Blood Group Detection Using Fingerprint

Bhagyashree S Lipane[1], Pravin R Rathod[2]

[1,2]Computer Science and Engineering Department, Deogiri Institute of Engineering and Management Studies, Ch. Sambhajinagar Dr. Babasaheb Ambedkar Technological University Lonnere

***Abstract*—Before performing blood transfusions in severe situations, blood group detection is necessary. It is done before a blood transfusion in an emergency or when checking a person's blood group for donation. Currently, lab personnel perform tests manually in the laboratory. This takes time and may result in human mistake when determining blood type. The goal of the study survey is to use image processing to reduce the amount of physical labor required to identify blood groups. The presence or absence of agglutination reaction of blood with antigen will be used to determine the blood group**

I. INTRODUCTION

Blood is an essential to life. It circulates through human body and brings oxygen and nutrients to all the parts of body so that they can keep working. It carries carbon dioxide and other waste material to the lungs, kidneys and digestive system so that waste material to be removed from the system. Blood group is a classification of blood based on the presence or absence of antigenic substances in blood cells. Blood types where first discovered by an Austrian physician, Karl Landsteiner. In 1901, he observed that there are substances in the blood like antigen and antibody that form clumping of red cells when one type of blood is added to another type of blood. Based on this he recognizes three types of blood groups as A, B and C. He defined that group A agglutinates with group B, similarly group B agglutinates with group A but group C blood is different because it agglutinates with both A and B. Thus, he discovered two antigens and two antibodies. In 1910, Ludwik Hirszfeld and Emil Freiherr von Dungern introduced the term O(null) for the group Landsteiner designated as C which has no antigens but antibodies anti-A and anti-B. The fourth less frequent blood group AB, was discovered by Sturli and von Decastello, which has both A and B antigens but no antibodies. The Rh blood group was discovered in 1940 by Karl Landsteiner and A. S. Weiner; they classify blood group according to the presence or absence of Rh antigen

Following are the blood groups present in human body
1) Group A positive or A negative:
Antigens are present on surface of blood cells. Anti-B antibodies are present in the plasma.
2) Group B positive or B negative:
B antigens are present on surface of blood cells. Anti-A antibodies are present in the plasma.
3) Group AB positive or AB negative:
A and B antigens are present on surface of blood cells. There are no antibodies in the plasma. People with group AB positive blood can usually receive from any group.
4) Group O positive or O negative:
There are no antigens are present on surface of blood cells. Both anti-B and anti-A antibodies in the plasma. O is a universal donor. People with O blood group can donate blood to people with any blood group.

Blood group identification is very important to make sure blood transfusion safety. Blood grouping is essential for many major medical procedures. Blood Detection is the most important and essential activity in human life. Patients with Thalassemia require a regular blood transfusion. So, it is important to identify the correct blood group before blood transfusion, donation, and other emergency situations, which may directly relate to the survival and life of the patient.

An ABO incompatibility reaction can occur if a patient receives the wrong type of blood during a blood transfusion. Where ABO incompatibility reaction is nothing but, antibodies that the patient already has in his or her blood will attack the donor blood cells and destroy them. This will cause some dangerous effects on the immune system such as fever, chills, chest or back pain, bleeding, increased heart rate, shortness of breath, kidney damage and human death is also possible.

The traditional method of determining blood type in the laboratory can be replaced by a digital method using image processing technology. Image processing is helping in many ways to achieve their goals, especially in the security and medical fields. In the medical field, image processing is used for various tasks like PET scan, X-ray imaging, medical CT, UV imaging, cancer cell image processing, and much more. Nowadays image processing techniques are widely used for blood group detection. It only takes a short time to determine the blood type and there should be no errors.

There are image matching algorithms such as scale invariant feature transform (SIFT), speed-up robust feature (SURF) and oriented fast and rotated brief (ORB) algorithm which are used to find out the similarities in the image.

SIFT is feature detection algorithm in computer vision. This algorithm helps to locate the local features in an image, commonly known as key points of the image. It takes an image and transforms it into a large collection of local features. This algorithm is distinctive where individual features can be matched for large database objects. It provides many features for even small objects. ORB is an efficient alternative to sift or surf algorithms used for feature extraction, in computation cost and matching performance. This algorithm has ability to reduce sensitivity to noise.

The various Deep Learning methods use data to train neural network algorithm to do a variety of Machine Learning tasks, such as the classification of different classes of objects. Convolutional Neural Network are Deep Learning algorithms that are very powerful for analysis of images.

There are many applications of this blood detection system, as correct blood group is required everywhere in the medical system. For example, before donating blood at a blood donation camp, correct blood group of donors is required, in rural areas, laboratories are not available to detect correct blood type, in such cases people can use this system to detect the blood group.

1.1 Problem Definition

To implement Blood Group detection system by using microscopic image of blood sample.

In this project, it is very important to determine the blood type quickly and accurately in an emergency before transfusion. Today, rapid blood typing methods based on image recognition technology are widely used in automated blood analyzers. This project proposes a fast, accurate, and robust blood group analysis method based on the imaging function of the ABO high-speed blood group analyzer. Then, according to the gray level distribution of the image, the characteristic parameters of the ABO blood group are extracted. With the agglutination reaction between the antigen and the antibody, the system ultimately determines the blood type. Experimental results show that this method can quickly and accurately classify ABO blood groups.

1.2 Aim and Objective

The proposed system aims to develop a system that performs blood tests based on the ABO and Rh blood group using image processing technique. The objectives of the project are

1. To preprocess the blood drop images.
2. To implement a model for identifying blood group is A, B, AB or O (Positive or Negative) category.
3. To provide result within the shortest possible time along with storage of result for further use.

