



Ain Shams University

Faculty of Computer & Information Science

Bioinformatics Department

Automated Complete Blood Count test

**This documentation is submitted as required for the degree of bachelors in
Computer and Information Sciences.**

By

Habiba Alaa El-Din Hassan [BIO]

Khaled Mohamed Badawi [BIO]

Rana Hesham Mamdouh [BIO]

Saifeldin Ayman Abodaif [BIO]

Shehab Mohamed Mamdouh [BIO]

Mohaned Mohamed Abd-El-Hafiz [BIO]

Under Supervision of

Dr. Salsabil Amin

Basic Science Department,

Faculty of Computer and Information Sciences,

Ain Shams University.

T.A Esraa Hamdi

Bioinformatics Department,

Faculty of Computer and Information Sciences.

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people, and we are extremely privileged to have got this all along the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them all.

All praise and thanks to ALLAH, who provided me the ability to complete this work. I hope to accept this work from me.

I am grateful to my parents and my family who are always providing help and support throughout the whole years of study. I hope I can give that back to them.

I also offer my sincerest gratitude to my supervisors, Dr. **Salsabil Amin**, T.A **Esraa Hamdi** who have supported me throughout my thesis with their patience, knowledge and experience.

Finally, We would like to thank my friends and all the people who gave me support and encouragement.

Abstract

A complete blood cell count is an important test in medical diagnosis to evaluate overall health condition. Traditionally blood cells are counted manually using hemocytometer along with other laboratory equipment and chemical compounds, which is a time-consuming and tedious task.

In this work, we present a machine learning approach for automatic identification and counting of three types of blood cells using ‘you only look once’ (YOLO) object detection and classification algorithm.

YOLO framework has been trained with BCCD Dataset of blood smear images to automatically identify and count red blood cells, white blood cells, and platelets. Moreover, this study with other convolution neural network architectures considers architecture complexity, reported accuracy, and running time with this framework and compares the accuracy of the models for blood cells detection. We also tested the trained model on smear images from a different dataset and found that the learned models are generalized.

Overall the computer-aided system of detection and counting enables us to count blood cells from smear images in less than a second, which is useful for practical applications.

Our aim is to develop an easier method to perform complete blood count tests with less time and a higher accuracy and reduce human intervention.

We developed a web/desktop application, a machine learning framework that helps in counting red blood cells (rbcs) ,white blood cells (wbcs) and platelets.

Table of Contents

Chapter	Page
Acknowledgement	1
Abstract	2
Table of Contents	3
List of Figures	4
List of Abbreviations	7
1- Introduction	8
1.1 Motivation.....	8
1.2 Problem Definition.....	9
1.3 Objective.....	10
1.4 Document Organization.....	11
2- Background	12
2.1 Similar Systems.....	15
3- Analysis and design	17
3.1 System Overview.....	17
3.2 System Users.....	23
3.3 System Analysis and Design.....	24
4- Implementation	34
4.1 Dataset preparation.....	34
4.2 Training model using google colab.....	35
4.3 Developing desktop application using python.....	39
4.4 Database using MySQL.....	41
4.5 Developing a website using ruby on rails.....	45
5- User Manual	56
5.1 Desktop application data entry form.....	56
5.2 Desktop application report form.....	60
5.3 Website.....	61
6- Conclusions and Future Work	65
6.1 Conclusions.....	65
6.2 Future Work.....	66

List of Figures

Figure 1-hemocytometer	9
Figure 2-blood smear	10
Figure 3-normal vs leukemia	12
Figure 4-CBC	13
Figure 5-RBC	14
Figure 6-system architecture	18
Figure 7-use case	24
Figure 8-class diagram	26
Figure 9-sequence diagram1	28
Figure 10-sequence diagram2	29
Figure 11-sequence diagram3	30
Figure 12-sequence diagram4	31
Figure 13-sequence diagram5	32
Figure 14-Database Diagram	33

Figure 15-labeled blood smear from dataset	34
Figure 16-flowchart1	35
Figure 17-Training phase	36
Figure 18-ms coco object detection	37
Figure 19-YOLO Output	38
Figure 20-flowchart2	40
Figure 21-blood smear after detection	41
Figure 22-database users table	44
Figure 23-database reports table	44
Figure 24-MVC	45
Figure 25-Rails architecture	48
Figure 26-Data flow	49
Figure 27-Controller Methods	50
Figure 28-Flowchart3	51
Figure 29-Docker Steps	55
Figure 30-Data entry form	56
Figure 31-Error	57

Figure 32-Browse Image	58
Figure 33-Before detection	59
Figure 34-Result	60
Figure 35-Login page	61
Figure 36-Registration page	62
Figure 37-Report page	63
Figure 38-Add report page	64
Figure 39-Elements of blood	66

List of Abbreviations

Word	Abbreviation for
YOLO	you only look once
RBCS	Red blood cells
WBCs	White blood cells
PLT	Platelets
BCCD	Blood cell count and detection
CBC	Complete blood count
HCT	Hough circle transform
AIDS	Acquired immunodeficiency syndrome
CNN	Convolutional neural network
R-CNN	Regional neural network
mAP	Mean Average precision
DBMS	Database management system
RDBMS	Relational database management system
JWT	JSON web token

Chapter 1

Introduction

1.1 Motivation

Blood cells are produced and mature primarily in the bone marrow and, under normal circumstances, are released into the bloodstream as needed

A complete blood count is a common blood test that's done for a variety of reasons:

- To review your overall health: Your doctor may recommend a complete blood count as part of a routine medical examination to monitor your general health and to screen for a variety of disorders, such as anemia or leukemia.
- To diagnose a medical condition: Your doctor may suggest a complete blood count if you're experiencing weakness, fatigue, fever, inflammation, bruising or bleeding. A complete blood count may help diagnose the cause of these signs and symptoms. If your doctor suspects you have an infection, the test can also help confirm that diagnosis.
- To monitor a medical condition: If you've been diagnosed with a blood disorder that affects blood cell counts, your doctor may use complete blood counts to monitor your condition.
- To monitor medical treatment: A complete blood count may be used to monitor your health if you're taking medications that may affect blood cell counts.

1.2 Problem Definition

Cell Counting with a Hemocytometer

The hemocytometer is divided into 9 major squares of 1mm x 1mm size. The four corner squares (identified by the red square) are further subdivided into 4 x 4 grids. The height of the chamber formed with the cover glass is 0.1 mm, so a 1 mm x 1 mm x 0.1 mm chamber has a volume of 0.1 mm³ or 10⁻⁴ ml.

To count cells using a hemocytometer, add 15-20µl of cell suspension between the hemocytometer and cover glass using a P-20 Pipetman. The goal is to have roughly 100-200 cells/square. Count the number of cells in all four outer squares divided by four (the mean number of cells/square). The number of cells per square x 10⁴ = the number of cells/ml of suspension. This protocol works well for either adherent mammalian cells that have been trypsinized or for suspension cells including Sf9 insect cells. Red blood cells are typically too small and numerous for this protocol and utilize the middle square instead.

When finished, spray the hemocytometer and cover slip with 70% ethanol to kill the cells. Wash both with deionized water and wipe dry with a Kimwipe. Wrap in a clean Kimwipe and return to the storage box. Note, the coverslips for the hemocytometer are made of a special thicker/flatter glass. Please try to avoid breaking or losing it. If you do, reorder hemocytometer cover slips, not regular cover slips.

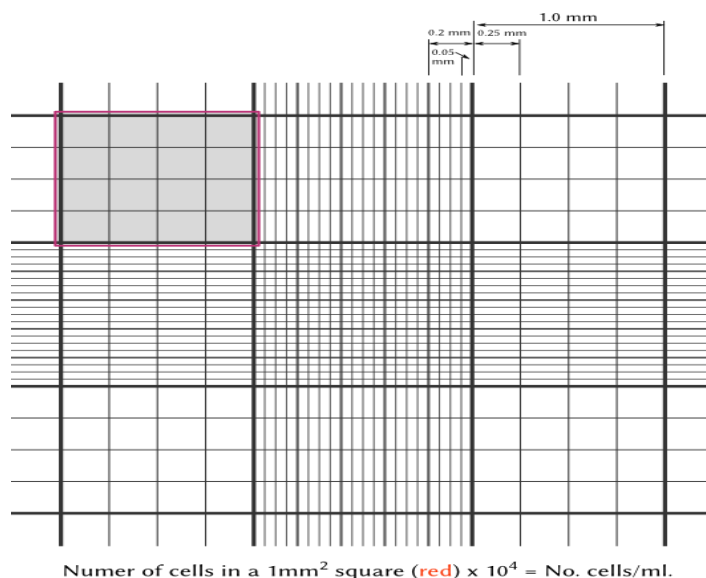


figure 1-hemocytometer

1.3 Objective

The main objective of our idea is to use the yolo model to detect the red blood cells, white blood cells and platelets using blood smear .

This method saves time and is more accurate than counting manually using hemocytometer along with other laboratory equipment and chemical compounds, which is time-consuming.

Using automated complete blood count improves the quality of determining your general health status; to screen for, diagnose, or monitor any one of a variety of diseases and conditions that affect blood cells, such as anemia ,bleeding disorder and cancer .

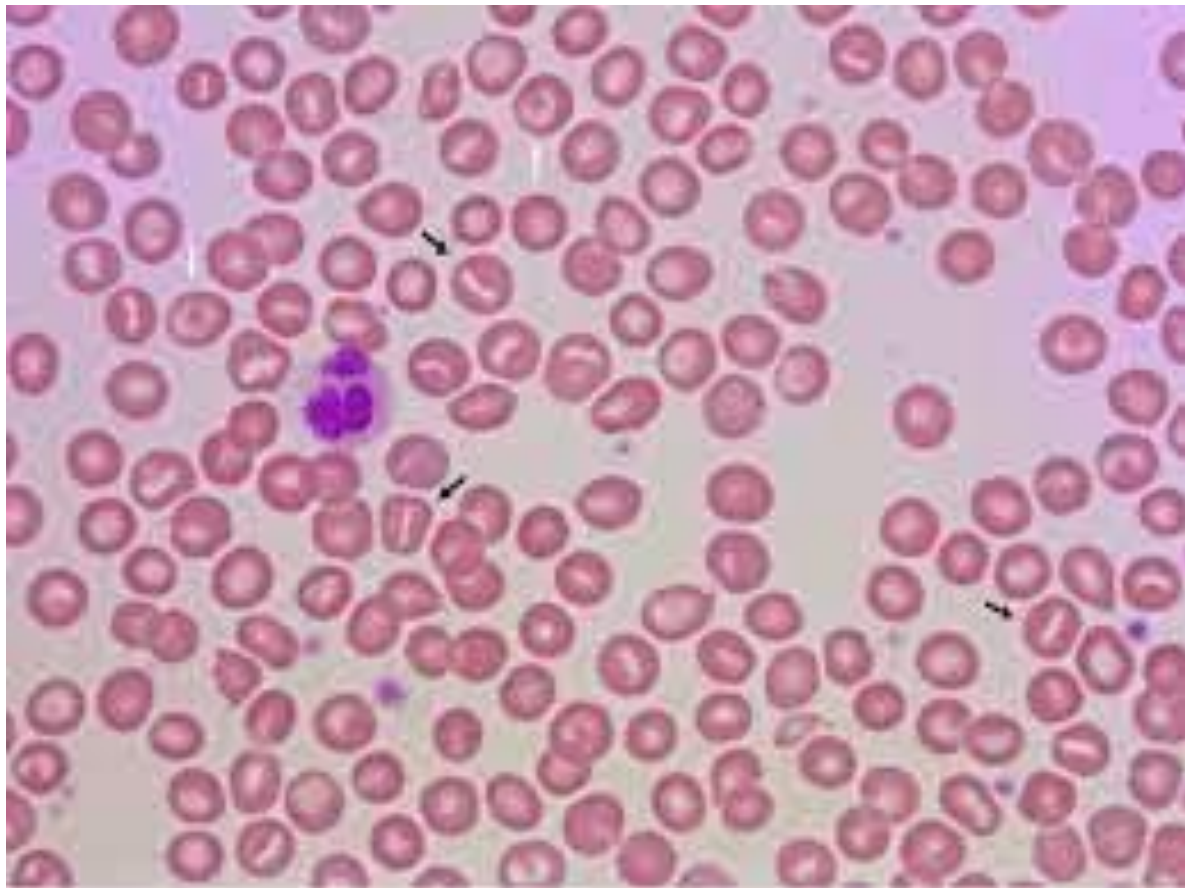


figure 2-blood smear

1.4 Document Organization

this document is divided into six chapters describing all aspects of our project as follows:

- **Chapter 2 (background):**

This chapter, we define the project field, scientific background, and the objective of the system.

- **Chapter 3 (Analysis and design):**

This chapter describes the functional requirements, users of the system, and contains all system analysis diagrams and the specifications of each one.

- **Chapter 4 (Implementation):**

This Chapter describes in detail the usage of the pretrained model, the algorithms and the techniques that were used to adapt this project starting with using pycharm tool to make the desktop application then using visual studio code to develop the web service.

- **Chapter 5 (user manual):**

This chapter contains full instructions and guides that describe how to use the automated complete blood count web and desktop application.

- **Chapter 6 (conclusion and future work):**

This chapter contains our conclusion after the implementation of the project and what we have done. At the end we describe our future vision about this project.

Chapter 2

Background

Complete blood count is required for overall health evaluation and diagnosis of many disorders including anemia, infection and leukemia, complete blood count is required. The human blood consists of three types of blood cells such as red blood cell (RBC), white blood cell (WBC) and platelets (PLT) . A person's health is determined using complete blood count. Blood cell segmentation and identification is important as blood being a health indicator. Abnormal increase or decrease in cell count indicates that a person has an underlying medical condition.

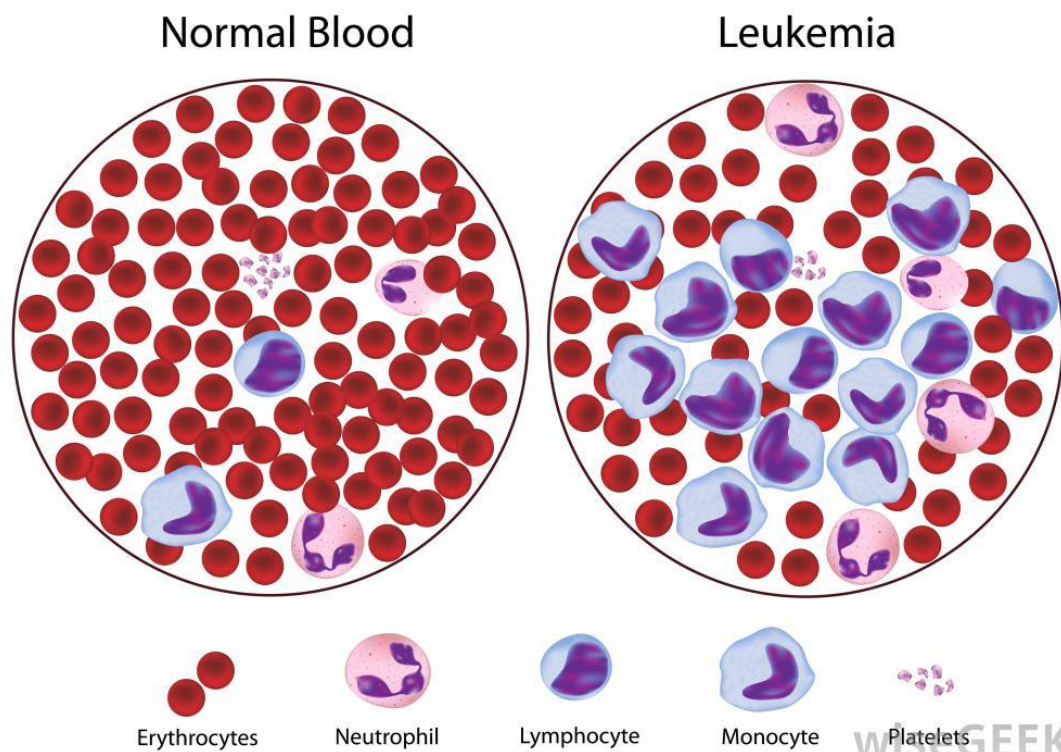


figure 3 -Normal vs leukemia

Red blood cells, also known as erythrocytes are the most important and numerous blood cells in the human body. Main function of RBCs is to carry oxygen and deliver it to the cells in the body. They are minute disc shaped. They do not contain nucleus but a protein called hemoglobin. Both inner and outer layers of cells are made of protein that gives red color to blood. Hemoglobin actually does the work of grabbing and carries oxygen. Usually the level of hemoglobin is tested in blood tests. Decrease in level may cause severe diseases including anemia, blood loss, leukemia and malnutrition.

