instead of their full form. As said before regexes are powerful and versatile, they can be hard to understand, maintain, and debugging one solution can lead to another problem. We went thoroughly through every possible acronym of these lab\_metrics we could find and developed an advanced regex pattern that detects these lab\_metrics from the report in their full form as well as in any possible acronym. A detailed explanation of each regex pattern is given below.

**2.2.1. Detecting Names of Patients:-**

The names of patients were given in various formats like “Patient’s Name: xyx”, “Name of Patient: xyx”, “Name: xyz”, and many more. So we first detected the name by searching a keyword **Name** in the extracted text. The pattern was very simple (**name**) with ignoring the case set as true. Next, We searched for a prefix title like Master, Mr, or so exist, and if it exists we ignore it since some paper reports mention the title and some reports don’t. The pattern used for detecting the prefix title is

```
(Mr|master|mrs|ms)[:\.\.]?s (5)
```

After this we can easily detect the name of the patient by using the pattern

```
([A-Z][a-zA-Z]*\s)([A-Z][a-zA-Z]*\.\.?) (6)
```

This will return the name of the patient. Both the patterns [5] [6] will ignore the case because we can’t tell whether the report may or may not have all letter capital or not.

**2.2.2. Searching for the date:-**

There are different formats for a date like dd/mm/yyyy or yyyy/mm/dd or dd/mm/yy or dd-mm-yy or maybe 12th Mar, 2020 in which we have an ordinal character and month is described using the first three characters of every month. As we can see the date and month are separated either through a slash( / ) or by a hyphen( - ). Another thing we have to add is the ordinal character for the numbers, since there are 4 ordinal characters namely th, st, nd, and rd. So the pattern for the pattern will be having option for putting 1 or 2 digits for date and 2-4 digits for the year final pattern will be

```
\d{2}([-/])\d{2}([a-zA-Z]{3})\1\d{4}\d{2}|\w{3}s\d{2}[.]\s\d{4} (7)
```

After extraction of the date, we further convert the date into pandas standard date format .eg such as 12th March, 2020 get converted to 12-03-2020, 2019/05/13 get converted to 13-05-2020.

**2.2.3. Detecting the values for the metrics:-**

For detecting the values for the metrics, we assume that the value of metrics will always come after the Metrics. For example, If we want to detect the value for the Hemoglobin, we have to search for the number after we have found Hemoglobin. Here comes the first issue, the regex pattern may detect the range or unit as a value.

To avoid this we made an advanced pattern

```
(?<[-|\d|\.\.])\d[\d|a-z]?[\.\.]?d*(?![-|\d|-\|^|/|,]) (8)
```

which ignores ranges and detects only the numerical value .It can also deal with scenarios where the extraction of numbers is wrong and a point ‘ . ’ will be detected as

comma ‘ , ’.In extraction numbers often get replaced with a similar alphabetical character or a special character. Eg 5 gets replaced as S or \$. To handle this we used leet(leetspeak) which is used for the alphabet in the initial days of the internet (refer Table 1 for more details).

Number	Symbols Similar to Number
1	'l', 'L', 'I', 'E', 'T', 'I', 'i', 'J', 'I'
2	'Z', 'R'
3	'E', '€', '[-', 'B', 'ß', '-]
4	'A', '@'
5	'S', '\$', '§', 'z'
6	'G', 'C-', 'b'
7	'T', 'Z', '^-'
8	'B', 'ß', 'oo', 'o', 'g'
9	'_0', 'g'
0	'C', 'O', 'D', 'Ø', 'o'

**Table 1 indicates the most common symbols the were incorrectly extracted from the OCR engines. To improve the accuracy, we have mapped the most symbols that resemble a number in actual**

Sometimes extraction makes an error in finding decimal ”.” in the numerical value. This result is the wrong extraction of the lab\_metric. Eg 13.4 gets extracted as 134.(refer Table 2 for extreme values) This makes the regex pattern detect 134 as the value of lab\_metrics which is not correct to avoid this if we filter the value detected. We created extreme ranges of the lab metrics. These are not normal ranges; these are the extreme values of the metrics. Having such value patients can go into shock. The extreme values were created by consulting medical expertise. Implementing this we assure that we don’t detect any wrong value.