1.3 Scope and Limitation of Project

The technique captures a single image of a blood sample. Then, to locate the agglutination in the image, use an image matching method (SIFT or ORB). CNN is also used to classify and train the images.

Antigen mixed blood pictures are required by the system. It also fails to detect the Bombay blood group, which is a rare blood type

Efforts: The effort is only a function of the number of lines of code and some constants evaluated according to the different software system.

Efforts = a (KLOC) b
= 2.4 (1.970)1.05
= 2.4 * 2.03
= 4.87

Time: The amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, months.

Time = c (Efforts)d
= 2.5 (4.87) 0.38
= 2.5 * 1.82
= 4.55

Persons Required: Persons required is nothing but effort divide by time.

Persons Required = Efforts / Time
= 4.87 / 4.55
= 1.07

II. BACKGROUND STUDY & LITERATURE REVIEW

2.1 Literature Overview

medlinePlus.gov [1] Slide test is traditional method of detecting blood group. In this method, a drop of donor or recipient blood is mixed with anti-A, anti-B, anti-D separately. And agglutination or blood clumping pattern is observed. The test completes in 5 to 10min and it is inexpensive. However, the clumping pattern is observed by human so it may introduce human error.

Mehedi Talukdar, Md Rabiul slam et. al. [2] has suggested that blood group can be detected by using image processing technique. The system will take blood image and calculate standard deviation. If value of standard deviation is greater than 20 then agglutination reaction occurred and the test is stated as positive otherwise it will be considered as negative.

S. M. NaziaFathima [3], proposed semiautomated system for detecting blood group which can be identified by microscopic colouring image. Initially it performs image preprocessing by histogram equalization, color correction and then color space conversion for converting the RBC to HIS. It will detect color and texture of image using cumulative histogram and then corresponding person's blood group can be analysed by using SVM. In this system more skilled persons are needed to handle the system and it is tedious to do.

Enes Ayan, Erdem Kamil Yildirim [4], proposed a system for detecting blood using gel test method. The gel card used in this method which has a special gel. The blood cells which don't agglutinate passes through this gel and accumulate at the bottom of tube. Toz and colleagues developed a software for reading gel test cards by using image processing technique. Gel test method needs three devices these are gel test centrifuge, gel test incubator and gel test reader. These devices are very expensive.

Anurag Sadashiv phad, Tejas Sanjay Targhale, Bharat Bhalshankar, Sunita Kulkarni [5] proposed a system which will detect the blood by using Raspberry Pi-3. In this system, reagents are mixed with three samples of blood. After sometimes, agglutination may or may not occur. After the formation of agglutination, the slide is captu red using the raspberry camera module and using raspberry pi-3 module. Then, processing of image is carried out such as morphology, thresholding, segmentation, quantification etc. finally result is display on the LCD. But the limitation of this system is it require costlier hardwires.

Mansi K, Hita Shree M., Chandana Lakshman Hegde paper [6], blood group detection system is proposed using image processing technique which can be used by lab technician and novice user with no prior knowledge to blood group detection. Only they need to take the blood and add three antigens into the blood and take the image and pass it to the system. The system will be able to process the image and give final output. The main step involved in this application are Image segmentation, Thresholding, Morphology, Histogram and Quantification

2.2 Investigation of current project and related work:

Nuha Odeh, Anas Toma, Falah Mohammed [7], An Efficient System for Automatic Blood Type Determination Based on Image Matching Techniques is proposed fast and accurate system to determine type of blood automatically based on image processing. A number of blood samples were processed using different image matching techniques including ORB and SIFT. These algorithms are used to find the agglutination reaction in the blood image. Also, Convolution Neural Network is used to train the image and predict the blood group.

Advantage of proposed system is, this system will not require any costlier hardware.

III. REQUIREMENT GATHERING AND SPECIFICATIONS

3.1 Requirement Gathering

1. As a laboratory employee, login in Blood group detection software, so that I can handle the software.
2. As a laboratory employee, add new employee information, so that new employee also can handle software.
3. As a laboratory employee, add new patient information, so that I can use this information for detecting blood group of patients.
4. As a laboratory employee, select blood sample slide image, so that software can process further steps on the image
5. As a laboratory employee, A process of blood detection will be done.
6. As a laboratory employee, send email to the patient regarding patient blood group final report,

so that patient can get their blood group report easily.

7. As a laboratory employee, see donor list of patients, so that it is useful for blood donation.
8. As a laboratory employee, exit from blood group detection software. So that I can close software successfully.

3.2 User Stories

1. A laboratory employee can login in blood group detection software with username and password.
2. A laboratory employee can add new employee with name, contact number, email id, gender, state & password.
3. A laboratory employee can add new patient with name, email id, contact number, gender, state, donor (yes or not) and blood sample slide image.
4. A laboratory employee can select blood sample slide image.
5. A laboratory employee can process blood detection.
6. A laboratory employee can send email of patient blood group report to patient with registered email id and number.
7. A laboratory employee can see the donor list with name and blood group information.
8. A laboratory employee can exit from software by clicking on exit tab

3.3 Requirement Specification

No	Requirement	Essentials/	Description of Requirement	Remark
RS1	System should have a facility to login	Essential	After inserting valid login id and password and clicking on login button system should be login successfully into the software.	Username and password should valid
RS2	The system should have a facility to add new employee	Essential	After clicking on employee button/tab form should ask required information of employee and should add into the database.	
RS3	The System should have a facility to add new patient	Essential	After clicking on patient button/tab form should ask for required information of patient and should add into the database.	
RS4	The System Should select Blood ample slide image	Essential	After clicking on select slide button system should select blood sample slide image.	File extension should be jpg, jpeg or png
RS5	System should Detect blood group properly and store it into database	Essential	After clicking on image processing steps button system should apply all processes successfully and after clicking on store button system should store blood group information into database.	
RS6	System should send email to patient	Essential	After clicking on store button system should redirect to email and msg sending form.	
RS7	System should display the donor list of patients	Essential	After clicking on donor tab button system should display donor list of patients.	
RS8	System should exit from software successfully.	Essential	After clinking on exit button/tab system should exit from application successfully.	