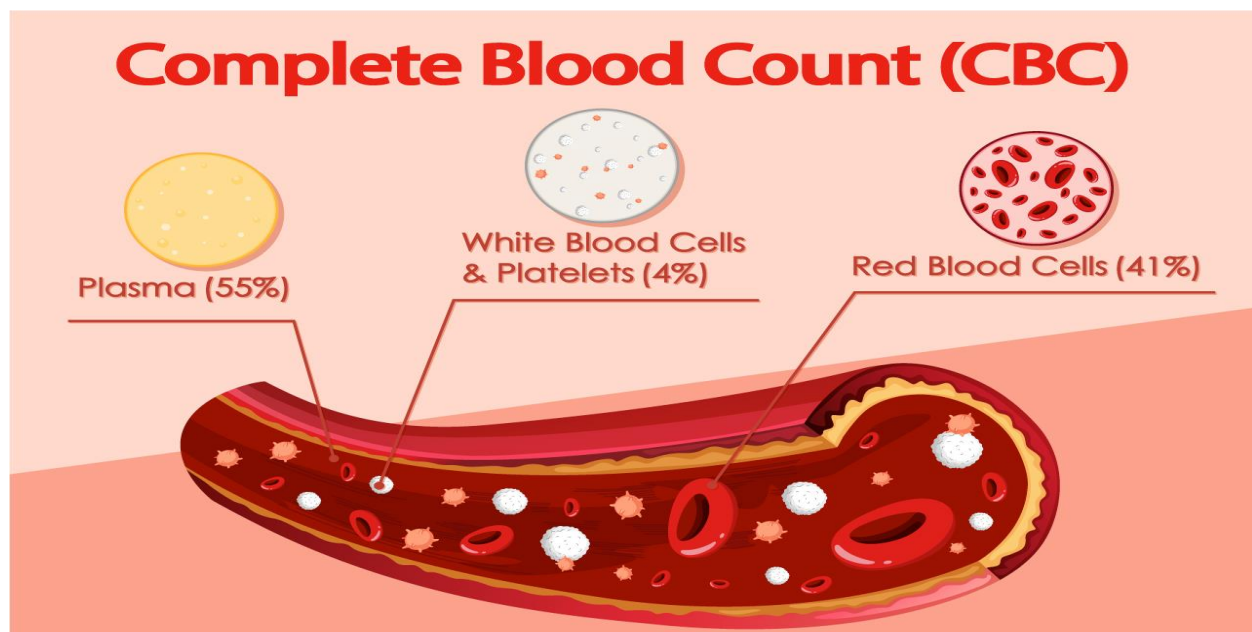


figure 4-CBC

A life span of RBC is around 120 days for a normal individual. A normal RBC count for an adult male is between 4.6×10^{12} and 6.2×10^{12} per liter of blood. Production of red blood cells takes place in the bone marrow from precursor stem cells. Typical red blood cell count (RBC) levels are:

- 4.2 to 5.4 million cells per microliter for women
- 2.6 to 4.8 million cells per microliter for children
- 4.5 to 6.2 million cells per microliter of blood for men.

In diagnosis of several diseases, a major step is automated detection and counting of red blood cells. In the conventional procedure, a haematologist manually counts and classifies the cells with the help of a microscope. The task is to measure the red blood cells and assess the size and shape of red blood cells. But this procedure is time consuming, complex and tedious. Also, the accuracy of recognition is affected by subjective factors like experience and fatigue due to human tiredness. As a solution to this problem, to provide automated, cost-effective and efficient alternatives to detection and counting of RBCs, image processing techniques are used.

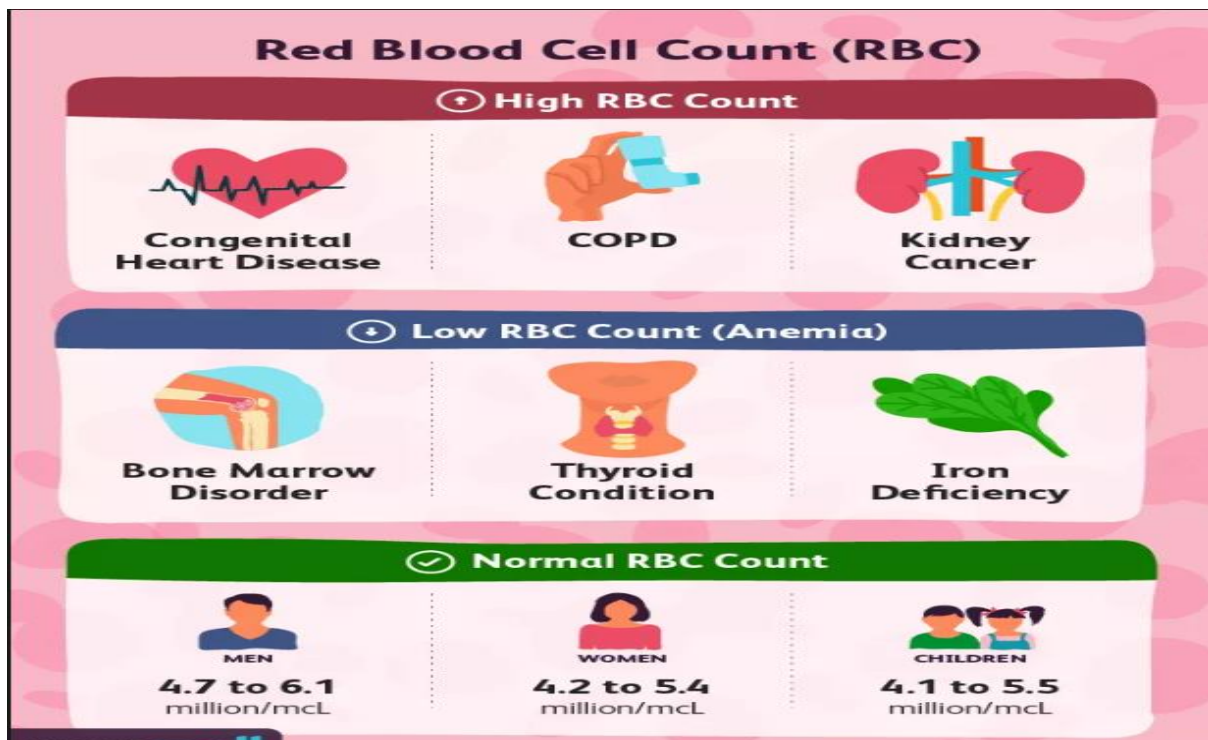


Figure 5-RBC

2.1 Similar Systems :

Mobile based Automated Complete Blood Count (Auto-CBC)

Analysis System from Blood Smeared Image [1]:

a portable mobile-based complete blood count (CBC) analysis framework with the aid of microscope is proposed, and the smartphone camera is mounted to the viewing port of the light microscope by adding a smartphone support. Initially, the blood smeared image is acquired from a light microscope with objective zoom of 100X magnifications view the eyepiece zoom of 10X magnification, then captured by the smartphone camera. Next, the areas constitute to the WBC and RBC are extracted using combination of color space analysis, threshold and Otsu procedure. Then, the number of corresponding cells are counted using topological structural analysis, and the cells in clumped region is estimated using Hough Circle Transform (HCT) procedure. After that, the analysis results are saved in the database, and shown in the user interface of the smartphone application.

White blood cells detection and classification based on regional convolutional neural networks [2]:

The number of WBC types and the total number of WBCs provide important information about our health status. Diseases such as leukemia, AIDS, autoimmune diseases, immune deficiencies, and blood diseases can be diagnosed based on the number of WBCs.

In this study, a computer-aided automated system that can easily identify and locate WBC types in blood images has been proposed. Contrary to traditional methods, convolutional neural networks (CNN), a deep learning architecture, can extract features from a part of an object and perform object recognition. In this case, a CNN-based system shows a higher performance in recognizing partially visible cells for reasons such as overlap or only partial visibility of the image. Therefore, it has been the motivation of this study to increase the performance of existing blood test devices with deep learning methods. Blood cells have been identified and classified by Regional Based Convolutional Neural Networks.

Designed architectures have been trained and tested by combining BCCD data set and LISC dataset. Regional Convolutional Neural Networks (R – CNN) has been used as a methodology. In this way, different cell types within the same image have been classified simultaneously with a detector. While training CNN which is the basis of R – CNN architecture; AlexNet, VGG16, GoogLeNet, ResNet50 architectures have been tested with full learning and transfer learning.

Cell types of Lymphocyte were determined with 99.52% accuracy rate, Monocyte with 98.40% accuracy rate, Basophil with 98.48% accuracy rate, Eosinophil with 96.16% accuracy rate and Neutrophil with 95.04% accuracy rate.

Chapter 3

Analysis and Design

3.1 System Overview

The automated blood count test project is a system designed to help doctors in detecting diseases by using the blood smear image of the patient and by using YOLO (You Only Look Once) object detector and some processes on this image to resize it and extract cell features and classify this cells we can count his red blood cells, white blood cells and platelets and then we store it in a database, so it's easier for doctors to detect the disease and know the health state of the patient and to leave comments on the result and to give the patient the medicine for his illness. Also the system enables the user to show his results via website by logging in with his national ID and view his test results and see the comments of his doctor and know possible medicines for his illness.

3.1.1 System Architecture

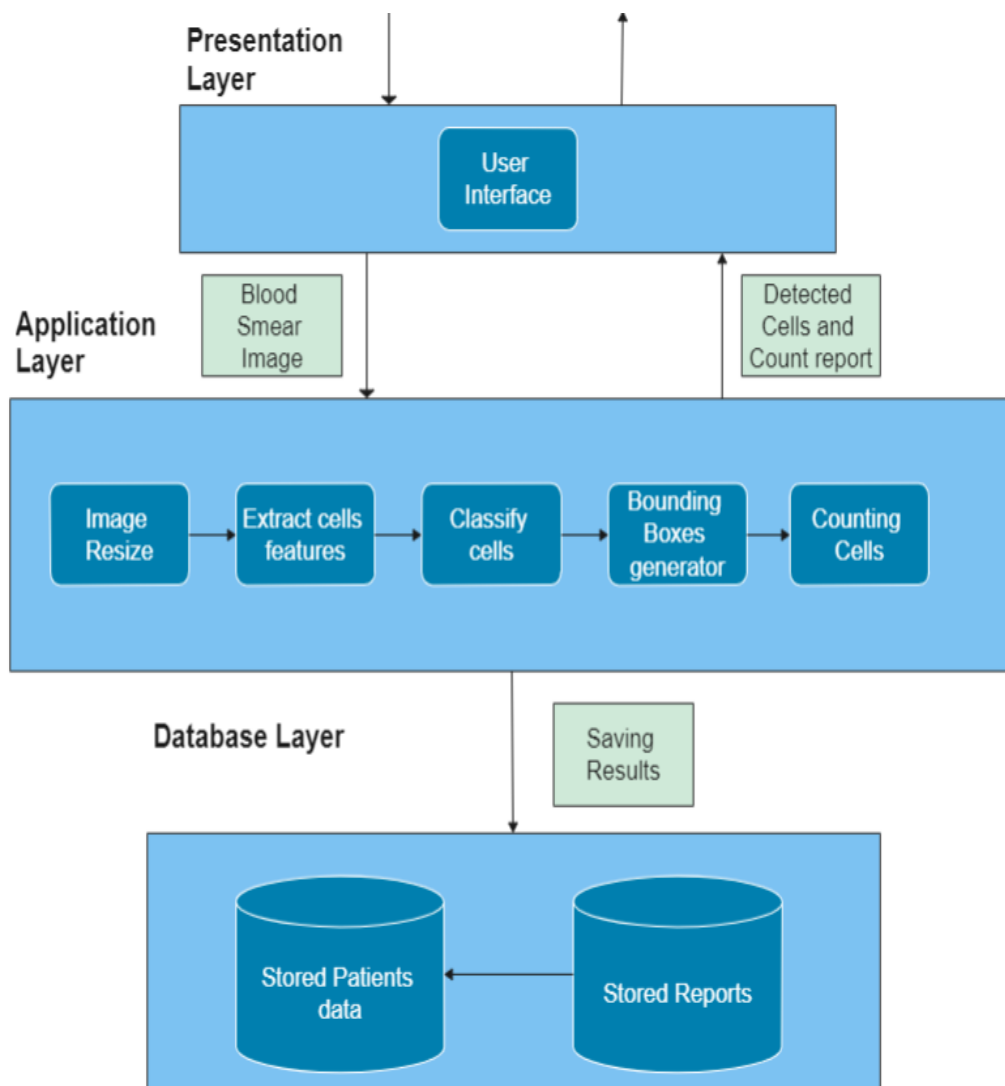


figure 6- system architecture

1-Application layer

Image Resize: Input image is resized to 416x416 to be used by the model.

Extract cells features: The model takes the image and generates a feature map for each class (cell type).

Classify Cells: According to the feature map each cell is assigned to a class(Platelets, RBCs, WBCs).

Generating Bounding Box: An image is created with bounding boxes surrounding each cell with its label.

Counting Cells: The number of repetition of each class is counted.

2-Database Layer

A new report is added to the patient with generated data so we can retrieve his history at any time.

3.1.2 Functional Requirements

Function	Function specifications
Register	<ul style="list-style-type: none">•<u>Description</u>: Through website the user shall fill the fields with required data.•<u>Input</u>: National ID, password and confirm the password.•<u>Source</u>: user.•<u>Output</u>: Confirm the user that the registration is complete.
Login	<ul style="list-style-type: none">•<u>Description</u>: Through the website,it enables user to log in to see his results and doctor comments.•<u>Input</u>: National ID and password.•<u>Source</u>: user.•<u>Output</u>: The system opens the user's account.

<p>Save a history</p>	<p>•<u>Description:</u> The system shall save the history of the user as can return to it anytime.</p> <p>•<u>Input:</u> the user's searches.</p> <p>•<u>Source:</u> system.</p> <p>•<u>Output:</u> The system shall display his history as the searches he did.</p>
<p>Change a password</p>	<p>•<u>Description:</u> The system shall change the password if the user requests that.</p> <p>•<u>Input:</u> The old password and a new password.</p> <p>•<u>Source:</u> user.</p> <p>•<u>Output:</u> The system shall display a confirmation message that the password has changed successfully.</p>
<p>Detect and Count cells</p>	<p>•<u>Description:</u> The system shall count the red,white blood cells and platelets.</p> <p>•<u>Input:</u> The blood smear image.</p> <p>•<u>Source:</u> user.</p> <p>•<u>Output:</u> The system shall display table with counts of RBC,WBC,platelets.</p>

3.1.3 Nonfunctional Requirements

•Performance:

The system must be interactive and the delays involved must be less. In every action-response of the system, there are no immediate delays.

•Safety:

Information transmission should be securely transmitted to the server without any changes.

•Reliability:

As the system provides the right tools for detecting cells, it must be made sure that the system is reliable in its operations and for securing the sensitive details.

•Availability:

If the internet service gets disrupted while sending information to the server, the information can be sent again for verification and System shall be available at all times.

•Security:

The system shall provide a certain type of security to make sure of the validity of data of the user.

•Usability:

The system is easy to handle and navigates in the most expected way with no delays. The system program reacts accordingly and transverses quickly between its states.

3.2 System Users

A. Intended Users:

- Doctors can use it to know the purpose of a disease in humans to diagnose it.
- Researchers can use the data we have collected about cell count and know the diseases that are related to these different counts of cells.
- Biologists can use it to know the relation between the number of cells and the affected functions of the body and the resulting diseases.

B. User Characteristics

- The user should have the basic knowledge of operating the internet and to have access to it.
- The user should have the basic biological information especially in proteins.

3.3 System Analysis & Design

3.3.1 Use Case Diagram

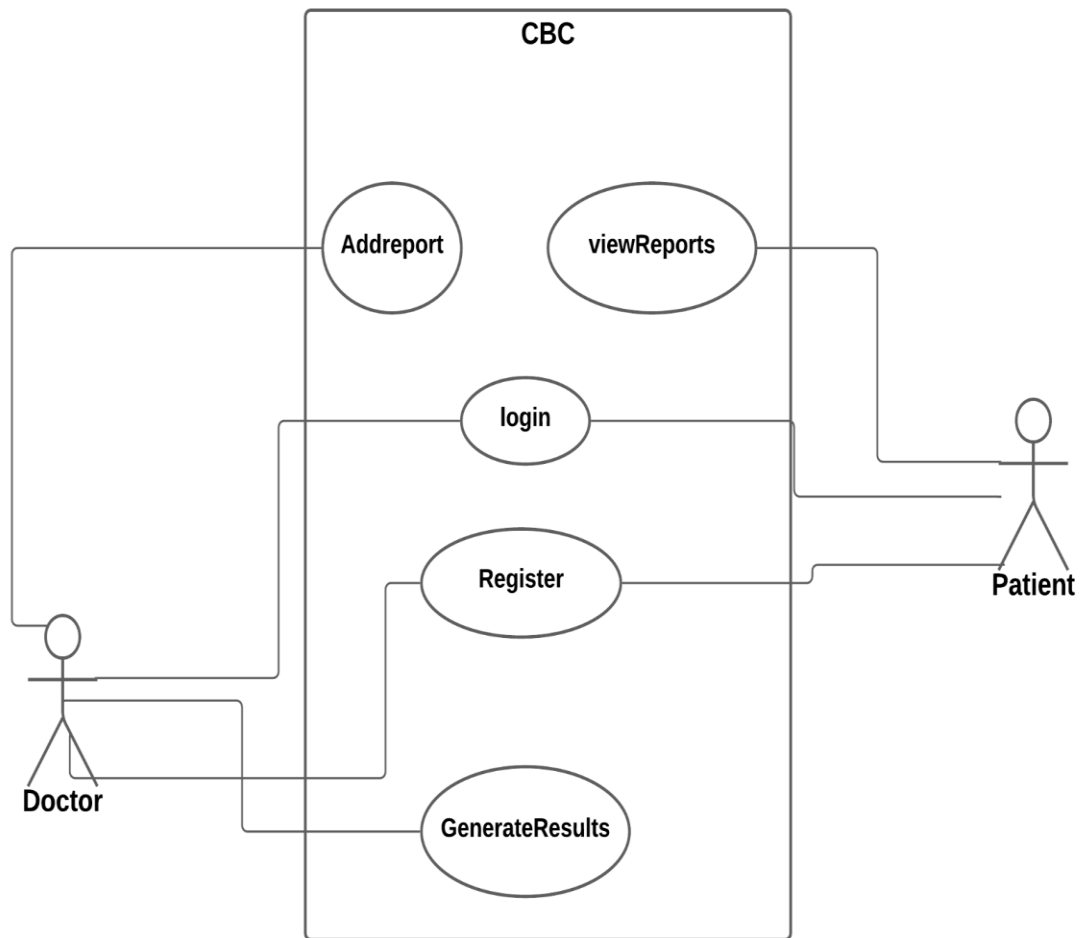


figure 7-use case

3.3.1 Use Case Diagram

Use case function	Description
Login	If the user has an account at the system he simply logs in with a National ID and password.
Register	if the user is new to the system he could simply create an account with his data and then login.
Generate result	The user (doctor) enters the blood smear image to the desktop application
Add reports	After the desktop application generates the result with counts of RBC,WBC,platelets cells the doctor uses these results to fill a report form through the website and add comments then he submits the report so the patient can view it.
View reports	After the user (patient) logs in through the website he could view his report with results and comments from his doctor.

