IOT DEVICE TO DETECT ANEMIA: A CASE STUDY

Project ID:19-129

M.H.M.Akmal

IT16134072

B.Sc. (Hons) in Information Technology Specializing in Computer Systems and Network Engineering

Department of Information Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Date: Signature:

Date: Signature of Supervisor:

Abstract

Detection of diseases is a key aspect to be considered in the world today. Diseases if not cured early it gets worse day by day. Therefore, early detection of the disease is a key aspect that needs to be considered. In this project the team shall be developing a device to detect Anemia which is a disease caused due to the lack of Fe3+ ions in the body. This is a disease that would even lead to organ failure eventually leading to a heart attack or death. Anemia is causing the red intensity of blood to reduce. Therefore, the team shall be using that factor to detect the disease. The team have built this device using a key symptom where the intensity of red in blood is measured and the output is sent to a central server, whilst the server also gets input from a questionnaire that is built in app. Using these data, the server will decide whether the user is suffering from anemia or not.

Acknowledgment

First of all, we would like to thank our supervisor Ms. Shahika Lokuliyana for the immense support and the motivation for our research project. Our supervisor also showed the path to carry on the research works and the discussions were very helpful to continue our work.

Finally, I would like to thank all the team members of the project for their contribution and effort towards achieving the project goals and objectives.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background Literature

Image processing is a technique that is used on most image detection systems for detecting symptoms in various diseases. In this project image processing is used to detect anemia. When a patient suffers from anemia there are various types of symptoms that can be seen in the patient. Some of the symptoms that an anemia patient displays are as follows;

- Easy fatigue and loss of energy
- Unusual rapid heartbeat, when exercised a lot
- Shortness of breath and headache, particularly with exercise
- Difficulty concentrating
- Dizziness
- Pale skin
- Leg cramps
- Insomnia

Out of the above examples the symptoms that an anemia patient faces, the symptom that is selected for the image processing purpose is the pale skin factor [1]. The blood is normally giving the red color by the Fe3+ ions in the blood. When the lack of iron in the blood the redness in blood tends to reduce thus causing the skin to turn a bit pale. At this stage a beam of light is sent through the skin and get the sample of the anemia patient in order to detect the color of the blood and tested against the existing samples of healthy as well as sick patients.

Detection of these anemia is also been done through image processing in some of the other projects. Image processing was used by Siti Madihah Mazalan where they used red blood cells to detect anemia [2]. In order to achieve that they are using blood samples and crops a red blood cell out and then detect the maximum radius of the cell using image processing and detect whether the patient is suffering from Anemia or not. Similarly, many projects have been done by extracting blood. But the

uniqueness of the project that is designed by the team is that it has no external blood been extracted. The blood extraction for anemia patient is also a risky factor as one of the causes of anemia major loss blood or lower creation of blood cells by the bone marrow and major destruction of red blood in the spleens [3]. Therefore, blood loss is vital in an anemic patient.

Anemia also can be detected externally using the pale skin that can also be seen under the eye. There has been a project done to assess anemia via the paleness in the eye. The project done by Azwad Tamir from the Bangladesh University of Engineering and Technology where the anterior conjunctiva pallor of the eye is used to detect anemia. In order to detect this image processing algorithm has been used. In this project an android application has used to detect the eye color and the following features can be detected. In this project the following are detected, "It operates by quantifying the conjunctival color from digital photographs of the eye taken with a smartphone camera of appropriate resolution under adequate lighting conditions with the help of an android application that has been devised. These images are then processed to obtain the red and green component spectra of the conjunctiva color and compared against a threshold to determine whether the patient is anemic or not" [4]. The algorithm in which the image processing functions is as shown in figure 1.1.1. In order to do that the following steps have been designed.

1. Taking the photograph
2. Image processing using MATLAB
3. Detection of anemia using RGB threshold

In order to take the image, the team has used the smart phone camera. The image should be taken by pulling the lower eyelid softly with the thumb and then the image is subjected to the image processing that is designed in MATLAB. In the image processing steps the image is well read and the edges are also designed accordingly. Here the Sobel edge detection algorithm was used. Then the RGB analysis is subjected after the algorithm in the MATLAB has been run. But the accuracy rate is different.