Lab_metric	Extreme range
Hemoglobin	[10,19]
Hematocrit	[30,50]
PCV	[37,52]
MCV	[80,95]
MCH	[26,32]
MCHC	[30,36]
Platelets	[1.5,4.5]
Lymphocytes	[20,45]
Neutrophils	[40,80]
Monocytes	[1,10]
RBC	[4,6]

**Table 2 Table consists of the Range of each metrics. If our values are not in the range mentioned in the above table, we discard the value and store null as a result.**

**2.2.4. Regex Pattern for Detecting Hemoglobin:-**

In various medical reports, different acronyms of hemoglobin are used, e.g., hgb, hb, haemoglobin, we created a pattern that detects hemoglobin in any possible acronym.

The pattern is `[[h+][aemo*][glo*](b)[in]*]` (9)

As shown in the pattern [9] character h, and b is compulsory, we can even detect hemoglobin values even if we have an acronym. We also have to ignore the case by using re. After this, we can use the number pattern [8] to find the value for the Hemoglobin.

**2.2.5 Regex Pattern for Detecting Hematocrit:-**

The acronym for *Hematocrit* is *HCT*, so we just have to make a pattern for words Hematocrit and HCT. the Pattern would be

`[[h+][aemto*](c)[ri*](t)]` (10)

As shown in the pattern [10] alphabets h, c, t are compulsory characters, the reason for having this is we can even tackle the acronym also. The pattern also deals with small spelling errors like Hematocrit written as Haemotocrit or so, this helps to deal with maximum errors,

also we have to ignore the case of the pattern so that we won't miss the word HCT even if the case is different. After this, we can use the number pattern [8] to find the value for the Hematocrit.

**2.2.6. Pattern for detecting Platelets:-**

Platelets are also called *thrombocytes* or *PLT*, but normally thrombocytes aren't used in medical reports then also we have included it but didn't make many changes to the pattern. The pattern for the platelets will be `[[p+][\s]*][late*][\s]*[[l+][ae*][\s]*][t+][es*][\s]*[c+][ount*]|thrombocytes` (11)

We even have to ignore the case while checking the word. As you can see the characters P, L, T are compulsory characters to tackle the acronym for the platelet. For platelets after recognizing the pattern [11], we detect the values for it using pattern [8]. The values of Platelets are often in 3 different units i.e. Lac/cumm,  $10^3/uL$ , /cmm. This can create problems when plotting since the units are not similar. To overcome this problem we created a function that converts the detected number in Lac/cumm.

**2.2.7. Regex Pattern for RBC:-**

In various reports, RBC was also stated as the Red Blood Cell. So we created a pattern where characters R, B, C are compulsory in the text to be recognized as RBC/ Red Blood Cell. The pattern for recognizing RBC is

`(r)[ed*][\s]{1,3}(b)?[lood*][\s]{0,3}(c)[ell*][\s]{0,3}(count)?` (12)

As we can see it also takes a condition where we have a point ( '.' ) between the characters in the case of acronyms. Also, we have to ignore the case while searching so as not to encounter different case issues. After this step, we will use the pattern [8] to get the RBC Count's value.

**2.2.8. Pattern for recognizing Monocytes:-**

Since there is no acronym for Monocytes, we just have to make a pattern that directly detects the word Monocytes. Here if we have an extraction error, the pattern won't work and return none and the value for the Monocytes will be stored as Null. The pattern for Monocytes is

`[<! [a-zA-z]](M[on]{2,3}[ocytes]+)` (13)

for this case even we have to ignore the case of the word. After finding the pattern, we will search the value for the Monocytes using the method given in 2.2.3 to get the value.

**2.2.9. Using Regex Pattern for detecting MCV:-**

Full-Form of MCV is Mean Corpuscular Volume, even known as erythrocytes. Name Erythrocytes are not used for medical reports but for safety we even used them so that we don't miss any edge cases. Pattern will be

`(([m]+)([\.\s]*)([ean]*)([\.\s]*)(c+)([orpuscular]*)([\.\s]*)([v]+)([colume]*| erythrocytes)` (14)

As shown in the pattern [14] characters M, C, V are compulsory, the reason for this is we have to handle the acronym also. We also have handled a case where we can have a full stop between every character in the case of the acronym. Ignoring the case will help us handle an edge case where we may get the first letter upper case and others in the lower case. After this, we will use method 2.2.3 to extract the value of the MCV.