3.3.2 Class Diagram

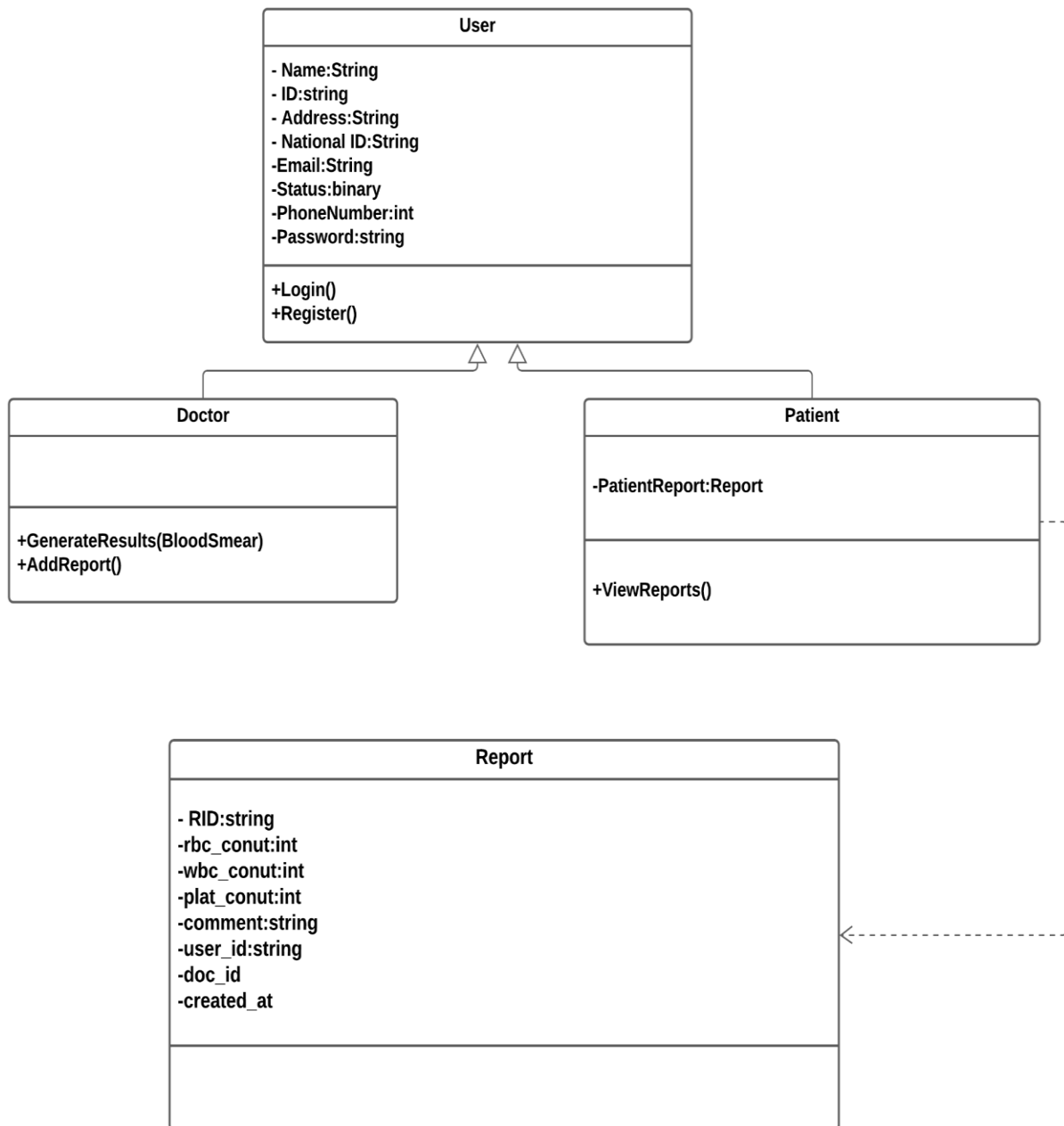


figure 8- class diagram

3.3.2 Class Diagram

Main Classes description :

1-User:this is the parent class for patient and doctor subclasses which has attributes like name,ID,phone Number,address,...etc.Patient and doctor classes inherit these attributes from it.

2-Report:This class is responsible for saving the results and history of the patients like the time and the lab branch and the results.

3-Doctor:the user who will use the desktop application and can also comment on the patient's results, this class is also responsible for generating the results by entering the blood smear image to the desktop application and then adding a report filled with results info.

4-Patient: the user who will use the website to view his results and his doctor's comments.