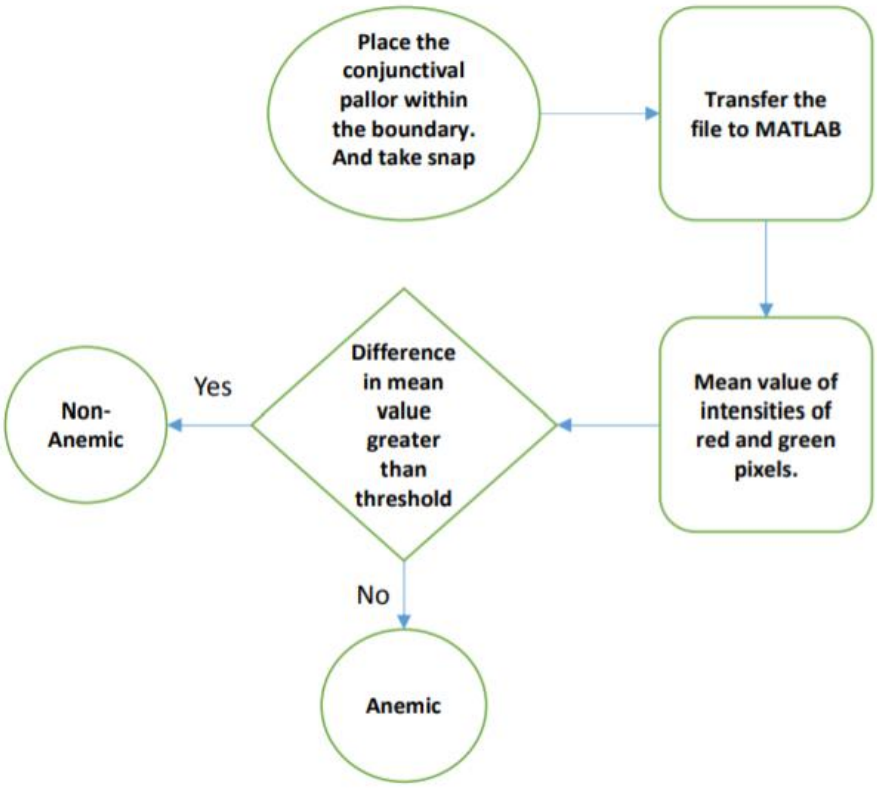

Figure 1.1.1: The steps in detecting anemia [4]

This explains the concept used to detect anemia. But the algorithm that is used in this project is not of a deep learning concept. Where in the project which is designed the disease anemia is detected using the keras algorithm which is a deep learning algorithm.

In another project done with the same concept as above a total image processing algorithm has been used. In the methodology used in the project is shown in the diagram figure 1.1.2 [5].

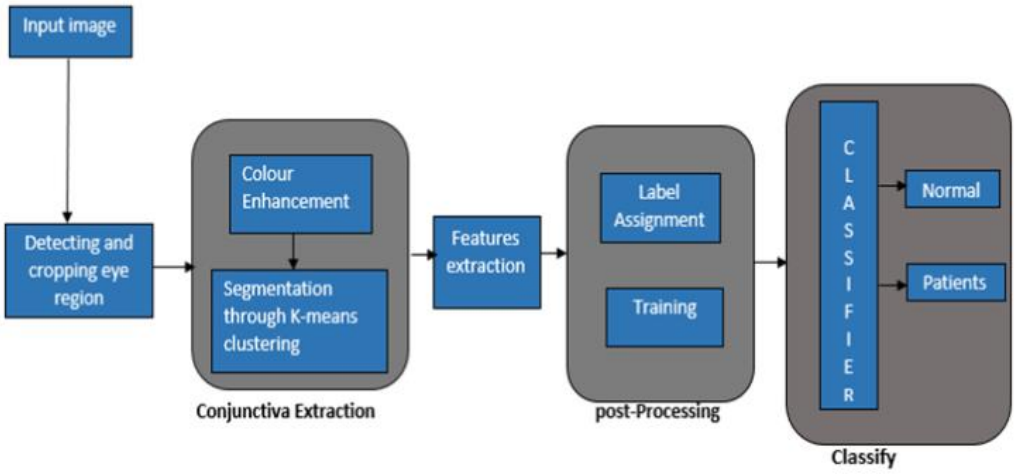

Figure1.1.2: The image classification algorithm used [5]

The image processing algorithm is also used in both the projects has a similar sample as shown in figure 1.1.3 shown below.