Table 1: Requirement Specification Table

3.4 Use Case diagram

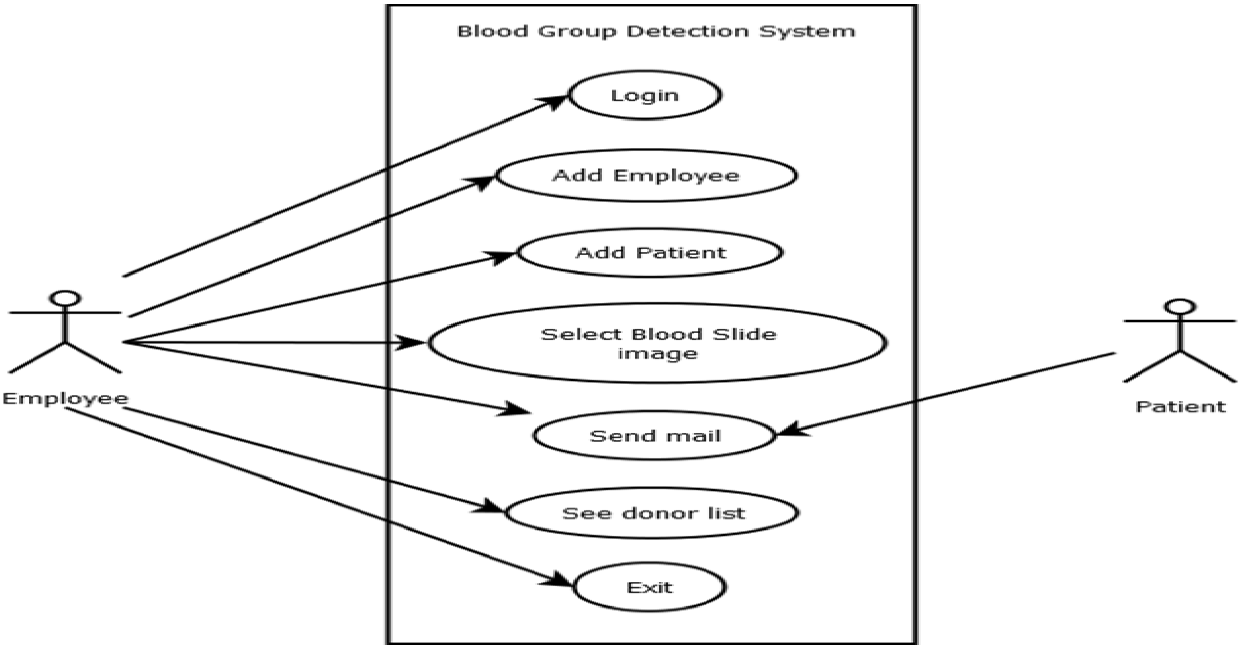

Fig 2: Use Case Diagram

IV. SYSTEM DESIGN

4.1 Architecture Diagram

When Employee login into system it sends request to database. Database confirm username and password is correct or not. Then employee can successfully login into system. After successful login employee can register new employee. Employee can register new patient and select blood slide image.

When Blood slide image is selected, it goes to image matching and classification techniques Finally, blood group is detected and store it into database.

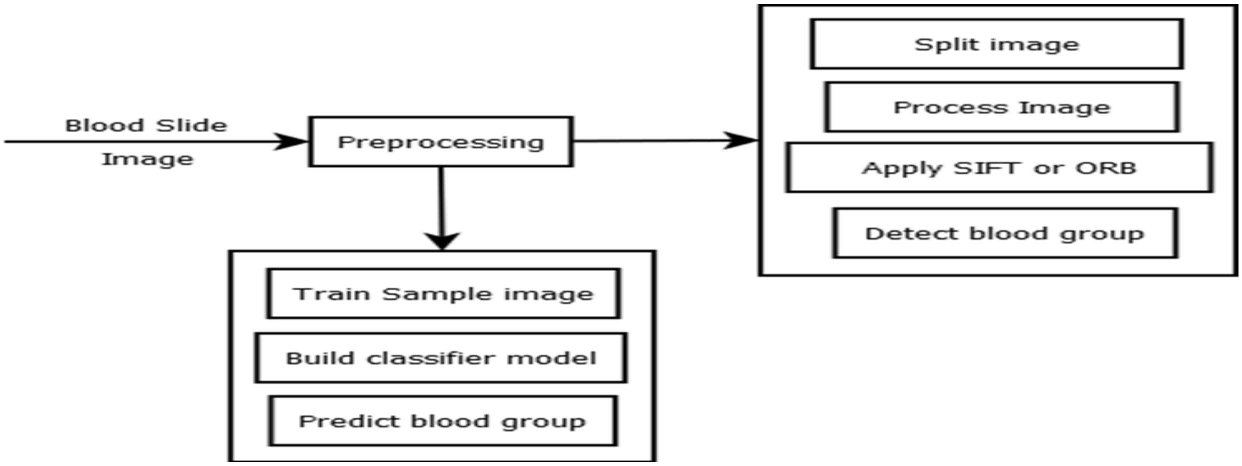

Fig 3: Architecture Diagram

4.2 Algorithms Description of each Module

1. Login

Login Successfully is essencial to System Algorithm:

```
1. Begin
2. Enter username and password
    a. if(correct)-> login Succesful
    b. else-> login unsuccesful
3. Go to entry menu
4. End
```

2. Registration:

There are two types of registraion employee and patient registration.