3.3.3.1 Sequence Diagrams

Register Sequence Diagram

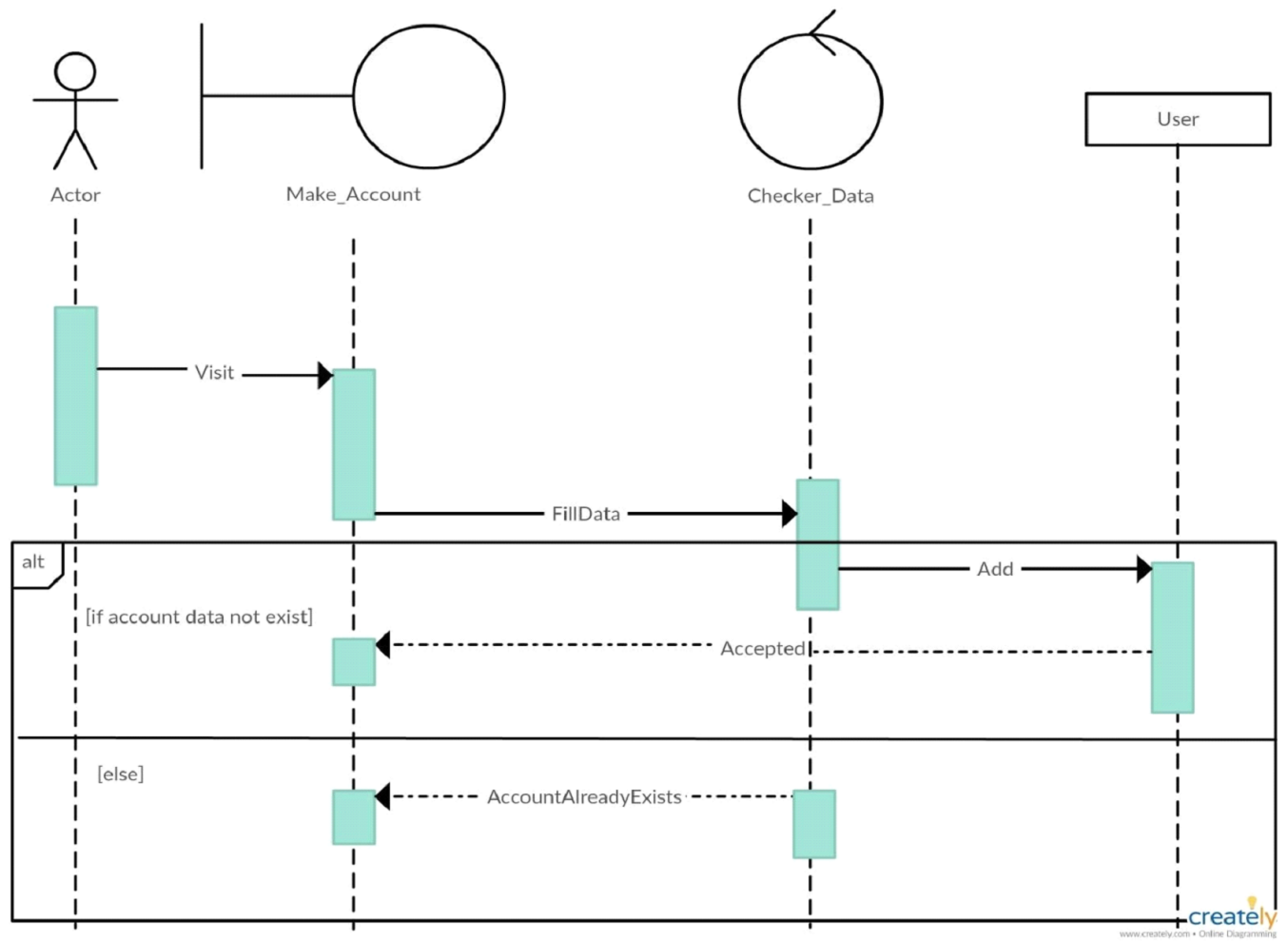


figure 9-sequence diagram 1

3.3.3.2 Sequence Diagrams

Login Sequence Diagram

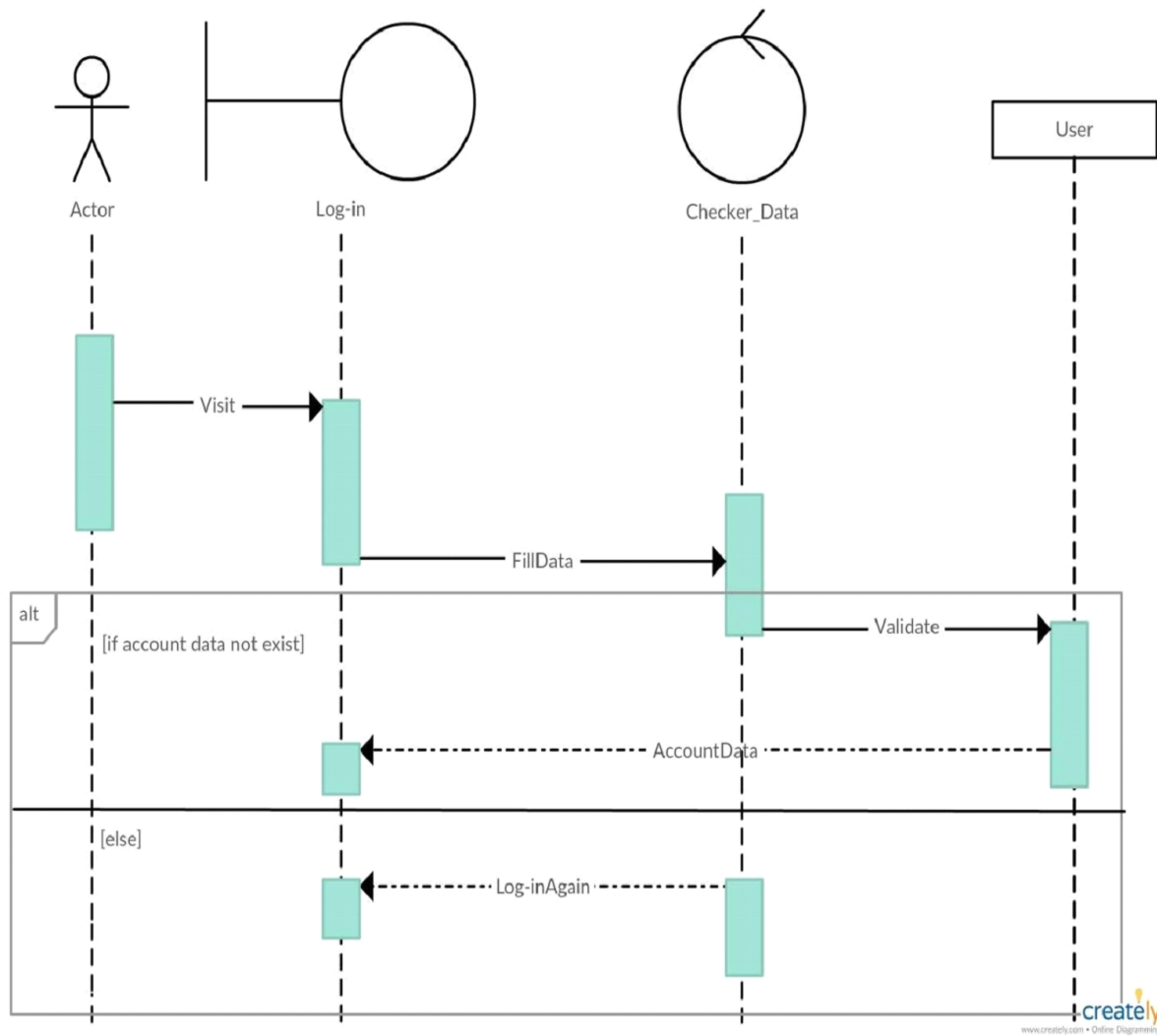


figure 10-sequence diagram 2

3.3.3.3 Sequence Diagrams

View reports Sequence Diagram

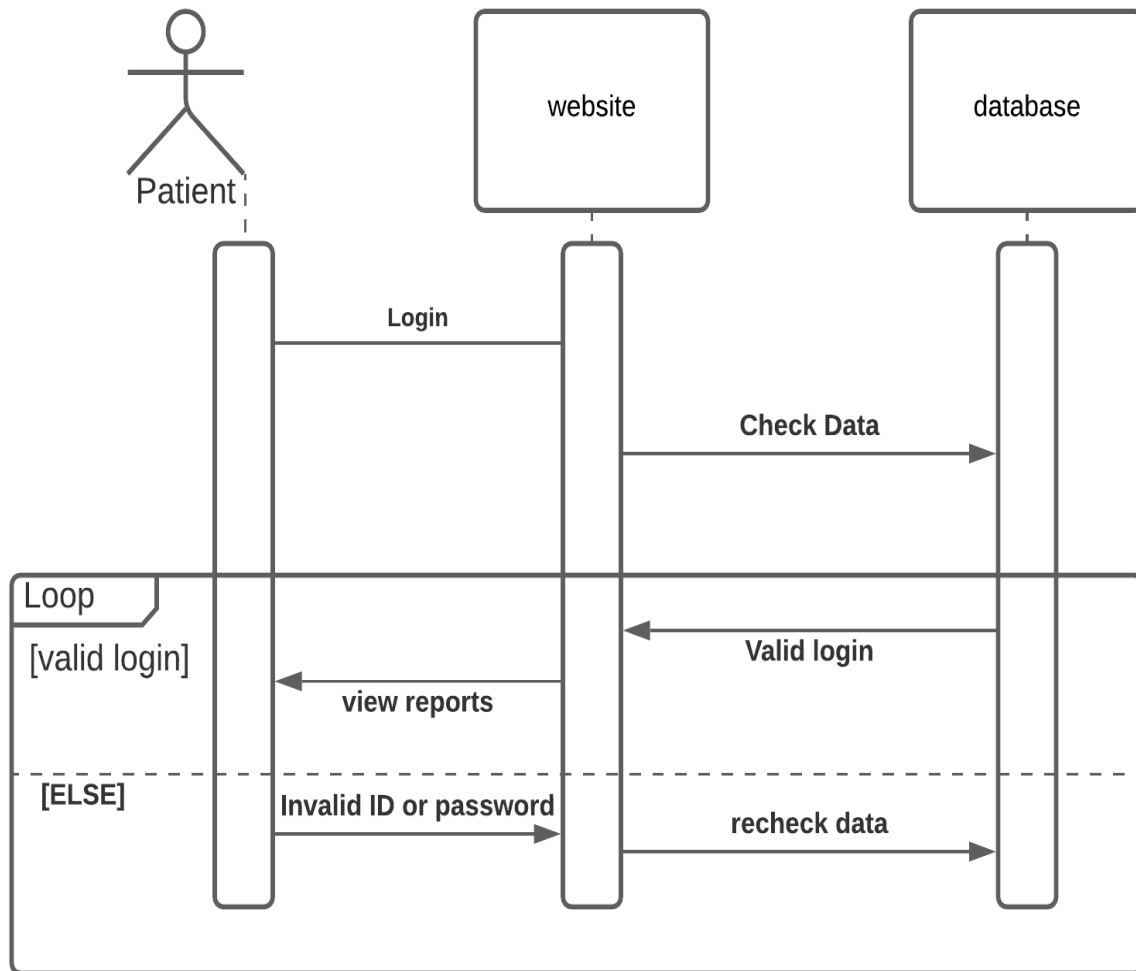


figure 11-sequence diagram 3

3.3.3.4 Sequence Diagrams

Add reports Sequence Diagram

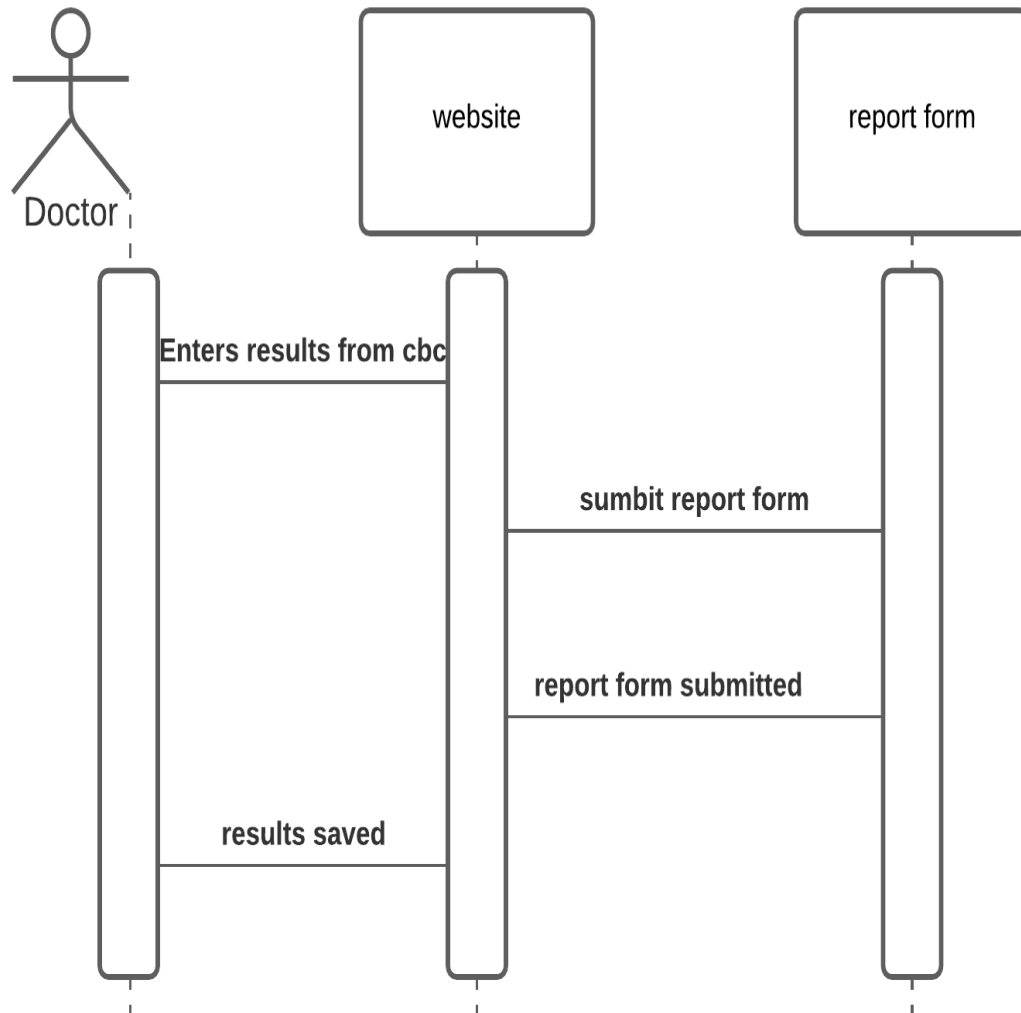


figure 12-sequence diagram 4

3.3.3.5 Sequence Diagrams

Generate reports Sequence Diagram

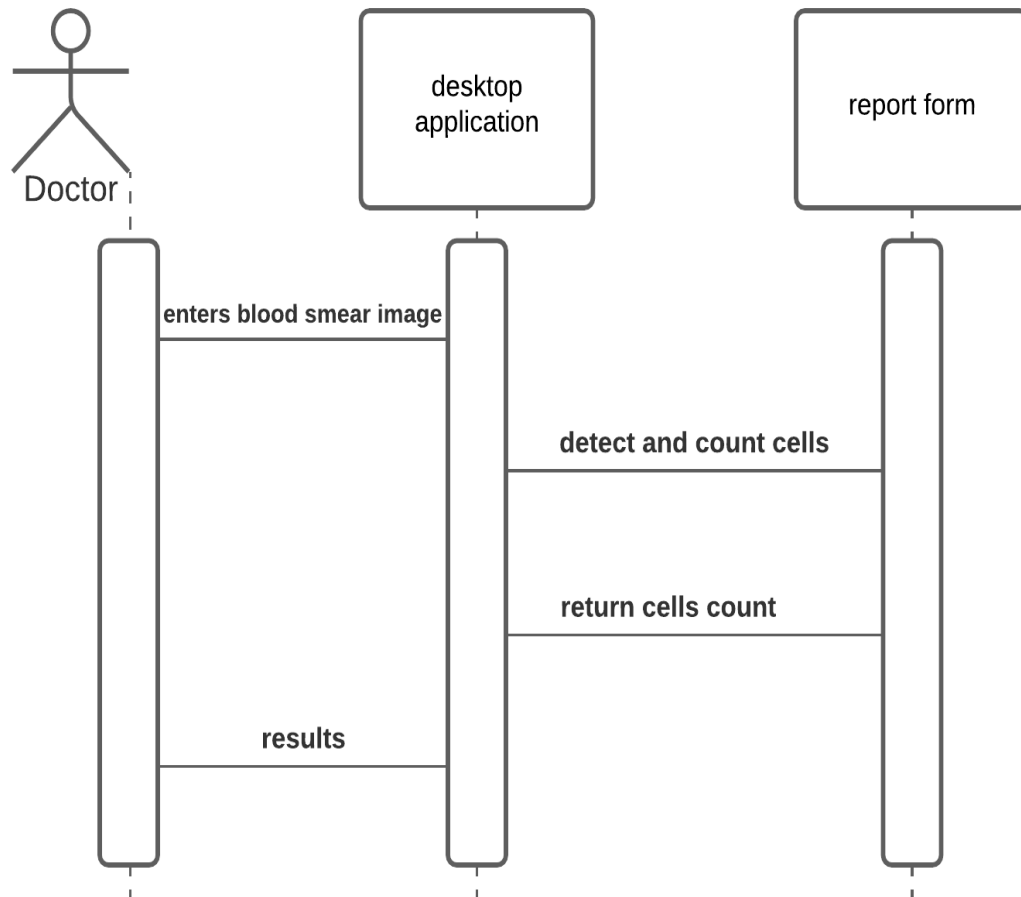


figure 13-sequence diagram 5

3.3.4 Database Diagram

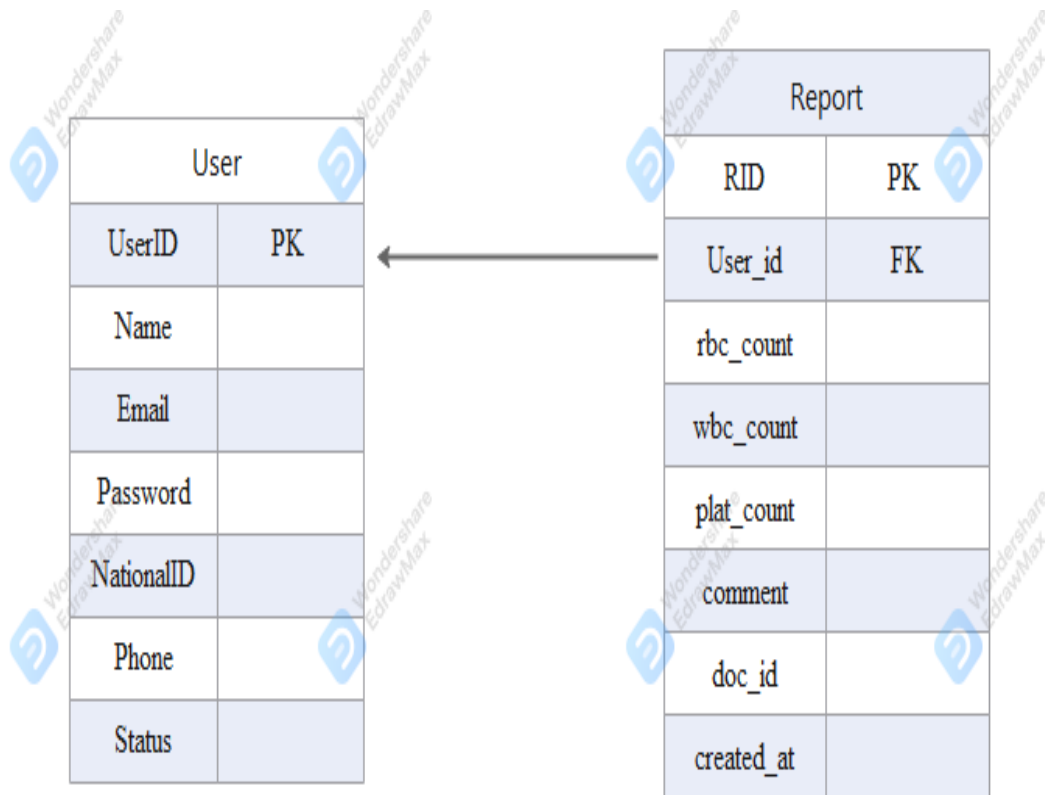


figure 14-Database Diagram

Table name	Description
User	A table for users which has an Id as a primary key and has other attributes like name,email,...etc of the desktop users and website users.
Report	A table which has a Report_id as a primary key and a user_id as a foreign key to access data from users table,also it contains other attributes for the report formation like wbc,rbc and platelets counts and doctor's comments.

Chapter 4

Implementation

4.1 Dataset Preparation:

We obtained blood cells dataset from the public BCCD dataset <https://public.roboflow.com/object-detection/bccd/3>, the dataset contains discrete data that consists of three different values which are platelets, WBCs and RBCs, it consists of 364 blood smear images and contains 4888 labels for different blood cells. We applied auto-orientation to the images then resized all the images to 416x416 so it can be used with our model and divided the dataset into 80% training set, 10% validation set and 10% testing set.

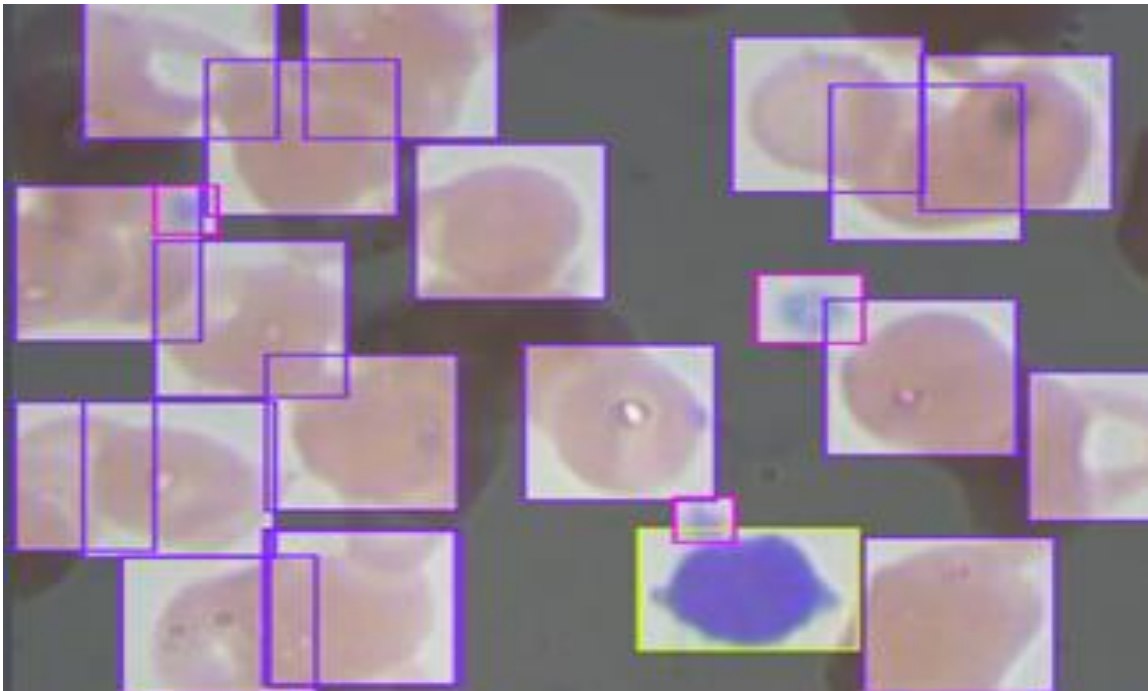


figure 15-labeled blood smear from data set

4.2 Training Model using google colab

Steps

- Images are extracted from roboflow.
- The model uses training set images for training

- Calculate accuracy while training using mean average precision(mAP)
- Create weights file for the model for further predictions

Flowchart

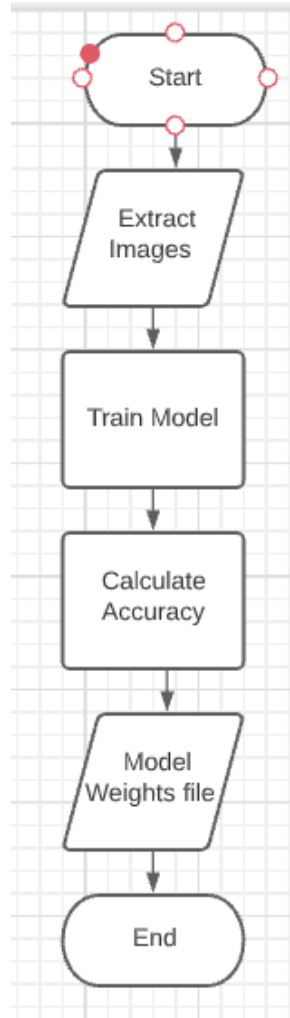


figure 16-flowchart 1

```
[ ] Loaded: 0.000049 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1322: 13.435305, 11.815582 avg loss, 0.001000 rate, 6.769212 seconds, 47592 images, 13.615391 hours left
Loaded: 0.000058 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1323: 11.976979, 11.831722 avg loss, 0.001000 rate, 6.770902 seconds, 47628 images, 13.567200 hours left
Loaded: 0.000071 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1324: 12.881464, 11.936696 avg loss, 0.001000 rate, 6.776662 seconds, 47664 images, 13.519495 hours left
Loaded: 0.000059 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1325: 12.523991, 11.995425 avg loss, 0.001000 rate, 6.817321 seconds, 47700 images, 13.472322 hours left
Loaded: 0.000050 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1326: 12.148710, 12.010754 avg loss, 0.001000 rate, 6.771620 seconds, 47736 images, 13.426130 hours left
Loaded: 0.000039 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1327: 12.347569, 12.044436 avg loss, 0.001000 rate, 6.792248 seconds, 47772 images, 13.379788 hours left
Loaded: 0.000051 seconds

(next mAP calculation at 1400 iterations)
Last accuracy mAP@0.5 = 88.10 %, best = 88.10 %
1328: 10.826280, 11.922620 avg loss, 0.001000 rate, 6.739977 seconds, 47808 images, 13.334158 hours left
Loaded: 0.000053 seconds
```

Figure 17-Training phase

- Object detectors are divided into two categories.
 - Two-stage detectors that decouple the task of object localization and classification for each bounding box.
 - One-stage detectors make the predictions for object localization and classification at the same time.

- Yolo(You only look once) : It is a network for object detection.
- Yolo uses the darknet framework.
- Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.
- YOLO is a real time object detector which gives high accuracy in short processing time.

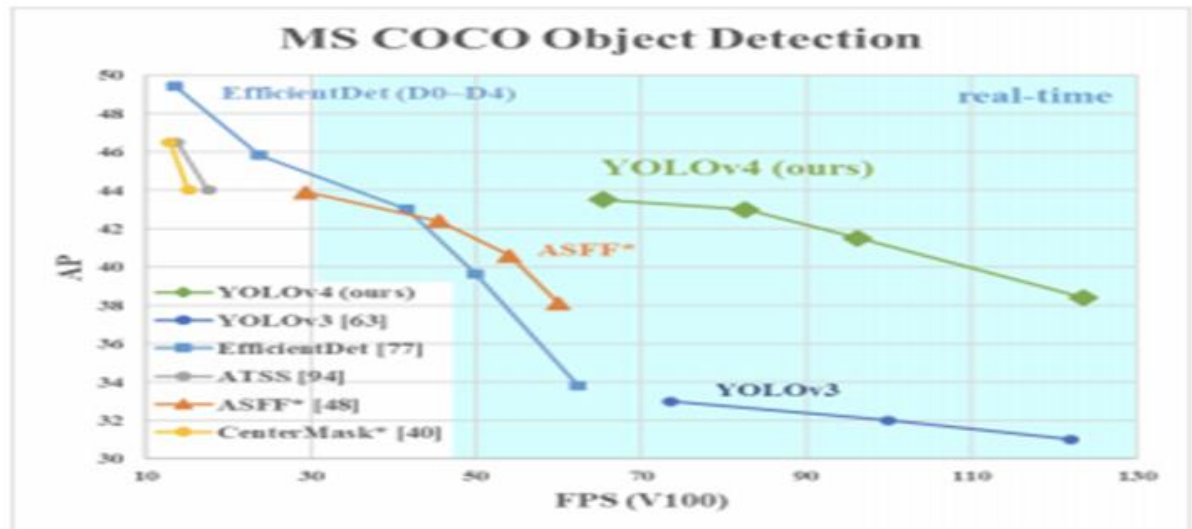


figure 18-ms coco object detection

- How does YOLO work?
 - YOLO divides the input image into an $S \times S$ grid, each grid cell predicts only one object.
 - Each grid cell predicts a fixed number of boundary boxes.
 - If the centre of the object falls into a grid cell that the grid cell is responsible for detecting that object.
 - Non-maximal suppression eliminates unwanted bounding boxes, the boxes that have the highest confidence remain.
 - The Intersection over Union (IoU): the amount of predicted bounding box that overlaps with the ground truth bounding box divided by the total area of both bounding boxes.
 - Precision measures how accurate your predictions are.
 - Precision = $TP / (TP + FP)$

- YOLO Output

- P_c : defines whether an object is present in the grid or not (it is the probability).
- B_x, B_y : X and Y coordinates for the center point of the bounding box.
- B_w, B_h : The width and height of the bounding box.
- C_1, C_2, \dots, C_n : If class n is detected then this flag will be 1 otherwise it will be 0.
- Also there is a confidence score for each bounding box.