Figure 1.1.3: Eye sample [5]

In the sample shown the figure1.1.3 the left side is the healthy sample where the person is healthy, and the Anterior Conjunctiva of the eye shown in the right side of the image is much lighter and the patient in the above image is also suffering from anemia. But in the research component of this project the team shall be using the finger tip as the symptom to detect anemia.

1.2 Research Gap

The main unique part of this solution is that the team have an image detection implemented on a fingertip rather than using an invasive method and detecting anemia. Although the detection of anemia can be done using these symptoms which are identified and subjected to the image processing algorithm is containing a certain accuracy. But the accuracy rate can only be with 80% to 90% as it cannot be achieved at a 100% rate. The lack of data samples which a freely available for the implementation of the machine learning concepts is also a constraint that can be seen in this project.

Although there are many methods of Anemia detection, we shall be using the finger as the source and will be implementing machine learning algorithm which is used for image classification. Although there are many projects that are designed to detect anemia most of them are based on a single symptom or based on the one input for example like taking images beneath the eye or finger tips or ear tips which are main places that indicate that the person is suffering from Anemia. But in the device the team has designed the inputs so that not only one is taken, but many inputs are taken via the app. But in terms of image processing the input that has been processed using earlier projects are the not using deep learning concepts, which means that the accuracy rates are lower than usual. Therefore, an increased accuracy rate is also essential.

1.3 Research Problem

Anemia is a disease that is extremely dangerous because it causes the reduction of iron in the body, which leads to the reduction of Hemoglobin in the blood. This would cause the reduction of oxygen content in the body that causes lower oxygen content in the body. The reduction of oxygen would cause the organs in the body to reduce the amount of metabolism that takes place due to the lack of oxygen. This would lead to organ failure. Thus, identifying the problem is vital.

But we need to develop a proper mechanism to identify the difference between the anemia diagnosed patient and the healthy person. But as there is a lack of data based on the images related to anemia patient a huge data sample collection should be designed. After the data sample is designed a proper algorithm should be built in order to detect the exact differences in the images. If a patient is suffering from anemia, then the detected result should be detected at early stage or else if it's delayed it could lead to organ failure and even heart attack as mentioned above.

The amount of Anemia patients across the country is a massive count. According to statistics of the UN the following can be seen.

Population group	Prevalence of anaemia		Population affected	
	Percent	95% CI	Number (millions)	95% CI
Preschool-age children	47.4	45.7-49.1	293	283-303
School-age children	25.4	19.9-30.9	305	238-371
Pregnant women	41.8	39.9-43.8	56	54-59
Non-pregnant women	30.2	28.7-31.6	468	446-491
Men	12.7	8.6-16.9	260	175-345
Elderly	23.9	18.3-29.4	164	126-202
Total population	**24.8**	**22.9-26.7**	**1620**	**1500-1740**

Table 1.3.1: Anemia percentages in 2008 [6]

Based on the table 1.3.1 there are more pregnant women who suffer from this disease than men. This is mainly due to blood loss that is involved when it comes to women. For example, they undergo pregnancy where a large amount of blood is loss. Similarly, a huge amount is also loss during the menstrual period. These are few reasons for women to be subjected to this disease. In case of a pregnant women both the child and the mother's life are at stake unless the disease is detected at an early stage. Therefore, detection of this disease is a vital part.