A. Employee: Algorithm:
1. Begin
2. Enter all information of employee
3. Click on submit button
4. End

B. Patient: Algorithm:
1. Begin
2. Enter all information of employee
3. Click on select slide button
4. End

3. Detection:

Detection module has four sub modules

A. Select slide: Algorithm:
1. Begin
2. Select blood slide image
3. Display image on screen
4. End

B. Split image: Algorithm:
1. Begin
2. Take the slide image
3. Divide it into three part
4. Display images on screen
5. End

C. Image Matching: Algorithm:
1. Begin
2. Take split images one by one and standard image.
3. Detect Keypoints and computes descriptors of split image and standard image.
4. Match the image split image and standard image by using descriptors.
5. Find out matching points between images.
6. With help of matching points detect the blood group.
7. End

4. Prediction

Algorithm
1. Begin
2. Load training model
3. Provide the selected image.
4. Test the selected image with the help of trained model.
5. Display the blood group
6. End

5. Email

Algorithm
1. Begin
2. Enter all details
3. Send mail
4. End

4.3 System Modelling

4.3.1 Dataflow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. Blood slide image passes through blood group detection system. Then image matching techniques and classification model are applied and final blood group is detected

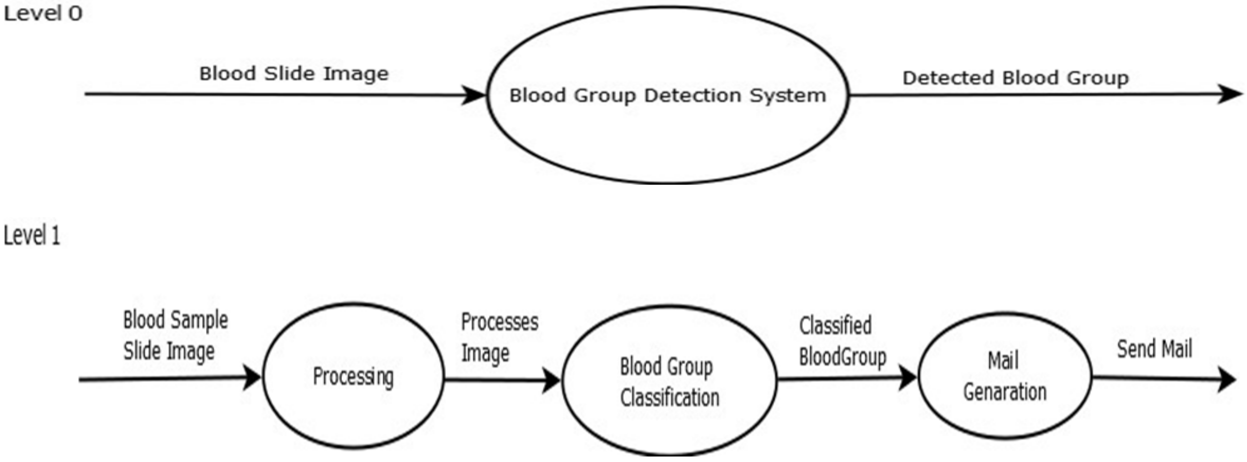

Fig 4: DFD Level 0 and DFD Level 1

4.3.2 Class Diagram

A class defines a method and variable of an object that is a particular entity in your program, or a unit of code that represents that entity. Each time a staff member selects a blood slide image, image processing techniques are applied to identify the final blood type. In addition, the blood type of the database is stored in the database.