#### 2.2.10. Pattern for PCV:-

Different terms used for PCV are P.C.V., Packed Cell Volume. For handling the acronym, we made characters P, C, V compulsory and even putting a space or a full stop between every character of the acronym. Pattern will be

`(([p]+)([\.\s]*)([acked]*)([\.\s]*)(c+)([ell]*)([\.\s]*)(v+)([colume]*)` (15)

the pattern [15] should use the ignore case function to ignore the case of the word. After detecting the metric we can search for the value of it using the 2.2.3 method.

#### 2.2.11. Regex Pattern for MCH:-

The full Form of MCH is Mean Corpuscular Hemoglobin. To detect it we have to make characters M, C, H compulsory characters to cover the probability of acronyms. The pattern for detecting MCH value will be

`(([m]+)([\.\s]*)([ean]*)([\.\s]*)(c+)([orpuscular|ell]*)([\.\s]*)([h]+)([aemoglobin]*)` (16)

After recognizing the Metrics, we will detect the values of the metrics using method 2.2.3

#### 2.2.12. Searching for MCHC:-

The pattern for MCHC will have 2 scenarios, first of detecting the word MCHC and other detecting the full form of MCHC which is Mean Corpuscular Hemoglobin Concentrated. In the pattern, the characters M, C, H, C are mandatory also there might be a full stop or space between the characters in MCHC. So the pattern will also handle this scenario. The pattern will also have to ignore the case of the word. The final pattern is

`(([m]+)([\.\s]*)([ean]*)([\.\s]*)(c+)([orpuscular|ell]*)([\.\s]*)([h]+)([aemoglobin]*)([\.\s]*)([c]+)([oncentration]*)` (17)

After this pattern is detected, we will search the value of the metrics using method 2.2.3

#### 2.2.13. Recognizing Lymphocytes using Regex:-

The pattern will be very much similar to the pattern of the Monocytes. Here we will search the lymphocytes and if we found the word we will use method 1.4.3 to find the value

of the metrics otherwise we will return Null into the CSV file. The pattern will be

`(?<![a-zA-z])(l[ymph]{2,4}[ocytes]+)` (18)

pattern [18] will also ignore the case of the word to get more accurate results.

#### 2.2.14. Finding the word Neutrophils:-

The pattern for the Neutrophils will just search for the word 'Neutrophils'. The pattern we used is `n[eutro]{3,5}[phils]+` (19)

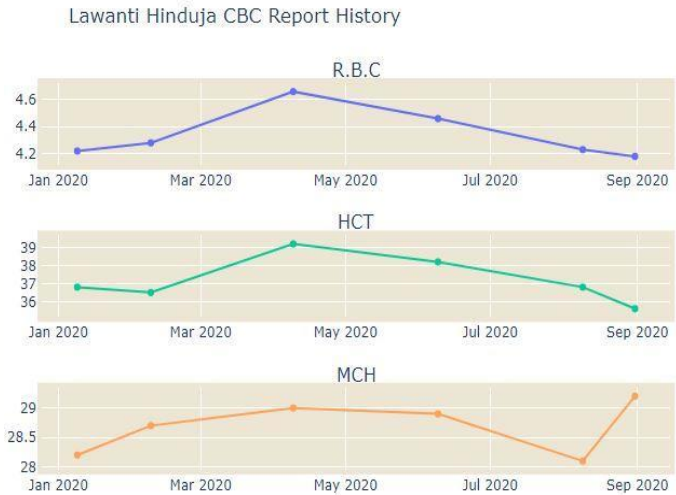
If the Regex finds the pattern, it will detect the value for Neutrophils using method 2.2.3 or it will return Null. The pattern[19] will also ignore the case and find the word using the pattern.

After investing significant hours in training patterns and debugging every single error on around 50 reports. We finalized the above-mentioned patterns as they were most accurate in detecting the exact required text on both of our train and test data. After finalizing the pattern, our next aim was to store this information and present it in the best possible manner.