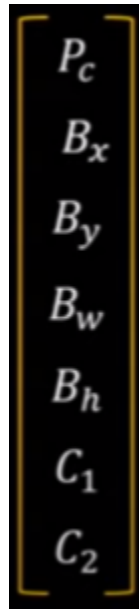


figure 19-YOLO Output

- About the model

- Yolov4
 - This version employs "**Bag of Freebies**": a collection of augmentation techniques that have been shown to improve performance.
 - deploys strategies called a "**Bag of Specials**": they add marginal increases to inference time but significantly increase performance.

4.3 Developing desktop application using python

During this phase we used tkinter for building the GUI of the application.

Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

This Python framework provides an interface to the Tk toolkit and works as a thin object-oriented layer on top of Tk. The Tk toolkit is a cross-platform collection of 'graphical control elements', aka widgets, for building application interfaces.

4.3.1 Main

The blood smear image of the patient is received from the laboratory and passed to our program to analyze it using the YOLO model.

Steps:

- Browse Blood Smear Image
- Check whether an image is selected or not
- if an image is selected, send it to detect function
- else ask the user to choose an image

4.3.2 Detect

It receives the image from the main function then analyzes it using the trained YOLO model and results in an image containing bounding boxes around the cells and counters for each type of cells.

Steps:

- Receive the blood smear image.
- Detect blood cells
- Draw bounding boxes around cells
- Count each class

Flowchart for the process

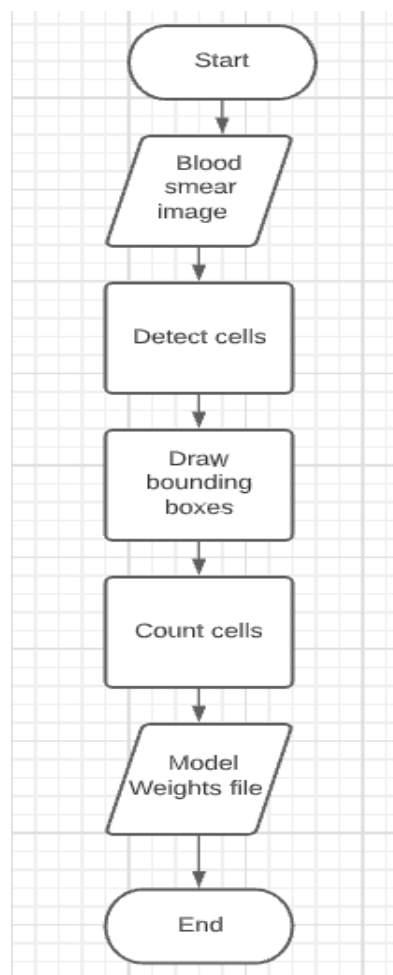


figure 20- flowchart 2

- Results

Blood smear image with bounding box around each cell and a counter for each cell type

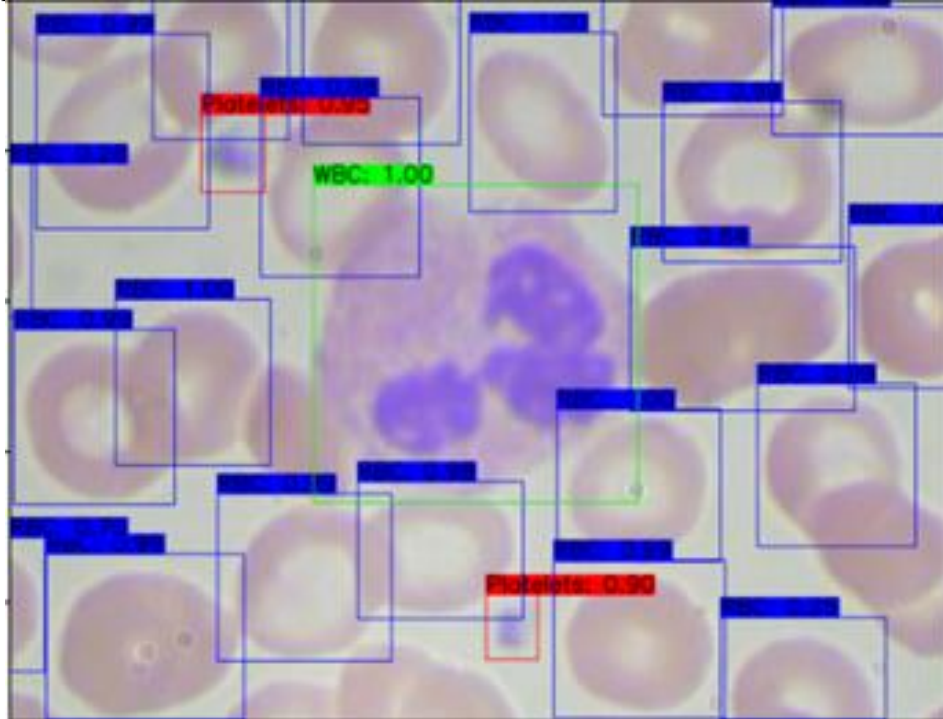


figure 21-blood smear after detection

4.4 Database using MySQL

4.4.1 About MySQL

- MySQL is widely used in the web development field.
- It is a widely used open source RDBMS.
- Its primary database model is relational DBMS
- It's fantastic for those who are much more advanced. It is extremely flexible and there is a ton of support.
- Pros
 - Speedy System
 - Independent Platform
 - Mass Data Manipulation
 - Above Average Support

- Cons
 - Simply doesn't scale as well as commercial databases.
 - The stored procedure and programming language is too limited compared to TSQL or PL/SQL.
 - It lacks some of the higher end features of commercial databases such as flashback recovery, updateable views, etc.
- Supported programming languages
 - C, C#, C++
 - Ada
 - Java
 - Javascript (Node.js)
 - PHP
 - Python
 - Ruby
 - Delphi
 - Perl

- Why MySQL

We can start working on MySQL very quickly, as it takes a few minutes for the download to install. Regardless of the platform, we can use MySQL quickly. Features like self-management capabilities like auto restart, space expansion and automatic configuration changes for ease of management are available in MySQL. Data security is very high in MySQL.

MySQL transactions work as a single unit, which means unless all the operational stages are completed successfully, they will not be cleared. If any stage fails, it will revert to the previous stage. This helps in the hassle-free online money transactions.

- Description

We used phpmyadmin as our interface between MySQL and the development environment of ruby on rails.

Our database consists of two tables, one for the users and the other one for the reports and they are connected through a user id which is foreign key in the reports table.

The user table user's data whether he/she is a patient or doctor and this is determined using the status variable which has two different values, 0 for patient and 1 for doctor, the primary key in the users table is the id and the table has national_id which is a unique variable for each user, also the email variable is unique too.

The other table is the reports table and it is connected to the users table by user_id, this table holds data about the report generated by the desktop application and is assigned to a patient the relation between the two tables is 1 to many relation as the user can have many reports but the report only has one patient at a time.

- phpmyadmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id 	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 3	Email	varchar(255)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 4	password_digest	varchar(255)	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 5	phone	int(11)			Yes	NULL		
<input type="checkbox"/> 6	national_id	bigint(20)			Yes	NULL		
<input type="checkbox"/> 7	status	blob			Yes	NULL		
<input type="checkbox"/> 8	created_at	datetime(6)			No	None		
<input type="checkbox"/> 9	updated_at	datetime(6)			No	None		

figure 22-database users table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id 	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	rbc_count	int(11)			Yes	NULL		
<input type="checkbox"/> 3	wbc_count	int(11)			Yes	NULL		
<input type="checkbox"/> 4	plat_count	int(11)			Yes	NULL		
<input type="checkbox"/> 5	comment	text	latin1_swedish_ci		Yes	NULL		
<input type="checkbox"/> 6	user_id 	bigint(20)			Yes	NULL		
<input type="checkbox"/> 7	doc_id	int(11)			Yes	NULL		
<input type="checkbox"/> 8	created_at	datetime(6)			No	None		
<input type="checkbox"/> 9	updated_at	datetime(6)			No	None		

figure 23-database reports table

4.5 Developing website using ruby on rails

4.5.1 MVC

The Model-View-Controller (MVC) architectural pattern separates an application into three main groups of components: Models, Views, and Controllers. This pattern helps to achieve separation of concerns. Using this pattern, user requests are routed to a Controller which is responsible for working with the Model to perform user actions and/or retrieve results of queries. The Controller chooses the View to display to the user, and provides it with any Model data it requires.

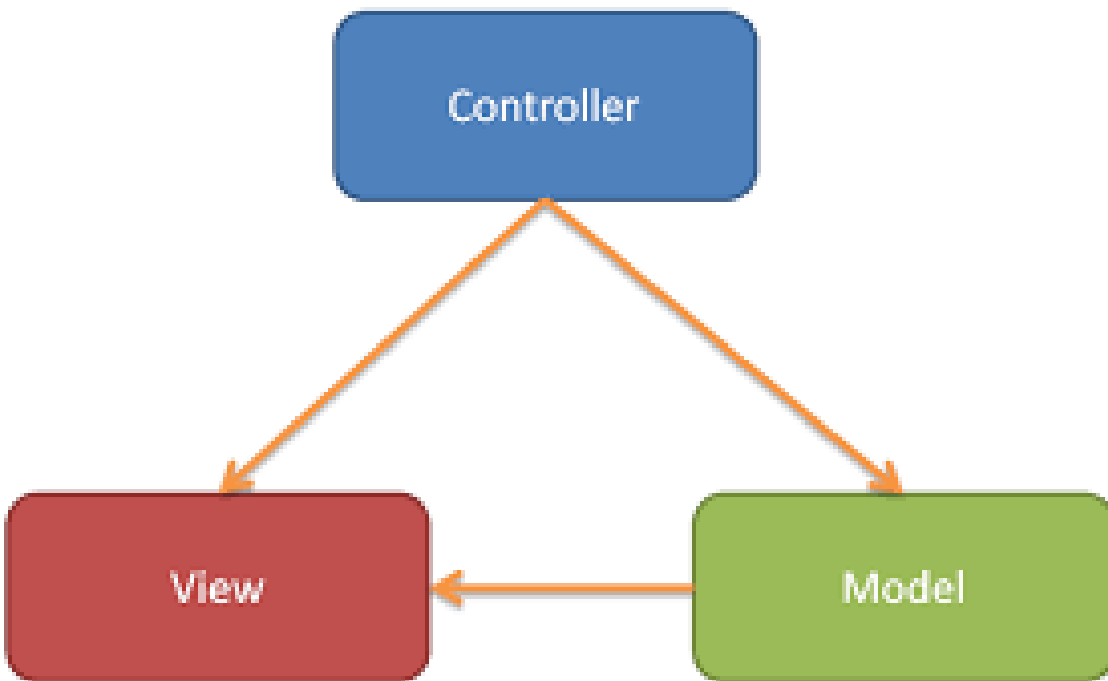


figure 24-MVC

4.5.2 Ruby on rails MVC framework components

- The Model in a MVC application represents the state of the application and any business logic or operations that should be performed by it.
- Views are responsible for presenting content through the user interface. They use HTML, CSS and JavaScript .
- Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render.
- Rails handles those concepts with modules which are Active record for the Model and Action controller for the controller and
- Active Record: Active Record is the M in MVC - the model - which is the layer of the system responsible for representing business data and logic. Active Record facilitates the creation and use of business objects whose data requires persistent storage to a database. It is an implementation of the Active Record pattern which itself is a description of an Object Relational Mapping system.
 - Object Relational Mapping, commonly referred to as its abbreviation ORM, is a technique that connects the rich objects of an application to tables in a relational database management system. Using ORM, the properties and relationships of the objects in an application can be easily stored and retrieved from a database without writing SQL statements directly and with less overall database access code.
- Action Controller:
 - Action Controller is the C in MVC. After the router has determined which controller to use for a request, the controller is responsible for making sense of the request, and producing the appropriate output. Luckily, Action Controller does most of the groundwork for you and uses smart conventions to make this as straightforward as possible.
 - For most conventional RESTful applications, the controller will receive the request (this is invisible to you as the developer), fetch or save data from a model, and use a view to create HTML output. If your controller needs to do things a little differently, that's not a problem, this is just the most common way for a controller to work.

- A controller can thus be thought of as a middleman between models and views. It makes the model data available to the view so it can display that data to the user, and it saves or updates user data to the model
- The MVC pattern helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. This separation helps you manage complexity when you build an application, because it enables you to focus on one aspect of the implementation at a time. So, you can focus on the view without depending on the business logic

4.5.2 Ruby on rails

Ruby on Rails, or **Rails**, is a server-side web application framework written in Ruby under the MIT License. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer and HTML, CSS and JavaScript for user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

- Why ruby on rails is preferred:
 - The process of programming is much faster than with other frameworks and languages, partly because of the object-oriented nature of Ruby and the vast collection of open source code available within the Rails community.
 - The Rails conventions also make it easy for developers to move between different Rails projects, as each project will tend to follow the same structure and coding practices
 - Rails is good for rapid application development (RAD), as the framework makes it easy to accommodate changes.
 - Ruby code is very readable and mostly self-documenting. This increases productivity, as there is less need to write out separate

documentation, making it easier for other developers to pick up existing projects.

- Rails has developed a strong focus on testing, and has good testing frameworks.
- Rails and most of its libraries are open source, so unlike other commercial development frameworks there are no licensing costs involved.

Rails architecture

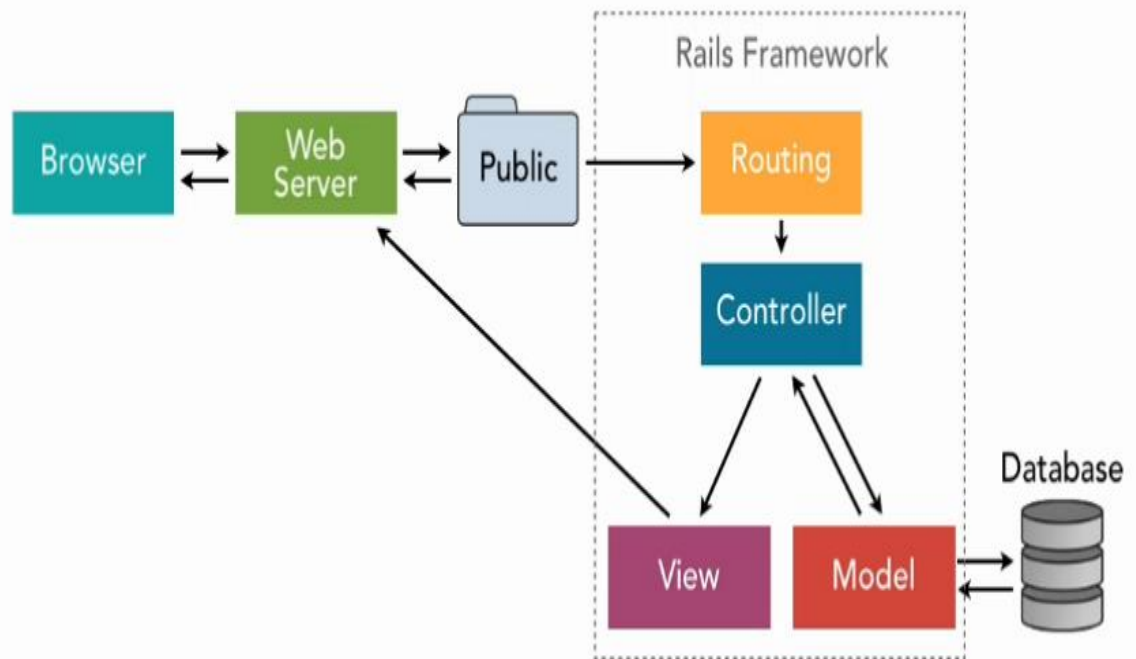


figure 25- Rails architecture

- Ruby on rails Data flow:
 - The user performs an action on the interface.
 - The controller takes the input event.
 - The controller notifies the user action to the model, which may involve a change of state of the model.
 - It generates a new view. The view takes the data model.
 - The user interface waits for another user interaction, which starts a new cycle