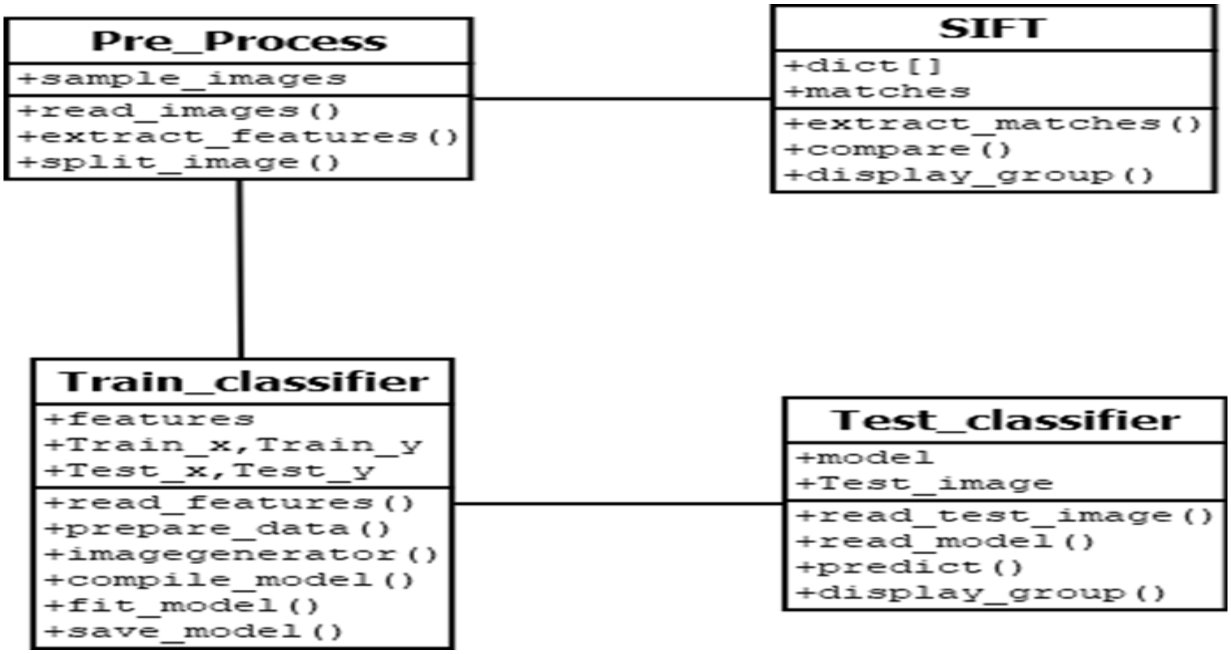

Fig 5: Class Diagram

4.3.3 Sequence Diagram

A sequence diagram is a chronological arrangement of object interactions. The solid bar indicates the active time of the module. The dotted line shows the inactivity time of the model. The sequence diagram shows the flow of the project sequentially.

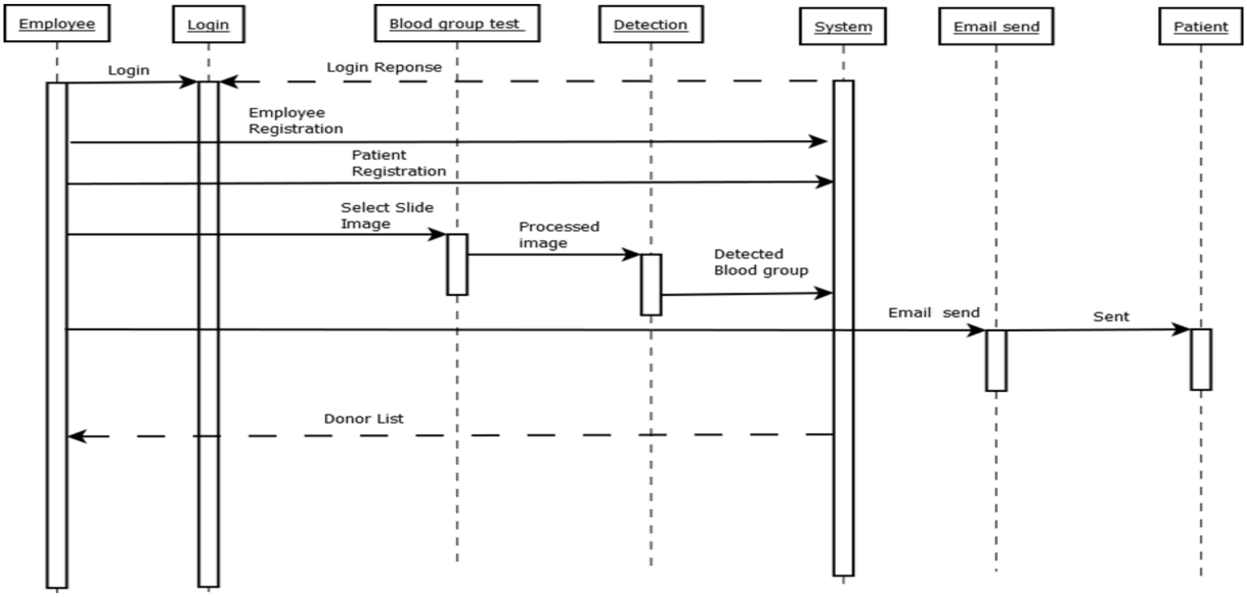

Fig 6: Sequence Diagram

4.3.4 Activity Diagram:

An activity diagram is basically a flow chart that shows the flow from one activity to another. Activities can be described as operating the system.