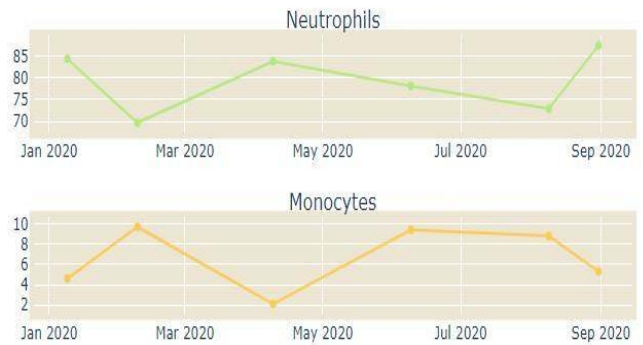
#### 2.3 Saving the Data in CSV:-

Our proposed algorithm stores the information of each patient in a separate CSV file. Whenever it sees a new patient whose reports it didn't find in the directory it creates a new CSV file with the name of the file as the name of the patient that it has extracted during the document analysis. This CSV file contains the date of the report, lab\_metrics, and their corresponding values.

The metrics that are not in the report or that did not get extracted well by the OCR engines, will be stored as none instead of giving a false value. It sorts the entries according to the date so that the CSV file can then be easily used for plotting each metric according to the timeline to have an analysis of how we are maintaining our health. Using this CSV file, it plots the graph using the plotly library in python, making the graph much more interactive to the user(patient). The following figure was obtained by passing 6 reports of a person to the proposed algorithm one by one. The algorithm stored the data of the person in CSV form and plotted graphs for each metrics separately.



(a)



(d)

Fig 17 (a),(b),(c),(d) indicates the interactive plot created by our proposed system based on the metrics value identified by the advance regex patterns.

### 3. Datasets and Performances

#### 3.1 Datasets:-

In this research, we gather 120 CBC Reports from several sources. On average we have 5-6 reports per patient. Out of which we used 85 reports for the process improvement of the model and 35 reports for testing the model.

#### 3.2 Evaluation Metrics:-

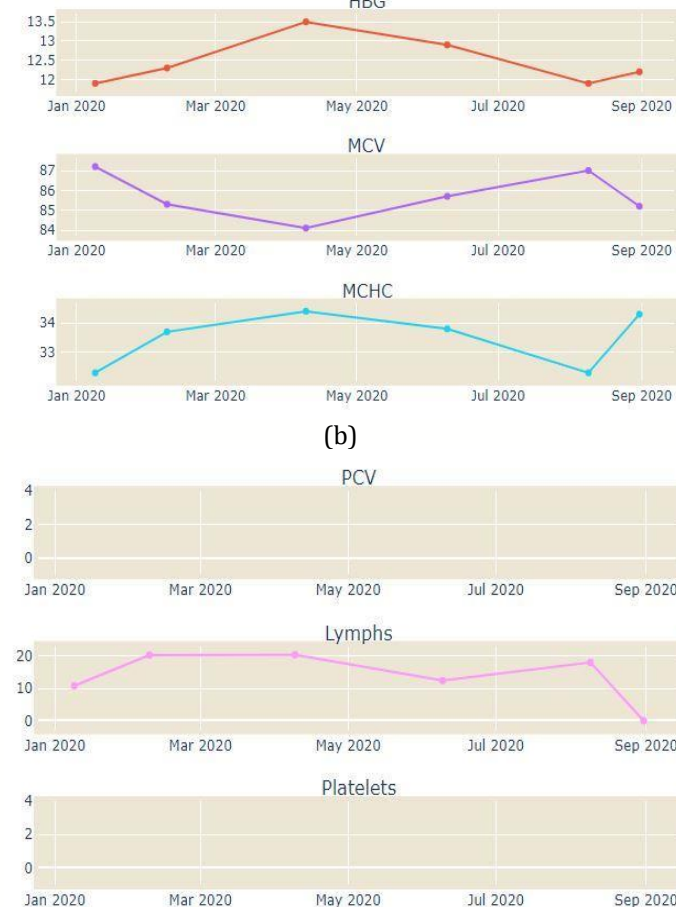
For calculating the accuracy of the algorithm. We used a simple accuracy metric that divides the total number of correct extraction by the total number of extraction in the entire test set.

$$Accuracy_{avg} = \frac{1}{n} \sum_{i=1}^n \frac{Extraction_{correct}}{Extraction_{total}} \times 100 \quad (20)$$