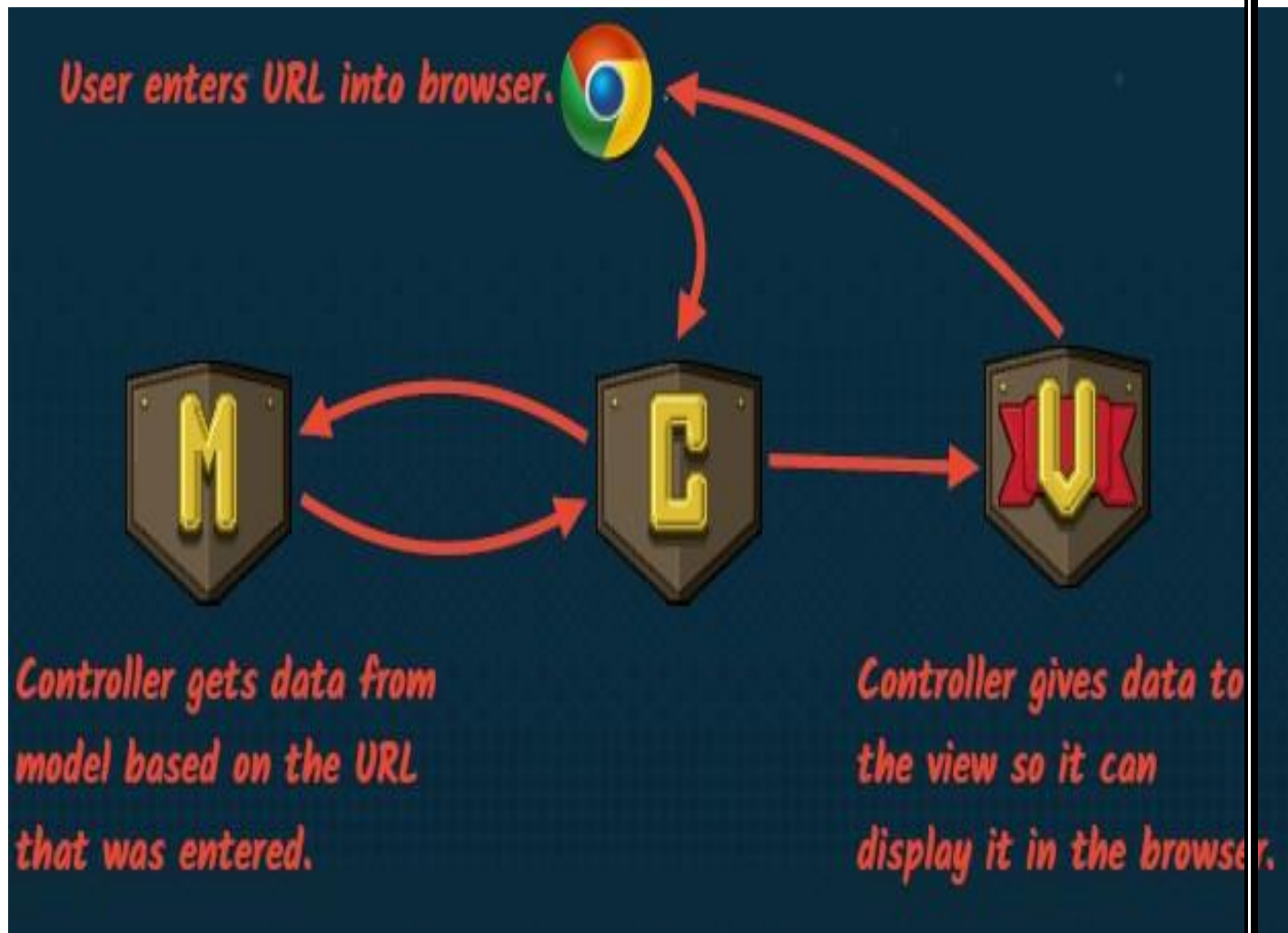


figure 26-data flow

- Controller Methods
 - GET Method: is used to request data from a specified resource, and is one of the most common HTTP methods.
 - POST Method: is used to send data to a server to create or update.

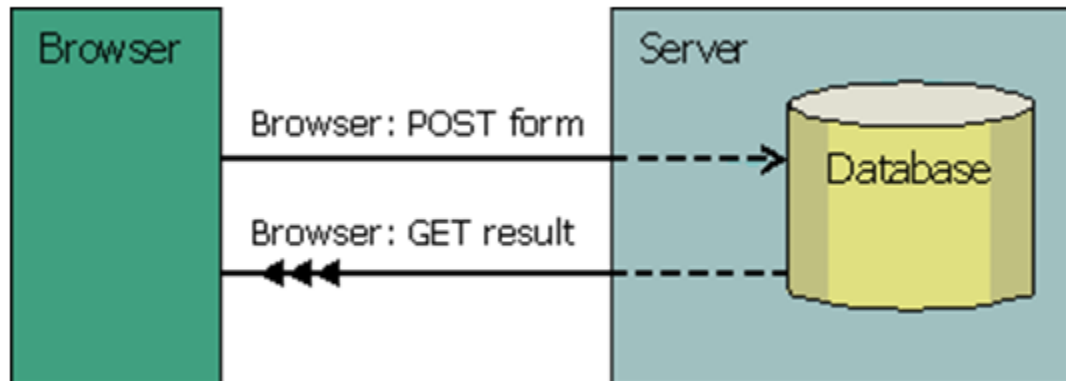


figure 27-Controller method

- Steps:
 - If the user has an account he can login.
 - else he must register first.
 - After login we check whether he is a patient or a doctor.
 - If the user is patient then he can view all his reports.
 - If the user is a doctor he can create a report for the patient.

- Flowchart

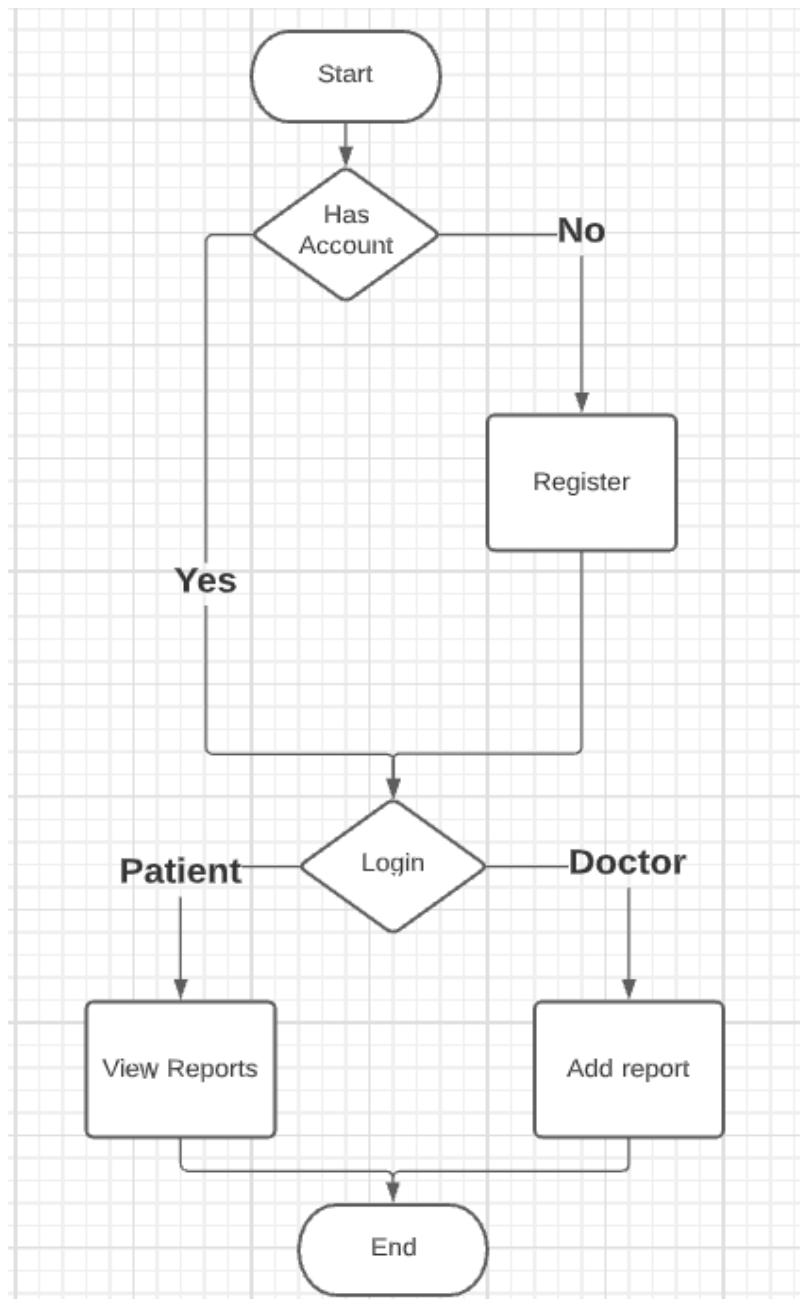


figure 28-Flowchart 3

Function	Description
Register Function	The function receives user data from the HTML form then the data is checked for validation and then a new record is created in the database.
Login function	This function searches for the user entered national ID and checks whether it is correct or not and if correct it checks whether the user is a doctor or patient.
Reports function	This function fetches all reports that are related to the patient ID and displays them in ascending order according to the date of creation
Add Report	This function is related to the doctor and it is used to add a report which is received from the desktop application.

4.5.3 Docker

- Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments.
- Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.
- Docker also refers to Docker, Inc. (link resides outside IBM), the company that sells the commercial version of Docker, and to the Docker open source project (link resides outside IBM), to which Docker, Inc. and many other organizations and individuals contribute.
- We used docker to build the project environment with a single command.

- To build the environment we need two files, the first one is the dockerfile: Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession. The second one is the docker compose file.
- Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration. To learn more about all the features of Compose, see the list of features.
- Compose works in all environments: production, staging, development, testing, as well as CI workflows. You can learn more about each case in Common Use Cases.
- Using Compose is basically a three-step process:
 1. Define your app's environment with a Dockerfile so it can be reproduced anywhere.
 2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
 3. Run docker compose up and the Docker compose command starts and runs your entire app. You can alternatively run docker-compose up using the docker-compose binary.

Inner-Loop development workflow for Docker apps

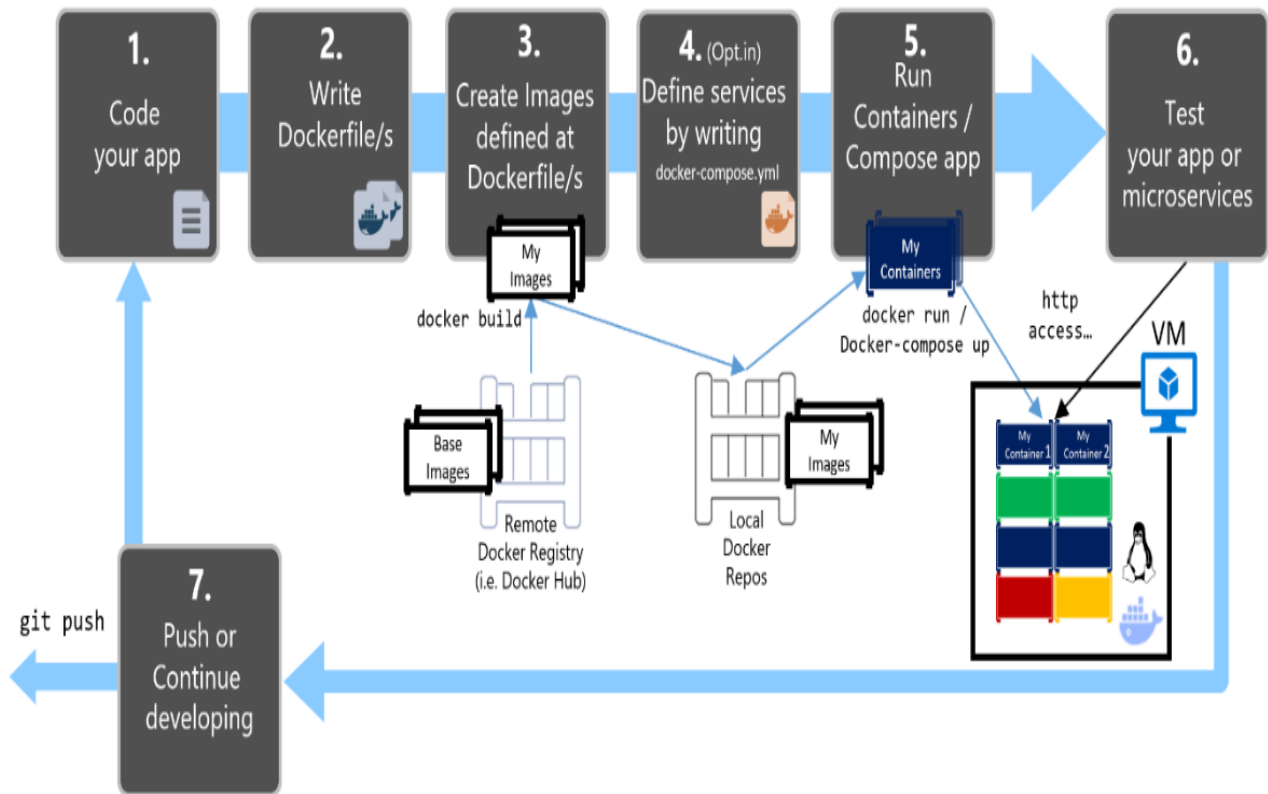


Figure 29-Docker Steps

As shown in figure 27, it describes the *inner-loop* development workflow for Docker container-based applications. The inner-loop workflow means it's not considering the broader DevOps workflow, which can include up to production deployment, and just focuses on the development work done on the developer's computer. The initial steps to set up the environment aren't included, since those steps are done only once.

Chapter 5

User Manual

5.1 Desktop application Data entry form

- This is the first form the user interacts with to analyze the blood smear image.

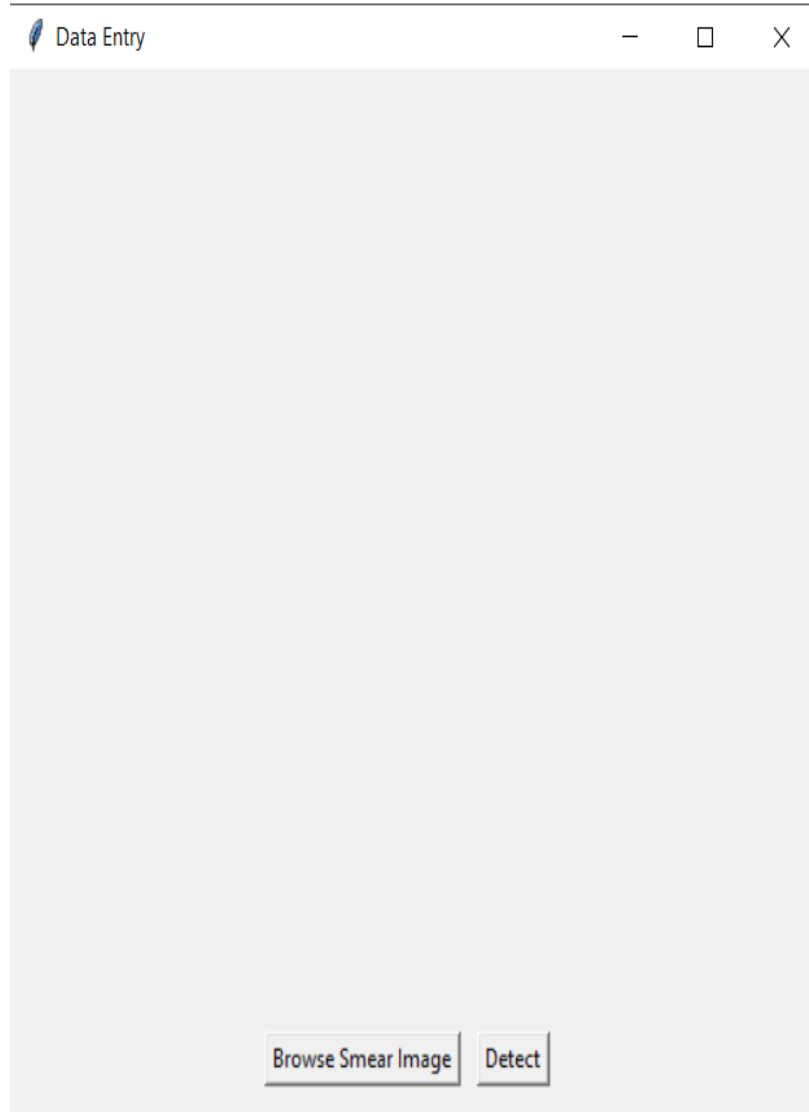


Figure 30-Data entry form

- If the user clicks detect without choosing an image a warning message box will pop up telling him to enter an image.