The accuracy of detection should also be higher due to the amount of prediction that the device can give based on the machine learning algorithm and that implemented for image processing purpose. The image should have enough information to be distinguish between the healthy patient and the sick patient. Therefore, a proper location should be selected based on the symptoms of the anemia patient. Based on the input the system should be designed. So, one of the main research components is to go through the symptoms of an anemia patient and get the relevant input and feed that into the image processing algorithm so that the detection can happen.
A proper image processing algorithm should be selected in order to feed these data and run. Therefore, out the supervised learning, non-supervised learning and reinforcement methods available under machine learning we should select an appropriate model in order to run it in the server. Under each of these branches there are different methods and selection of one an appropriate model for the image processing is also a vital factor. Because the selected image processing algorithm should be able to detect the difference in the image and also provide higher a accuracy rate.

1.4 Research Objectives

The main goal of this project is to design a device so that the user is able detect whether they are diagnosed with Anemia or not. In this stage the detection of the symptoms of the disease is important. When a patient is diagnosed with Anemia many symptoms can be seen. But one of the key symptoms that can be seen in an anemia infected patient is that the amount of iron reduces which leads to the reduction of oxygen in the blood, makes the blood look pale than red. This is a key factor which we can detect without injecting and extracting blood. Therefore, we are designing a device where the picture of the fingertip, where plenty of blood capillaries are available by sending light and getting the image, then the image would be subjected to image processing where the image would be compared with a patient's image sample and a healthy person. The image that is captured at the end of the device is transferred to the server. Then is subjected to image processing. The main objective of the image processing algorithm is to gain a higher accuracy rate on the comparison and the result to be sent for further processing.

2. Methodology

2.1 Methodology

In order to design the image processing algorithm, we shall be using a deep learning algorithm. In the methodology that is designed we shall be initially fetching the image that is captured at the device end to the server. The server that is used in order to achieve this is the AWS (Amazon Web Service). In the amazon web server, the team shall be hosting a S3 bucket where the input from the device is saved to the location. The device which uses a raspberry pi is configured using the raspberry pi OS where it uses the Linux platform. Then we push the image that has been taken by the camera module to the S3 bucket. Then the image is taken for the process in the server created using an EC2 instance in AWS. We shall be using a machine learning algorithm for the image processing purpose. According to the figure 2.1.1 shown the classification of machine learning is as follows;