	RBC	HGB	Neutrophils	MCV	MCH
<b>Actual</b>	4.18	13.7	60	85.86	28.4
<b>Predicted</b>	-	13.7	60	85.86	28.4

	PCV	HCT	Lymph	Monocytes	Platelets	MCHC
<b>Actual</b>	41.3	-	36	01	1.96	33.17
<b>Predicted</b>	41.3	-	36	1	1.96	33.17

Table 3 For a clear understanding below is the accuracy calculation for the sample report mentioned in Fig 11



(c)

$$\text{Accuracy} = \frac{10}{11} \times 100 = 90.9\%$$

### 3.3 Performance analysis:-

Our current proposed algorithm takes 2.1 seconds on average to extract information from a pixelated report and stores or updates the extracted information in the CSV file of the patient and plot an interactive graph. It determines the Name of the patient, Date of the report, and 11 lab metrics from each report with an accuracy of 95%.

### 4. Conclusion & Future Work:-

The research's main objective is to understand the patient's perspective of the ease of understanding their laboratory reports. This study is the first of its kind to understand the laboratory reports using classical computer vision techniques and regular expressions in Natural Language Processing (NLP). Our results have shown an accuracy of 95% in the data extraction from the reports with a detailed visualization of every metric mentioned in the report. The system also incorporated a sorting technique using regular expressions which reduces the manual intervention of the patient to sort the report before uploading. This system has significant applications in pathological labs and patients who would like to monitor their health history. Our future work would not just limit to CBC reports but can be scaled to all the types of reports that include sugar fasting, Albumin tests, and others. Considering the types of tests and the scale of the reports, the next step would be to also incorporate a deep learning approach to improve our OCR and eliminate the need for a rule-based regular expressions approach with enhanced methods of NLP being introduced.

### References:-

[1] Liu, Jinhui & Jain, Anil. (2000). Image-based form document retrieval. *Pattern Recognition*. 33. 503-513. 10.1016/S0031-3203(99)00066-7.

[2] Zhu, G. & Doermann, D.. (2007). Automatic Document Logo Detection. *Document Analysis and Recognition, International Conference on*. 2. 864-868. 10.1109/ICDAR.2007.68.

[3] Dan Claudiu Ciresan and Ueli Meier and Luca Maria Gambardella and Jurgen Schmidhuber, "Convolutional Neural Network Committees for Handwritten Character Classification", 2011 International Conference on Document Analysis and Recognition, IEEE, 2011

[4] Badawy, W. "Automatic License Plate Recognition (ALPR): A State of the Art Review." (2012): 1-1.

[5] Andrew S. Agbemenu KNUST Kumasi, Ghana Ernest O. Addo "An Automatic Number Plate Recognition System

using OpenCV and Tesseract OCR Engine" *International Journal of Computer Applications* (0975 - 8887) Volume 180 - No.43, May 2018

[6] Li, Xiaojun & Wang, Weiqiang & Jiang, Shuqiang & Gao, Wen. (2008). Fast and effective text detection. *Proceedings / ICIP ... International Conference on Image Processing*. 969 - 972. 10.1109/ICIP.2008.4711918.

[7] Puri, Shalini & Khan, Nouman. (2016). A study on text detection techniques of printed documents. 10.1109/WiSPNET.2016.7566589.

[8] H.S. Ackley. "Methods for optical character recognition (OCR)". Publication of US20102649A1, 2017

[9] Shrey Dutta, Naveen Sankaran, Pramod Sankar K., C.V. Jawahar, "Robust Recognition of Degraded Documents Using Character N-Grams", IEEE, 2012

[10] Chirag Patel, Atul Patel and Dharmendra Patel. "Optical Character Recognition By open source OcrOCR Tool Tesseract". *International Journal of Computer Applications* (0975 - 8887), October 2012

[11] Remus Petrescu, Sergiu Manolache, Costin-Antin Boingiu, Giorgina Violetta, Cristian Avatavului, Marcel Prodan and Ion Bucur. "Combines Tesseract and aspire Results to improve OCR Text Detection Accuracy" Article in *Journal of Information Systems Management* · May 2019

[12] Aggarwal V., Jajoria S., Sood A. (2018) Text Retrieval from Scanned Forms Using Optical Character Recognition. In: Urooj S., Virmani J. (eds) *Sensors and Image Processing, Advances in Intelligent Systems and Computing*, vol 651. Springer, Singapore

[13] Harraj, A.E.; Naoufal, R. OCR Accuracy Improvement on Document Images through a Novel Pre-Processing Approach. arXiv 2015, arXiv:1509.03456.