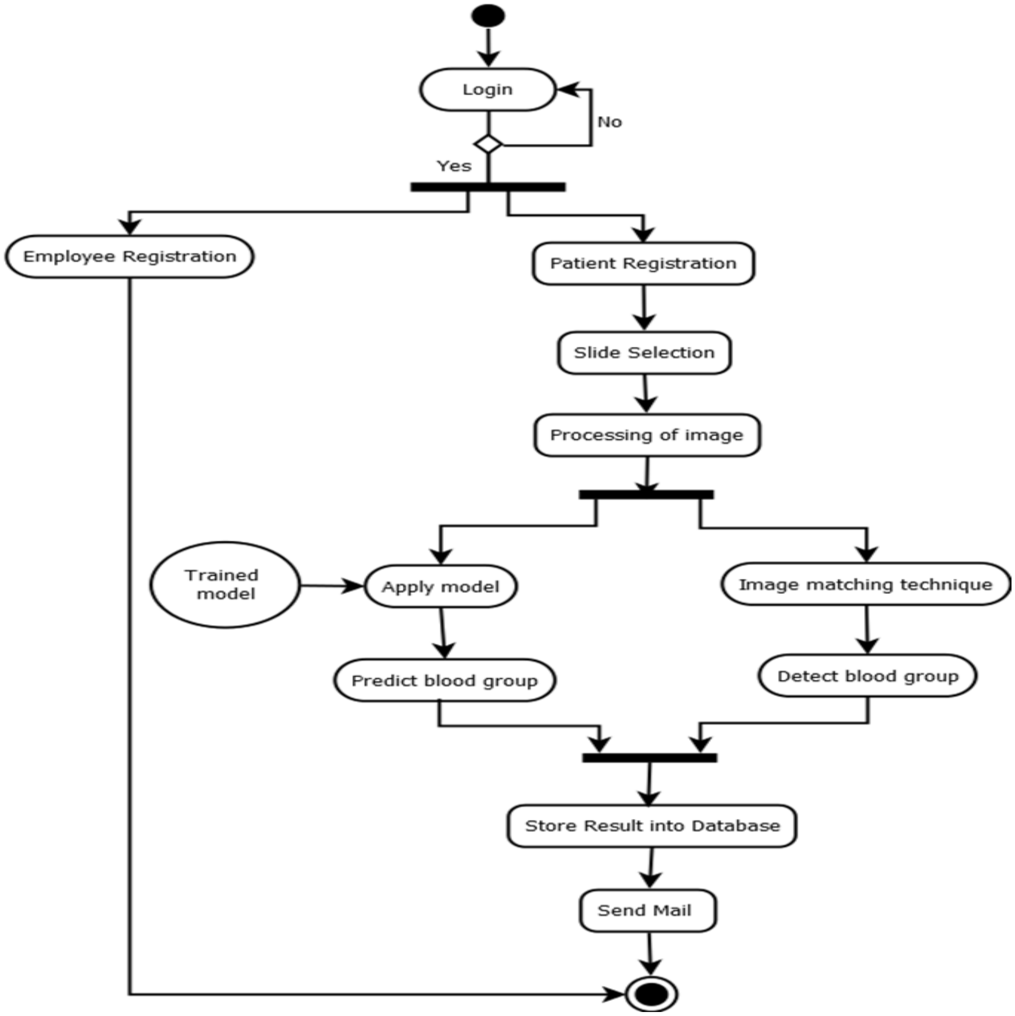

Fig 7: Activity Diagram

4.3.5 Database Design:

1. Database name: BloodGroup Details
2. Tables:

a. Employee table:

Field: Name, Email, Contact_Number, Gender, State, password

Name	Email	Contact Number	Gender	State	Password

Table 2: Employee Table

b. Register table:

Field: Name, Email, Contact_Number, Gender, State, Register date, slide_path, Donor, Blood Group

Name	Email	Contact_ Number	Gender	State	Register_Date	Slide_path	Donor	Blood_Group

Table 3: Register Table

V. IMPLEMENTATION

5.1 Environmental settings for running the project

- to run this project database (SQLite) must be installed into the system.
- to run this project python environment particularly Anaconda environment must be installed into the system.
- to run this project following packages are required.

a) Pillow
b) TensorFlow
c) Keras
d) Matplotlib
e) OpenCV
f) Tkinter
g) Sqlite3
h) sklearn

5.2 Detailed Description of Methods

Our system has different types of methods for handling all the operations. The different methods of our system are

1. Check login (): This function checks the login credentials.
2. Insert record (): This function enters all values of employee and patient into database.
3. Select image (): This function is for selecting slide image.
4. Split image (): This function is split the blood image into three separate images for further comparison.
5. Detection (): This function finds out the matching distance between each split images and standard image and with help of matching distance it detects the blood group.
1. 6. Prediction (): In this function train images model is loaded. Test the selected image with help of trained model predict the blood group.
6. Send mail (): This function is for send mail to patient registered mail id

5.3 Implementation Details

1. Check login ():

```
def check login(): un=self.email_tf.get()
ps=self.pwd_tf.get()
messagebox.showinfo('confirmation',ps) con =
sql.connect("BloodDetails.db") cur = con.cursor()
#statement = "SELECT Email from Register WHERE
Email='"+un+"' AND Password = '{ps}';"
statement = "SELECT Email from Employee WHERE
Email='"+un+"' AND Password = '"+ps+"';"
cur.execute(statement) if cur.fetchone():
messagebox.showinfo('confirmation', 'Login
Successful') self.ws.destroy()
e=entrymenu.entrymenu() e.run()
else:
messagebox.showinfo('confirmation', 'Login
UnSuccessful')
```

2. Insert_record():

```
def insert_record(): check_counter=0
warn = ""
if self.employee_name.get() == "": warn = "Name
can't be empty"
else:
check_counter += 1
if self.employee_email.get() == "": warn = "Email
can't be empty"
else:
check_counter += 1
if self.employee_mobile.get() == "": warn = "Contact
can't be empty"
else:
check_counter += 1
if self.var.get() == "": warn = "Select Gender"
else:
```

```
check_counter += 1
if self.variable.get() == "": warn = "Select Country"
else:
check_counter += 1 try:
con = sqlite3.connect('BloodDetails.db')
nm=self.employee_name.get()
em=self.employee_email.get()
mb=self.employee_mobile.get () gn=self.var.get()
st=self.variable.get()
ps=self.employee_password.get()
cur = con.cursor() insert_stmt = (“insert into
Employee (Name,
Email,Contact_Number,Gender,State,Password)""val
ues (?
?,?)")
data = (nm,em,mb,gn,st,ps)
cur.execute(insert_stmt, data) con.commit()
messagebox.showinfo('BloodDetails','Record Save')
except Exception as ep:
messagebox.showerror('Error',format(ep))
```

3. Select_image():

```
def select_image(self):
self.filenames = filedialog.askopenfilename()
self.nm=self.filenames
im = Image.open(self.filenames)
resized=im.resize((350,150),Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(resized)
self.imgdis.configure(image=tkimage)
self.imgdis.image=tkimage
```