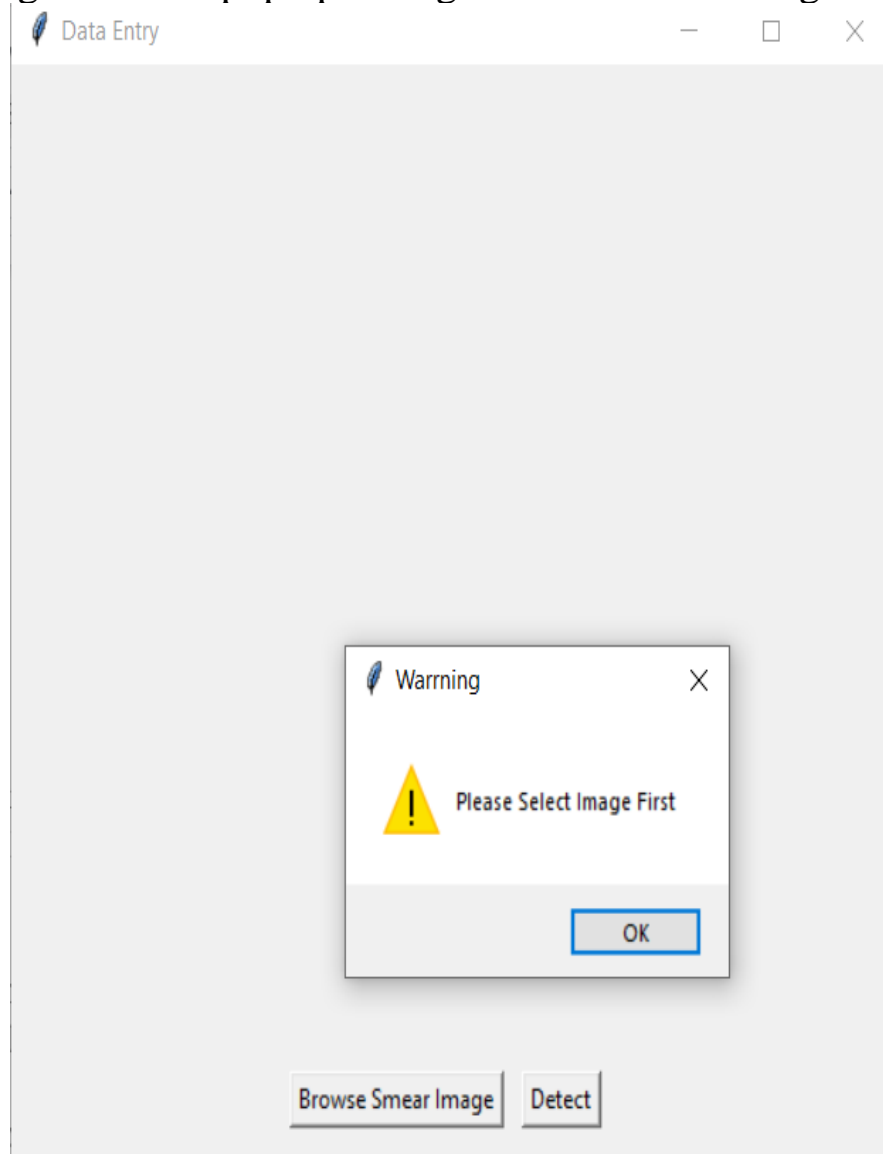


Figure 31-error

The user clicks on “Browse Smear Image” button to choose an image from the local storage, this image is delivered to the user from the laboratory.

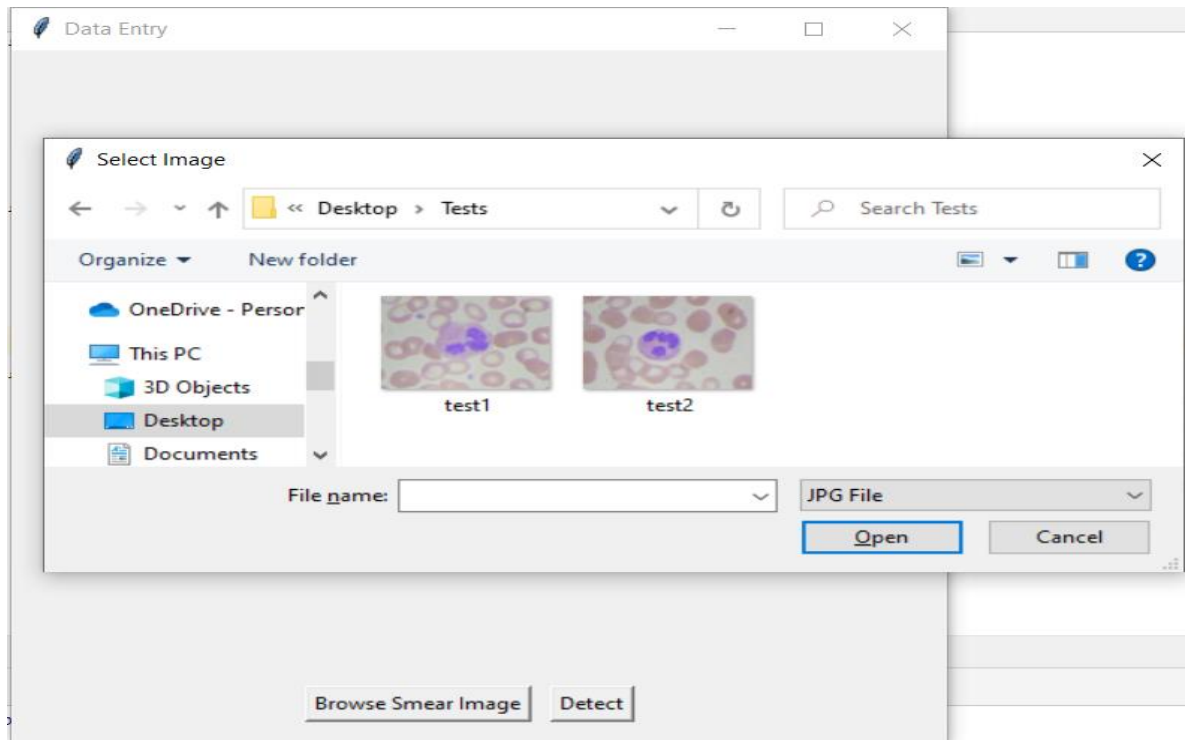


Figure 32-browse image

After choosing the image the user presses the “detect” button and then he is transferred to the output form.

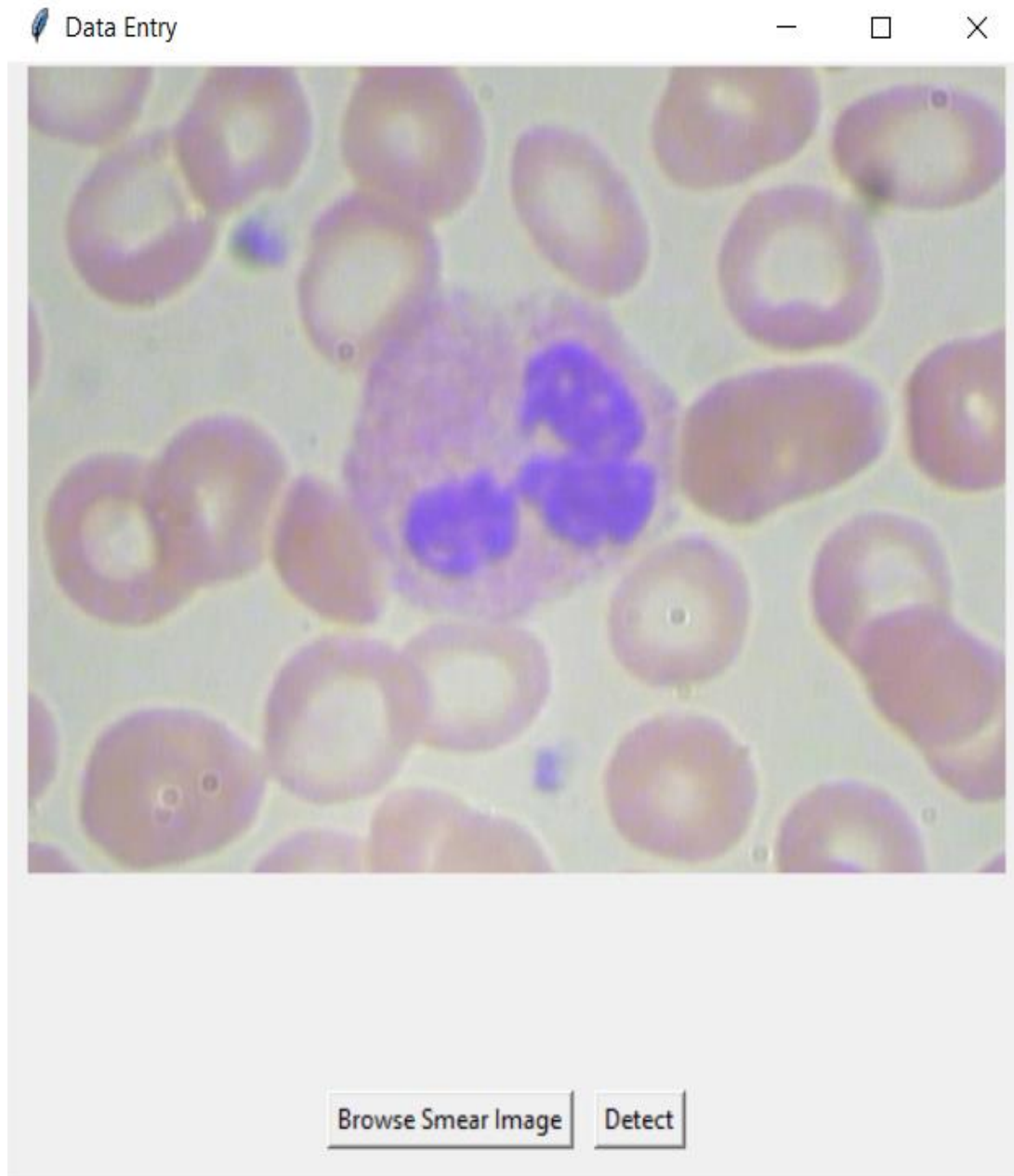


Figure 33-before detection

5.2 Desktop application report form

- This form is the final report which contains the entered image with bounding boxes surrounding the cells and the number of repetition of each type of cells.

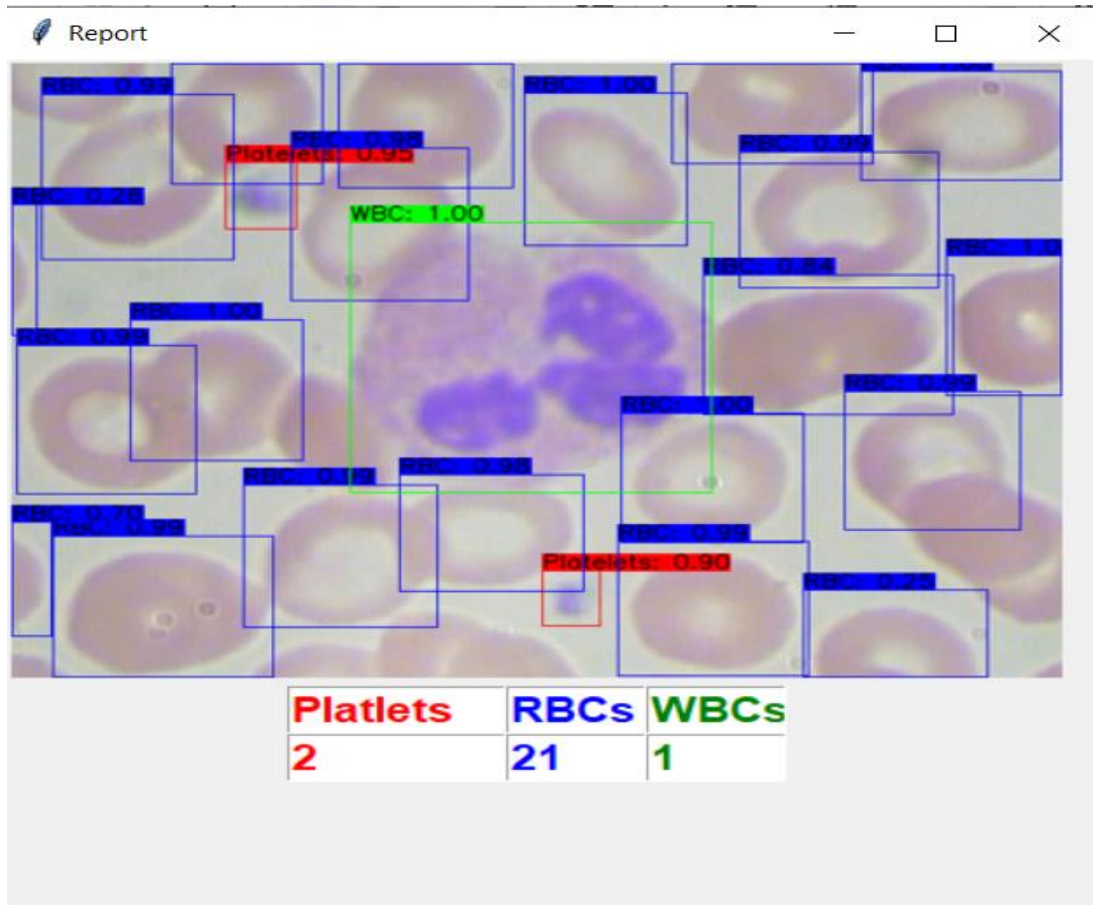


Figure 34- Result

- Important Installations
 - Download python with all of the used libraries (tensorflow, tkinter, PIL, os, numpy, cv2 and matplotlib).
 - Download CUDA to use the GPU for processing.

5.3 Website

5.3.1- Login Page

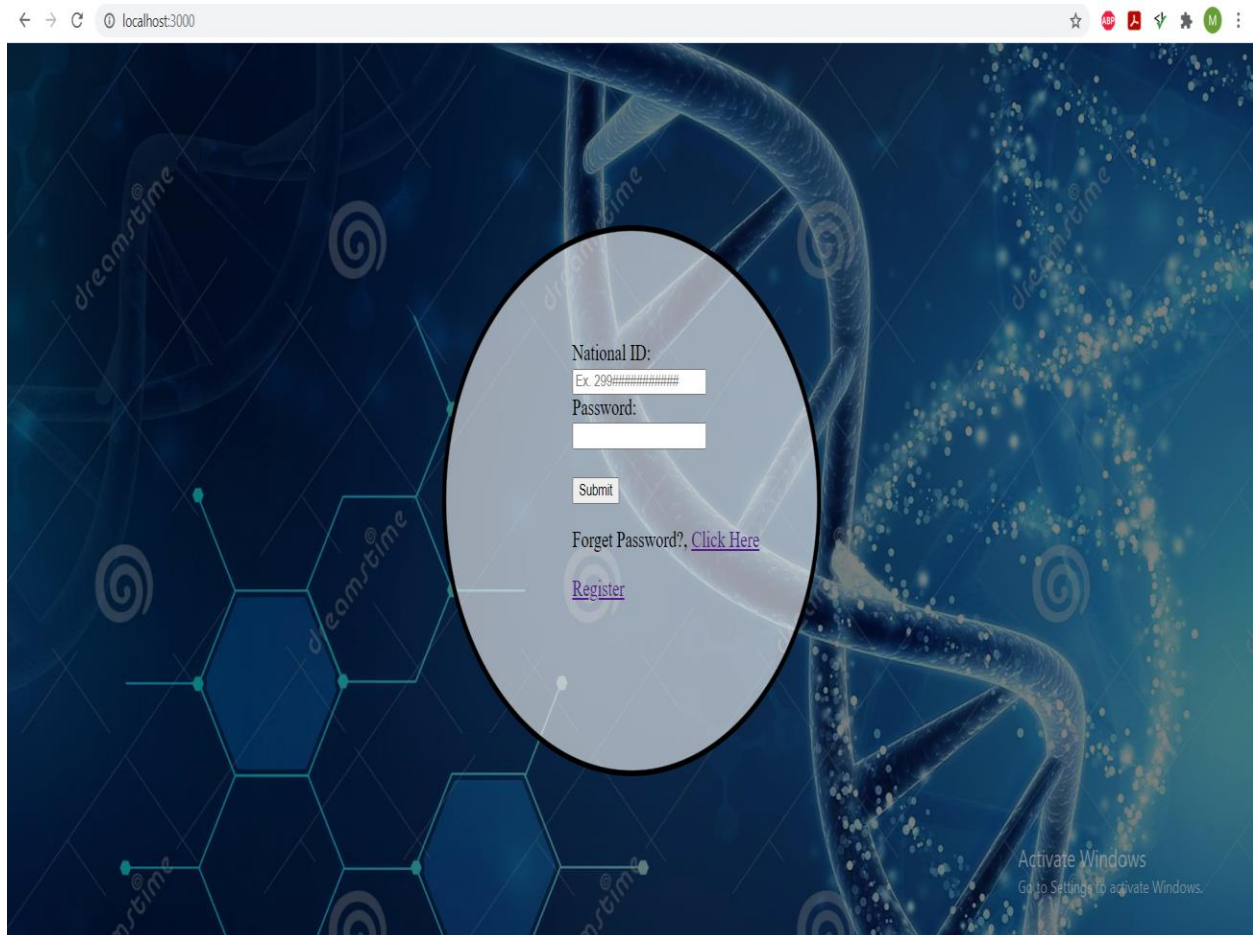
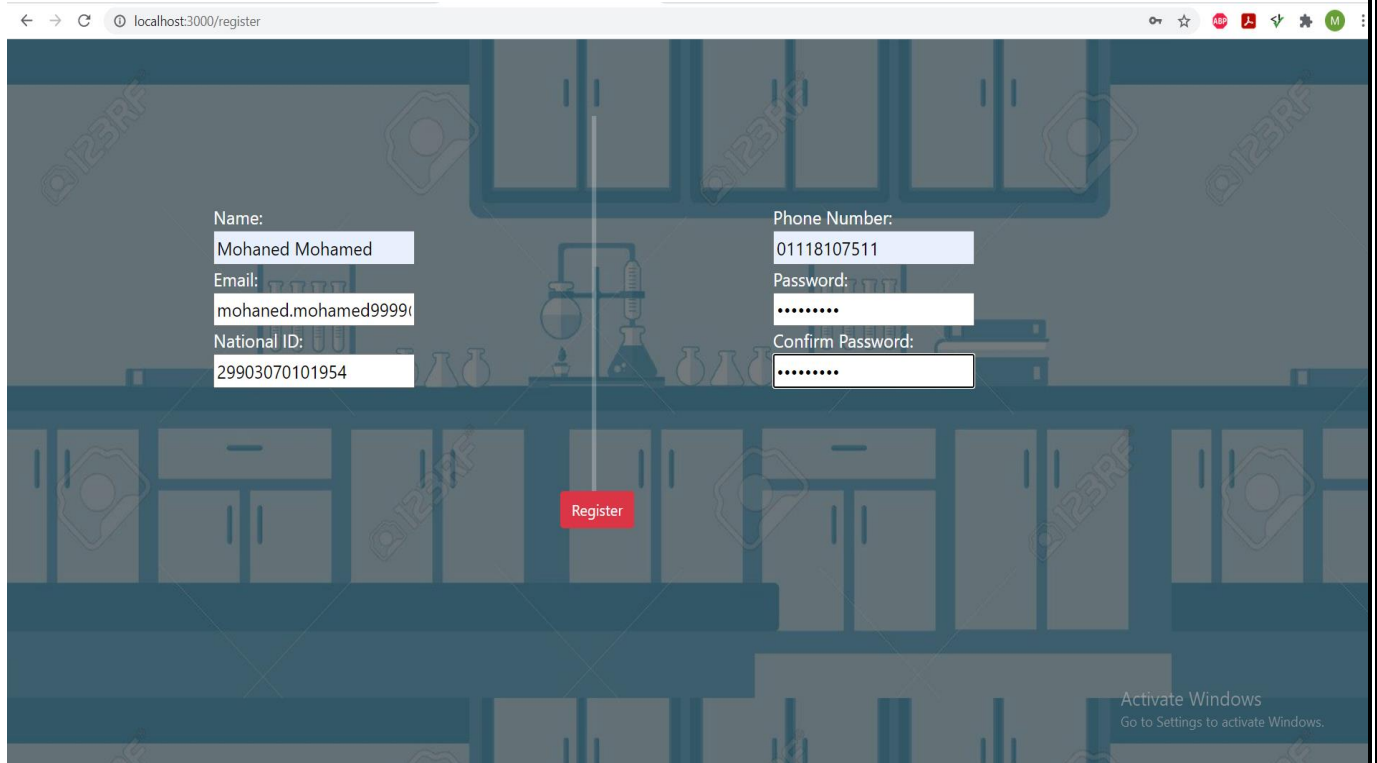


Figure 35-Login

- This is the first page the user interacts with in the website so he can use other website privileges.
- Firstly the user enters his data which are the national id and the password and the login function in the user controller checks for the validity of this data and according to it if the user is patient he can use the view reports page otherwise the doctor can add a report for a patient.
- Then the user id is encoded in the JWT header for authentication.