[14] SMITH, R. 2007. An Overview of the Tesseract OCR Engine. In *proceedings of Document analysis and Recognition. ICDAR 2007. IEEE Ninth International Conference*.

[15] Niblack, W.: An introduction to digital image processing (1985)

[16] Sauvola, Jaakko & Seppänen, Tapio & Haapakoski, Sami & Pietikäinen, Matti. (1997). Adaptive Document Binarization. *Pattern Recognition*. 33. 147-152 vol.1. 10.1109/ICDAR.1997.619831.

[17] Zineb Hadjadj1,2(&), Abdelkrim Meziane2, Yazid Cherfa1, Mohamed Cheriet3, and Insaf Setitra2. ISauvola: Improved Sauvola's Algorithm for Document Image Binarization

[18] J. Liu and A.K.Jain, "Imaged-Based Form Document Retrieval," *Pattern Recognition*, vol. 33, no.3, pp.503-513, 2000.

[19] Manesh B. Kokare, M. S. Shirdhonkar, "Document Image retrieval: An Overview," *International Journal of Computer Applications*, vol. 1, no. 7, pp. 128-133, 2010.

- [20] Bilal Bataineh<sup>1</sup>, Siti N.H.S. Abdullah<sup>2</sup>, K. Omar<sup>3</sup>, and M. Faizul Adaptive Thresholding Methods for Documents Image Binarization
- [21] Keshao D. Kalaskar and Mahendra P. Dhore Preprocessing Challenges in Document Image Analysis
- [22] Pratik Madhukar Manwatkar and Dr. Kavita R. Singh A Technical Review on Text Recognition from Images IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO) 2015
- [23] ARCHANA A. SHINDE, D. 2012.Text Pre-processing and Text Segmentation for OCR. International Journal of Computer Science Engineering and Technology, pp. 810-812.
- [24] E. Larson, "[Research Paper] Automatic Checking of Regular Expressions," 2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM), Madrid, Spain, 2018, pp. 225-234, doi: 10.1109/SCAM.2018.00034.
- [25] Abd Al-salam Selami, Ameen & Fadhil, Ahmed. (2016). A Study of the Effects of Gaussian Noise on Image Features. Kirkuk University Journal / Scientific Studies (1992-0849). 11. 152 - 169. 10.32894/kujss.2016.124648.
- [26] Dauwe, A & Goossens, Bart & Luong, Hiep & Philips, W. (2008). A fast non-local image denoising algorithm. Proc SPIE. 6812.
- [27] Mustafa, Suleiman & oyeniran, kola. (2017). Comparing Median and Gaussian Blurring for GrabCut Segmentation of Melanoma.
- [28] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty and R. Ray, "Adaptive thresholding: A comparative study," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, 2014, pp. 1182-1186, doi: 10.1109/ICCICCT.2014.6993140.
- [29] Gatos, B., Ntirogiannis, K., Pratikakis, I.: ICDAR 2009 Document Image Binarization Contest. In: 10Th International Conference on Document Analysis and Recognition, Beijing, China (2009)
- [30] Abbas Hussien Miry. Article: Iterative Thresholding and Morphology Operation based Melanoma Image Segmentation. *International Journal of Computer Applications* 118(2):15-19, May 2015
- [31] Zacharias, E., Teuchler, M., & Bernier, B. (2020). Image Processing Based Scene-Text Detection and Recognition with Tesseract. *ArXiv, abs/2004.08079*.
- [32] del Ninno E., Nicchiotti G., Ottaviani E. (1997) A general and flexible deskewing method based on generalized projection. In: Del Bimbo A. (eds) *Image Analysis and Processing. ICIAP 1997. Lecture Notes in Computer Science*, vol 1311. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-63508-4\\_177](https://doi.org/10.1007/3-540-63508-4_177)
- [33] Islam, Noman, Zeeshan Islam and Nazia Noor. "A Survey on Optical Character Recognition System." *ArXiv abs/1710.05703* (2017): n. pag.