4. Split_image():

```
img = cv2.imread(self. filenames,0)
img1 = Image.open(os.path.join(self.filenames))
width, height = img1.size
tile (self.filenames, "output/", int (width / 3))
cv2.adaptiveThreshold(img,255,
cv2.ADAPTIVE_THRESH_MEAN_C,
im = Image.open("output/1.jpg")
resized=im.resize((150,150),Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(resized)
self.thradis.configure(image=tkimage)
self.thradis.image=tkimage
im = Image.open("output/2.jpg")
resized = im.resize((150, 150), Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(resized) self.
thradis1.configure(image=tkimage) self.
thradis1.image = tkimage
im = Image.open("output/3.jpg")
resized = im.resize((150, 150), Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(resized) self.
thradis2.configure(image=tkimage) self.
thradis2.image = tkimage
```

5. Detection ():

```
img = cv2.imread(self.filenames, 0)
#kernel = np.ones((5,5), np. uint8)
#img_dilation = cv2.dilate(img, kernel, iterations=1)
#cv2.imwrite('./setgenerated/morph.png’,
img_dilation) dict = []
for ind in range (1, 4):
img1 = cv2.imread('output/' + str(ind) + ".jpg")
# img1 = cv2.cvtColor(img1,
cv2.COLOR_BGR2GRAY) img2 =
cv2.imread('standard/std.jpg')
# img2 = cv2.cvtColor(img2,
cv2.COLOR_BGR2GRAY)
# Create SIFT object
# sift = cv2.xfeatures2d. SIFT_create() sift =
cv2.ORB_create(nfeatures=1500)
# Detect SIFT features in both images
keypoints_1, descriptors_1 =
sift.detectAndCompute(img1, None) keypoints_2,
descriptors_2 = sift.detectAndCompute(img2, None)
# Create feature matcher
bf = cv2.BFMatcher(cv2.NORM_L1,
crossCheck=True)
# match descriptors of both images
matches = bf.match(descriptors_1, descriptors_2)
# sort matches by distance
matches = sorted (matches, key=lambda x: x.distance)
# print(len(matches)) dict.append(len(matches))
# Draw first 50 matches
matched_img = cv2.drawMatches(img1, keypoints_1,
img2, keypoints_2, matches [:50], img2, flags=2)
# Show the image cv2.imshow('image', matched_img)
cv2.waitKey(0)
# Save the image
# cv2.waitKey(0) cv2.destroyAllWindows()
cv2.imwrite("matched_images.jpg", matched_img)
im = Image.open("matched_images.jpg")
resized = im.resize((350, 150), Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(resized)
self.morh.configure(image=tkimage) self.morh.image
= tkimage
print (self.filenames, dict)
if ((t1 == 0 and t2 == 0) or (t1 == 1 and t2 == 0) or (t1
== 0 and t2 == 1)): print ("O group")
```

```
group = "O" else:
if (t1 >= 10 and t2 >= 9): # or (t11 <5 and t22 <7):
group = "AB"
print ("AB group") else:
if ((t1 - t2) >= 4) and (t2 >= 0 and t2 <= 7): # or t2<=4):
print ("A group")
group = "A" else:
if ((t1 >= 0 and t1 <= 5) or (t2 - t1) > 10): print ("B
group")
group = "B" if (t3 >= 0 and t3 <= 2):
rh = "-" else:
if (t3 > 4):
rh = "+" print(rh)
self.fnimpr = group+" "+rh
messagebox.showinfo('BloodDetails', group+" "+rh)
```

6. Prediction ():

```
import cv2
from keras.models import Sequential
from keras.models import model_from_json from
sklearn.metrics import classification_report
model_file = open ('Data/Model/model.json', 'r')
model = model_file.read()
model_file.close()
model = model_from_json(model)
# Getting weights
model.load_weights("Data/Model/weights.h5")
# Y = predict (model, X) import numpy as np labels =
['tests'] img_size = 224
import os
def get_data(data_dir): data = []
for label in labels:
path = os.path.join(data_dir, label) class_num =
labels.index(label) for img in os.listdir(path):
try:
img_arr = cv2.imread(os.path.join(path, img))[.: -1] #
convert BGR to RGB format
resized_arr = cv2.resize(img_arr, (img_size,
img_size)) #
Reshaping images to preferred size
data.append([resized_arr, class_num]) except
Exception as e:
print(e) return np.array(data)
val = get_data('Data/Test') x_val = []
y_val = []
for feature, label in val: x_val.append(feature)
y_val.append(label)
x_val = np.array(x_val) / 255 x_val.reshape(-1,
img_size, img_size, 1) y_val = np.array(y_val)
predictions = model.predict(x_val)
# predictions = predictions.reshape(1,-1)[0]
print(predictions)
# Run the inference
#prediction = model.predict(data)
res=np.argmax(predictions)
# messagebox.showinfo('BloodDetails',res)
if res==0: self.fn="A+"
messagebox.showinfo('BloodDetails',"A+") elif
res==1:
self.fn=res="A-"
messagebox.showinfo('BloodDetails',"A-")
elif res==2: self.fn="AB+"
messagebox.showinfo('BloodDetails',"AB+") elif
res==3:
self.fn="AB-"
messagebox.showinfo('BloodDetails',"AB-")
elif res==4: self.fn="B+"
messagebox.showinfo('BloodDetails',"B+") elif
res==5:
self.fn="B-" messagebox.showinfo('BloodDetails',"B-
")
elif res==6: self.fn="O+"
messagebox.showinfo('BloodDetails',"O+") elif
res==7:
self.fn="O-"
messagebox.showinfo('BloodDetails',"O-")

self.msg=self.msg+"Blood Group: "+self.fn+"\n"+
"Thanks for visiting binary pathology Lab"
```