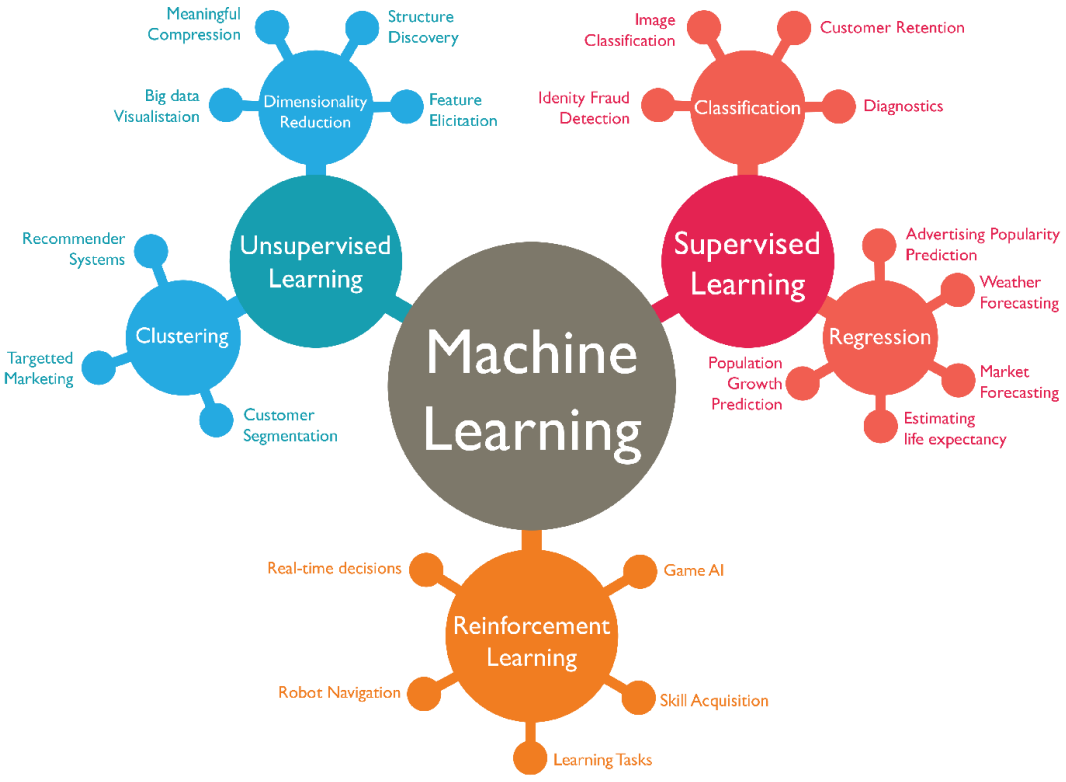

Figure 2.1.1: Machine Learning Classification [7]

In this project we shall be using the supervised learning method. Under supervised learning we shall be using the classification method. In supervised learning the function executes by mapping an input for an output and the classification model is used recognize the image. The image is fed into the process and the training is given to the learning process. The algorithm is then run against the input taken from the device.

In the supervised learning under the classification model we shall be using TensorFlow to build the algorithm and the API used is keras [8]. Keras is high-level API and is subjected to deep learning model. It has certain features like user friendly, modular and composable and easy to extend. The algorithm is build based on the above algorithm. Keras is a model that is used in deep learning where the libraries are also assigned. The input taken from the S3 bucket where the images taken from the device are stored is run against the built algorithm. The algorithm is run from the instance that is created under EC2 in the Amazon Web Server as shown in figure 2.1.2.

The following processes takes place in the image processing.

- Collecting data sample
- Importing the library and split the dataset collected
- Build a CNN
- Build the full connection
- Data Augmentation
- Training the network
- Test