5.3.2- Register Page

A screenshot of a web browser displaying a registration form. The browser's address bar shows 'localhost:3000/register'. The form is set against a dark blue background with a faint laboratory-themed illustration. It contains two columns of input fields. The left column has fields for 'Name' (filled with 'Mohaned Mohamed'), 'Email' (filled with 'mohaned.mohamed9999@'), and 'National ID' (filled with '29903070101954'). The right column has fields for 'Phone Number' (filled with '01118107511'), 'Password' (filled with seven dots), and 'Confirm Password' (filled with seven dots). A red 'Register' button is positioned centrally below the input fields. In the bottom right corner, there is a watermark that reads 'Activate Windows Go to Settings to activate Windows.'

localhost:3000/register

Name: Mohaned Mohamed

Email: mohaned.mohamed9999@

National ID: 29903070101954

Phone Number: 01118107511

Password:

Confirm Password:

Register

Activate Windows
Go to Settings to activate Windows.

Figure 36-Registration page

- In this page the user can create an account for himself to be used to track his reports.
- After registration the user can use the website to see his old reports in addition to newly added ones through the reports page.
- When the user enters his data and presses the register button, the register function in the user controller searches if the entered data is not repeated if the data is not repeated then a new account is made and the data is stored in the database.

5.3.3- Report page

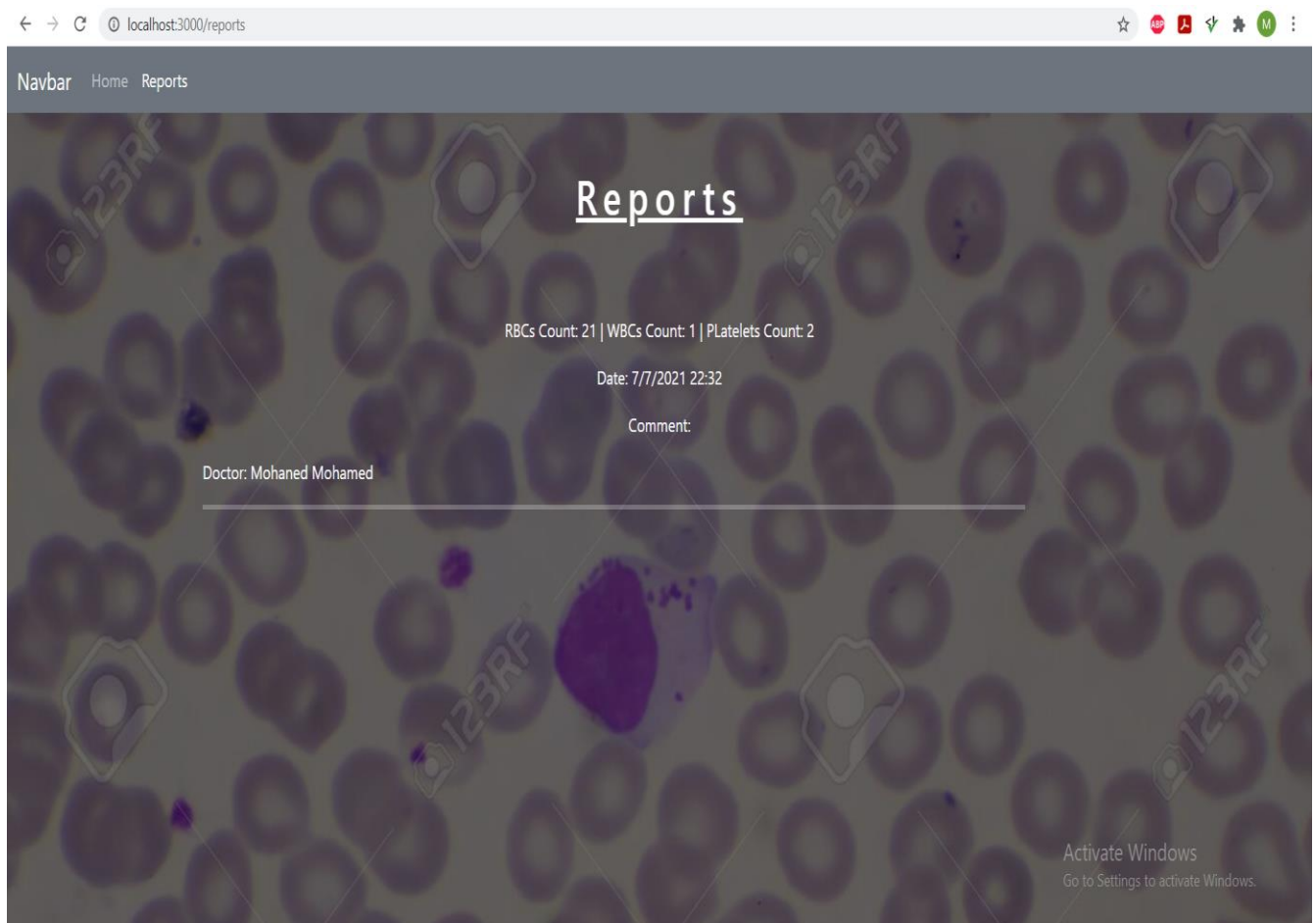


Figure 37-Report page

- In this page the patient can view all his reports which are ordered ascendingly according to the date of creation, so the new ones are at the top and old ones are down.
- When the user requests his reports, the reports function in the user controller fetches the reports table with the user id to retrieve all the reports that are related to him.

5.3.4- Add report page

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/addreport'. The page has a dark red background with a microscopic view of blood cells. A form is centered on the page with the following fields and values:

- National ID: 29903070101954
- RBCs Count: 21
- WBCs Count: 1
- Platelets Count: 2
- Comment: (empty text area)

Below the form is a 'Submit' button. In the bottom right corner, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

Figure 38- Add report page

- In this page the doctor enters the data that he gets from the desktop application and enters the user national id which is received from the laboratory.
 - When the user presses the Submit button, the add_report function in the user controller checks for data validity and then adds a new record in the report table.
- ❖ Important Installations
- Install ruby and rails
 - Install docker engine
 - run docker-compose up to set the project environment

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Complete blood count test is a highly important test in medical diagnosis , as a result we developed an application to automatically identify and count blood cells from a smear image based on YOLO algorithm., Our proposed method is based on publicly available datasets. It is observed for the test dataset that our method accurately identifies RBCs, WBCs, and Platelets. Even though different models with different depths have been tried in this field, it is observed that the method is considerably fast for counting and marking the smear images. The proposed method has also been tested on a different dataset of smear images, where it has performed satisfactorily. With the accuracy and the detection performance of the proposed method, it can be said that the method has the potential to ease up the manual blood cell identification and counting process.

The model comes with average accuracy 88.56 % .

87.72% in detecting blood platelets , 77.96% in detecting RBCs , and 100% in detecting WBCs.

6.2 Future Work

In the future we hope to enhance our model to be able to detect different types of WBCs and provide much better accuracy .

Moreover, we hope to provide our web-application on a server to be faster and reachable, also many other options for the user such as notifying him when something new has been discovered or any update in the system, and provide other operations to make in his own account.

We also aim to develop a mobile application in order to expand our project domain and provide the users multiple easy ways to follow up with their lab results conveniently

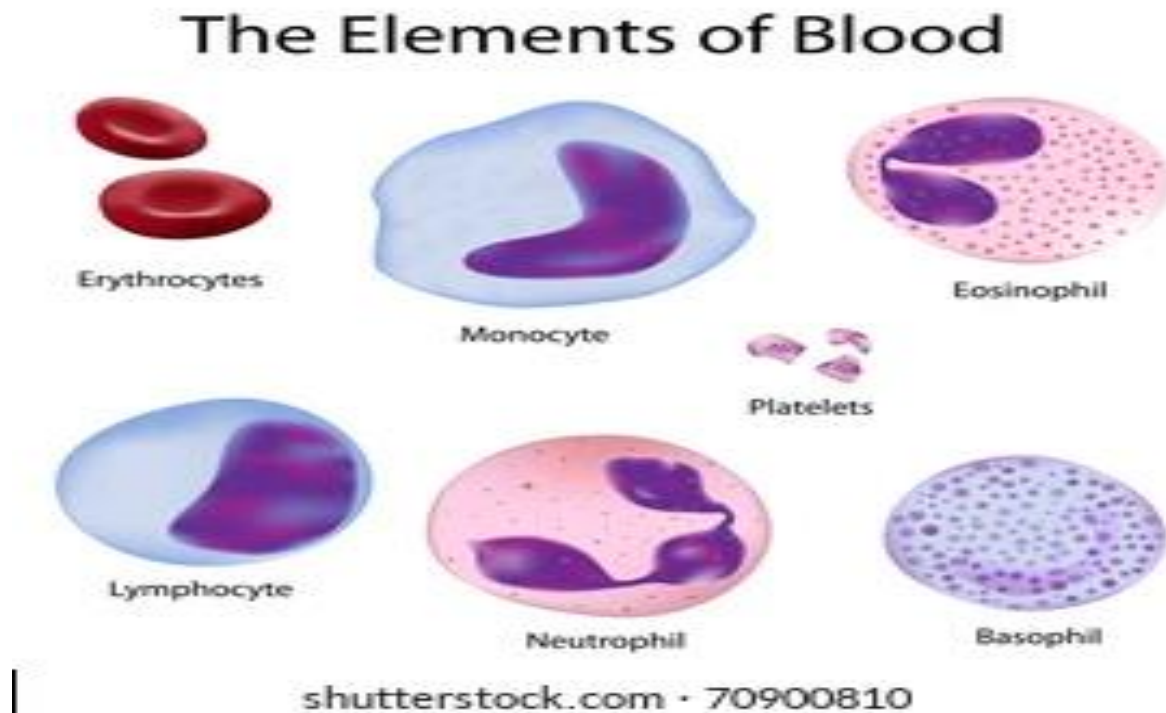


figure 39 - Elements of blood

References

1. Complete blood count. Lab Tests Online.
<https://labtestsonline.org/understanding/analytes/cbc/tab/test>. Accessed Aug. 29, 2016.
2. CBC with differential, blood. Mayo Medical Laboratories.
<http://www.mayomedicallaboratories.com/test-catalog/Clinical+and+Interpretive/9109>. Accessed Aug. 29, 2016.
3. Understanding blood counts. Leukemia and Lymphoma Society.
<http://www.lls.org/managingyourcancer/labandimagingtests/understandingbloodcounts>. Accessed Aug. 29, 2016.
4. Cruz D., Jennifer C., Valiente L.C., et al. : ‘Determination of blood components (WBCs, RBCs, and platelets) count in microscopic images using image processing and analysis’. Proc. Int. Conf. on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), December 2017 [[Google Scholar](#)]
5. Habibzadeh M., KrzyÅijak A., Fevens T.: ‘White blood cell differential counts using convolutional neural networks for low resolution images’. Proc. Int. Conf. on Artificial Intelligence and Soft Computing, 2013, pp. 263–274 [[Google Scholar](#)]
6. Acharjee S., Chakrabartty S., Alam M.I., et al. : ‘A semiautomated approach using GUI for the detection of red blood cells’. Proc. Int. Conf. on Electrical, Electronics, and Optimization Techniques, 2016, pp. 525–529 [[Google Scholar](#)]
7. Lou J., Zhou M., Li Q., et al. : ‘An automatic red blood cell counting method based on spectral images’. Proc. Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics, October 2016, pp. 1391–1396 [[Google Scholar](#)]

8. Avendi M.R., Kheradvar A., Jafarkhani H.: ‘A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI’, *Med. Image Anal.*, 2016, 30, pp. 108–119 (doi: 10.1016/j.media.2016.01.005) [[PubMed](#)] [[Google Scholar](#)]
9. Gulshan V., Peng L., Coram M., et al. : ‘Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs’, *J. Am. Med. Assoc.*, 2016, 316, (22), pp. 2402–2410 (doi: 10.1001/jama.2016.17216) [[PubMed](#)] [[Google Scholar](#)]
10. Ren S., He K., Girshick R., et al. : ‘Faster R-CNN: towards real-time object detection with region proposal networks’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, 39, (6), pp. 1137–1149 (doi: 10.1109/TPAMI.2016.2577031) [[PubMed](#)] [[Google Scholar](#)]
11. Redmon J., Divvala S., Girshick R., et al. : ‘You only look once: unified, real-time object detection’. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016 [[Google Scholar](#)]
12. Acharya V., Kumar P.: ‘Identification and red blood cell automated counting from blood smear images using computer-aided system’, *Med. Biol. Eng. Comput.*, 2018, 56, (3), pp. 483–489 (doi: 10.1007/s11517-017-1708-9) [[PubMed](#)] [[Google Scholar](#)]
13. Sarrafzadeh O., Dehnavi A.M., Rabbani H., et al. : ‘Circlet based framework for red blood cells segmentation and counting’. IEEE Workshop on Signal Processing Systems, December 2015 [[Google Scholar](#)]
14. Kaur P., Sharma V., Garg N.: ‘Platelet count using image processing’. Int. Conf. on Computing for Sustainable Global Development (INDIACom), March 2016 [[Google Scholar](#)]
15. Zhao J., Zhang M., Zhou Z., et al. : ‘Automatic detection and classification of leukocytes using convolutional neural networks’, *Med. Biol. Eng.*

- Comput.*, 2017, 55, (8), pp. 1287–1301 (doi: 10.1007/s11517-016-1590-x)
[PubMed] [Google Scholar]
16. Habibzadeh M., Jannesari M., Rezaei Z., et al. : ‘Automatic white blood cell classification using pre-trained deep learning models: ResNet and inception’. AIP Conf. Proc., 2017, vol. 1883, pp. 1–8 [Google Scholar]
 17. The privalsky lab
[Online]<http://microbiology.ucdavis.edu/privalsky/hemocytometer>
 18. Xu M., Papageorgiou D.P., Abidi S.Z., et al. : ‘A deep convolutional neural network for classification of red blood cells in sickle cell anemia’, *PLoS Comput. Biol.*, 2017, 13, (10), pp. 1–27 (doi: 10.1371/journal.pcbi.1005746)
[PMC free article] [PubMed] [Google Scholar]
 19. He K., Zhang X., Ren S., et al. : ‘Deep residual learning for image recognition’. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016 [Google Scholar]
 20. Szegedy C., Vanhoucke V., Ioffe S., et al. : ‘Rethinking the inception architecture for computer vision’. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016 [Google Scholar]
 21. Simonyan K., Zisserman A.: ‘Very deep convolutional networks for large-scale image recognition’. Proc. Int. Conf. Learning Representations, San Diego, 2015, pp. 1–14 [Google Scholar]
 22. Howard A.G., Zhu M., Kalenichenko B.C.D., et al. : ‘Mobilenets: efficient convolutional neural networks for mobile vision applications’, arXiv preprint arXiv:1704.04861, 2017
 23. [2017] ruby on rails [online]
https://guides.rubyonrails.org/getting_started.html