7. Send_mail():

```
def send_msg(self): subject=self.sub.get()
body=self.message.get()
msg="Subject:{}\n\n{}".format(subject, body) smtp =
smtplib.SMTP('smtp.gmail.com', 587) smtp.ehlo()
smtp.starttls()
smtp.login(self.mail.get(),self.passs.get())
smtp.sendmail(self.mail.get(),self.too.get(),msg
```

VI. INTEGRATION AND TESTING

6.1 Description of Integration Modules

Sr. No	Output Module	Input Module	Status				Status
	Module Name	Output	Type	Module Name	Input	Type	
1.	Patient Registration	Patient information saved in database	String	Login	Username & password	String	Ok
2.	Slide Selection	Blood slide Image	jpeg, jpg, png	Login	Patient information saved in database	String	Ok
3.	Split Image	Split the image into three separate images	jpeg, jpg, png	Slide Selection	Blood slide Image	jpeg, jpg, png	Ok
4.	Detection of blood group	Images with Matching distance and points, also pop-up message for detected blood group	jpeg, jpg, png and String	Split Image	Split the image into three separate images	jpeg, jpg, png	Ok
5.	Store	Pop up the message “Store”	String	Detection of blood group	Images with Matching distance and points, also pop-up message for detected blood group	jpeg, jpg, png and String	Ok

6.2 Testing

Sr. No	Test case Description	Expected Output	Actual Output	Status
1	Log in	System needs to recognize valid username and password	System able to recognize valid username and password	Pass
2	Register new employee	System should be in position to register new employee	System able to register new employee	Pass
3	Register patient	System should be in position to register patient	System able to register new patients	Pass
4	Selection of Blood slide	System needs to select the blood slide image	System able to select the blood slide	Pass
5	Splitting of image into three sub images	System should split the image into three sub images.	System able to perform Split the image into three sub images	Pass
6	Detection	System must find out the matching points between splitted images with standard image and detect the blood group	System able to correctly find out the matching points between split images with standard image and detect the blood group	Pass
7	Prediction	System must test the selected image with help of trained model and predict the blood group.	System able to test the selected image with help of trained model and predict the blood group.	Pass
8	Mail sends	After identifying the blood group system should send mail of detected blood group to the respective patient	After identifying blood group system able to send mail of detected blood group to patient	Pass

VII. PERFORMANCE ANALYSIS

This project is based on image processing which is based on a neural network concept which is been implemented by using TensorFlow and keras. Different layers like sequential, dense, dropout are used in model. This algorithm produces a probability for image classification.

The logic of image matching techniques such as SIFT and ORB are been implemented all the selected image to recognize a blood group. This recognition of blood group is important for donation purpose and for further classification of detail study blood group

VIII. FUTURE SCOPE

For future work, it is intended to develop a low-cost, portable device for automatic determination of blood group. Using portable device employee directly take photo of blood slide and detect blood group. The project is further extended to android application. In android application employees take picture of blood slide directly on their mobile devices to identify their blood type.

IX. APPLICATION

The blood group system is based on the processing of image to acquire during the slide test. A software is developed on image processing to determine the blood group during emergency situations without any error.

Improved the further export for many other applications to use: Medical Technology:

In the medical field, Image processing is used for various tasks like PET scan, X-Ray Imaging, Medical CT, UV imaging and Cancer Cell Image Processing. The introduction of image processing in medical technology has significantly improved the diagnostic process.

Computer Vision:

One of the most useful applications of Image Processing is in computer vision. Computer vision is used to make a computer aware of things, identify things, and handle the entire environment as a whole. An important use of Computer Vision is Self-Driving cars, Drones etc.

Video Processing:

Video is basically a fast movement of images. Image processing techniques are used in Video Processing. Some methods of video processing are noise removal, image stabilization, frame rate conversion and detail enhancement.

X. INSTALATTION GUIDE USER MANUAL

Installation guide

1) Create python environment particularly spyder environment installed into the system.
2) Install the packages into the system like Pillow, TensorFlow, Keras, Matplotlib, OpenCV, Tkinter, SQLite etc.
3) Select your project using Spyder application.
4) Run the project.
5) Select blood slide.

REFERENCES

[1] S MedlinePlus - Health Information from the National Library of medicine https://medlineplus.gov/ency/article/003345.html.

[2] Mehedi Talukdar, Md Rabiul Islam, Md. Mahfuz Reza, Mahbuba Begum, Md. Mahmudul Hasan, “Improvement of Accuracy of Human Blood Groups Determination using Image processing Techniques”, International Journal of Advanced Research in Computer and Communication Engineering. Vol. 4 Issue 10, October 2015.

[3] S. M. Nazia Fathima, “Classification of Blood Types by Microscopic Color Images”, International Journal of Machine Learning and Computing. Vol. 3, No 4, August 2013.

[4] Enes Ayan, Erdem Kamil Yildirim, “Real Time Blood Type Determination by Gel Test Method on an Embedded System”, International Journal of Applied Mathematics, Electronic and Computers. Vol 4, September 2016.

[5] Anurag Sadashiv Phad, Tejas Sanjay Targhale, Bharat Bhalshankar, Sunita Kulkurni, “Blood Group Detection by Using Raspberry Pi-3”, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE). Vol 7, Issue 4, April 2018.

[6] Mansi K, Hitashree M., Chandana Lakshman Hegdepaper, “Blood Group Determination through Medical Image Processing”, International Journal for Research in Applied Science and Engineering Technology. Vol 9, Issue v, May 2021.

[7] Nuha Odeh, Anas Toma, Falah Mohammed, “An Efficient system for Automatic Blood Type Determination Based on image Matching Techniques”, MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations. Vol 11, June 2021.