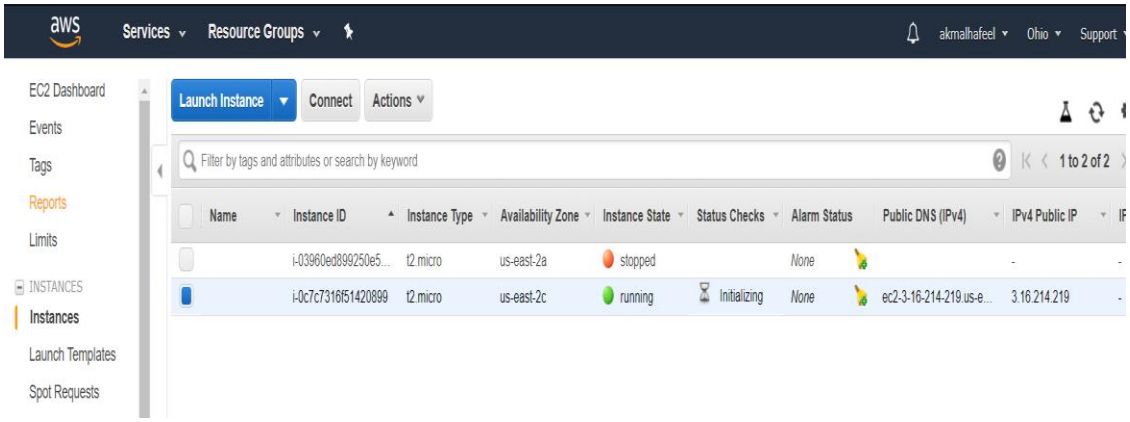

Figure 2.1.2: Instance launched for windows OS

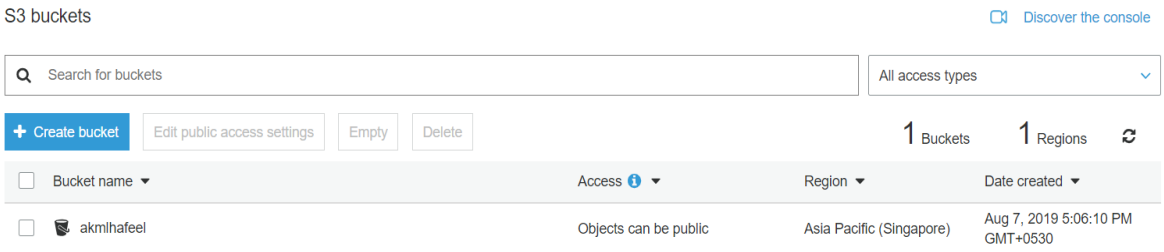

Figure 2.1.3: S3 bucket setup in AWS

Here the instances are setup as it is the main functioning part of the algorithm, where the functional part of the algorithm has defined in the particular instance and the images that are taken from the device are sent to the S3 bucket that has been created for this purpose. The S3 bucket that originally can hold any amount of data as required can be expanded using the AWS server. The server functionality is also fast the server uses SSD in EC2 instance of the server that has been setup.

In order for the continuous functionality of the device without any interruptions the device has to be connected to an instance that is running continuously without terminating. Incase the server terminates all he functional algorithms that are loaded in the server would be lost. Thus, proper server maintenance is needed.

If the cloud implementation is costly, we can also use a local method for implementing the server functionalities. Here we shall run a server instance locally created. A python server instance can be created and then the instance can be run. In order to run this python instance, we need a model of the images trained using the data set be available to use at any given instance. Initially the service should be able to fetch the image from the device after it has been taken and the image needs to be added to the service to be validated against the trained model that has been created using keras and tensor flow. Here the model is already trained using the sample datasets that have been collected using the anemia patient against the healthy patient's dataset. During the training of the data set we have to classify the dataset accordingly according to the defining naming conventions for anemia and healthy patients in order to get proper trained models. In order to train the model, we shall be using a library called Imageai where the keras, OpenCV, NumPy, h5py, matplotlib and TensorFlow are integrated so that they should be pre-installed in the environment where the library is being used. Then after training this model the model is used to compare the image where the TensorFlow technology is used for the image comparison with the model where the service has been made available. Then the python server instance where the image recognition is configured the taken image from the device needs to be mapped.

Initially the image that is taken from the device is sent via Wi-Fi to the location specified in the image processing algorithm. Then the image is run against the model that has been trained and the output is sent to the firebase server that has been connected to the service. This python server shall transmit the relevant data to the collection in the firebase that has been defined and it sends the accuracy rate of the healthy patient. Then the service shall look for the next request to function similarly. This is the method to implement the cloud server service locally as this is ideal for the cost reduction. But the availability of the server is local and would be available after it is manually triggered.

2.1.1 Collecting data sample

The primary requirement to set up a machine learning algorithm is that the exact data sample should be collected. In this project the samples collected are used to build the algorithm and the train the Keras model that has built. The data that is feed into the server is used to build and train the model to arrive at closer and exact results. The data samples collected are shown below.

The data sample of a healthy patient;

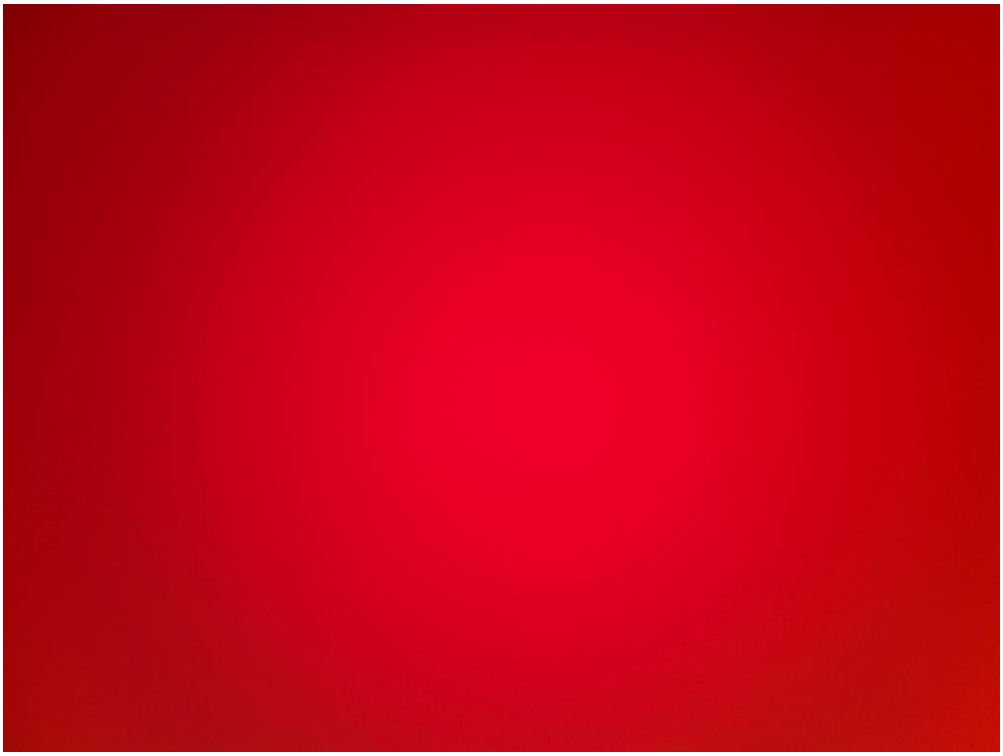

Figure 2.1.1.1: Data sample of a healthy patient

In the above image there is a greater intensity of red than of an anemia patient. This is mainly due to the amount of richness of Fe3+ ions in the blood which gives the intensive red color to the blood. This makes the blood more brighter and reddish compared to that of an anemia patient. Therefore a clear distinguish factor can be seen between the anemia sample as well as the healthy sample.

The data sample of an anemia patient can be seen in figure 2.1.1.2

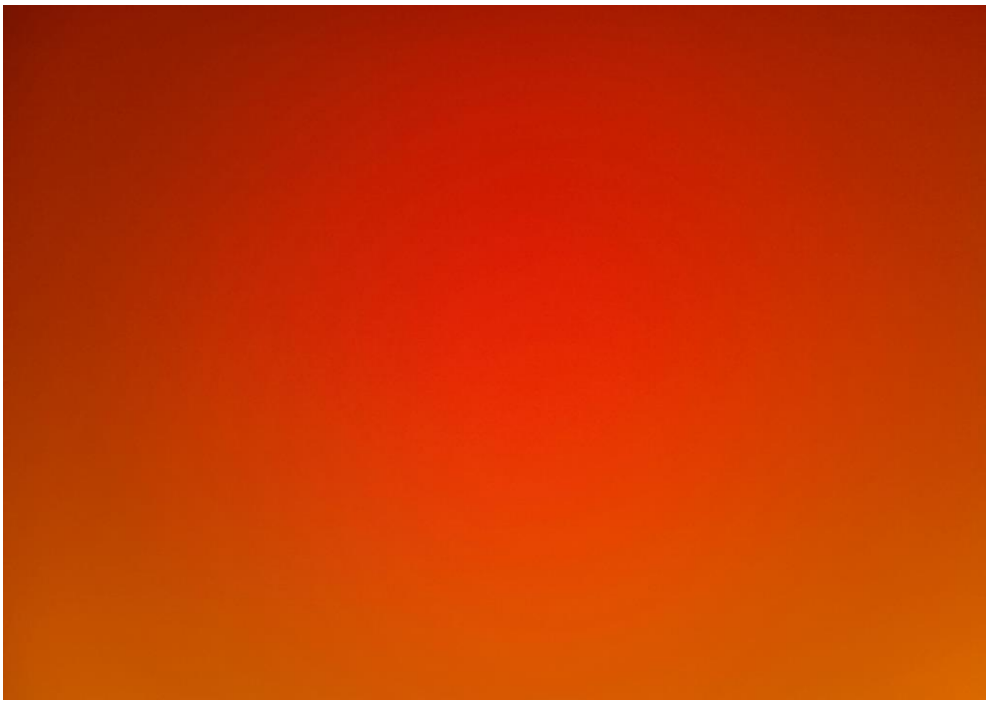

Figure 2.1.1.2: Image of anemia patient

According to the different data samples taken in Figure 2.1.1.1 and 2.1.1.2 there is a clear difference in terms of color. There is a yellowish color that can be seen in the anemia diagnosed patient. This is main difference in an anemia diagnosed patient is measured using the image processing algorithm that is designed.

This yellowish color in the skin is caused due to an effect in anemia which is the reduction of iron in the body. Iron is the main component which gives red color to the blood and the amount of iron reduction leads to the anemic condition in the body. The Red Blood Cells get the redness due to the Fe 3+ ions present in the red blood cells. Therefore, any individual suffering from this disease would have a plain skin where when a flash is sent through the image can be captured as shown above in the data samples and been fed to train the Keras model that is available in the server.

2.1.2 Importing the Library and split the dataset collected

If the model that was proposed is to be built, then the libraries related to Keras should be imported to the file. The following libraries should be called for this purpose.

```
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
```

Figure 2.1.2.1: Libraries that are imported to run the module

The libraries that are indicated in Figure 2.1.2.1 should be imported if the image processing methodology is properly run. These are dependent on the Keras API built under TensorFlow.

After the libraries are imported the data that has been collected should be divided into two parts as anemic and healthy. And this should be run accordingly to train the model.

2.1.3 Building a CNN

In the keras model the features from the input image is extracted, this is also known as convolution. Here the input image is learnt and results in preserving the relationship with the image by learning the pixels using small square inputs.
Then the image is subjected to polling where the subsampling or reduction of dimensionality of each feature in the feature map after the image samples are properly maintained.
After this the polling there is method called flattening. At this point the matrix formed is converted to a linear array so that the processing can be a easily run.

2.1.4 Building Full Connection

At this stage the convolutional network is converted to neural network. After this is done the network should be compiled to check for errors.

2.1.5 Data augmentation

At this stage data augmentation should happen. The data that we have should be set in different positions so that the training when occurs the model trained would be able to get the input in any different manner, yet the single output is given. This would also increase the amount of data that is available as well as make sure the accuracy rate increases.

2.1.6 Training the Network

After the data is set the model should be trained and the training should happen in an efficient manner. During the training session of the data set make sure the process keeps running even though the power drops are even expected. During the running of the training model we should make sure that ample RAM is available as well as a good GPU because the training is related to the video card as well.

2.1.7 Test and Validation

Using a random image of an anemia patient or a healthy patient which was not in the data sets that was used to train the model should be tested against the model that has been created and the accuracy rate should be checked. In order design a proper method for the validation of the image that is taken against the model that is created from the trained model using the test samples.

If the local instance is used as the method to process the device input, the method that is configured for the device to ship the output to the image processing algorithm

is using the Wi-Fi network. As the device is in the same network as the image processing algorithm that is implemented in the local server the image can be transferred locally pointing at a certain location in the server which can be used to store the input of the image taken from the individual. In order to do that we shall be suing a common location in the server and when the image is stored at that location the image processing algorithm fetches the image and subjects the image for image processing and validation. The image processing algorithm has a bat script written to fetch the image from the location and save it in the name called “sampe.jpeg” as taken is run against the model and the image is deleted when the next input is taken as this would be stored in the same name.

But the validation system should be always listening for an input so that the device is able to detect the input and trigger the validation algorithm. In order to do that we shall be having the firebase input configured to listen for an input. The firebase database is configured with a specific collection so that the device can send an input Boolean value to the database from the button click function. In the database the entry given would be false and when the button function in the device has been triggered then the device would send a Boolean value called true to the database. The python service that has been running in the validation algorithm waits for the value to be true in the location in the firebase database. When this change is detected in the firebase database the image is fetched from the device and tested against the model that is generated from the image processing algorithm used. In the image shown in figure 2.1.7.1, the ‘start_image_processing’ filed defines the triggering of the algorithm or not. If the particular filed is false, then the algorithm in the image processing component would not trigger the validation part of the code. But if the value of the field ‘start_image_processing’ is turned to true then the algorithm for validation would be triggered, this is designed using a python service for the usage to be simpler.

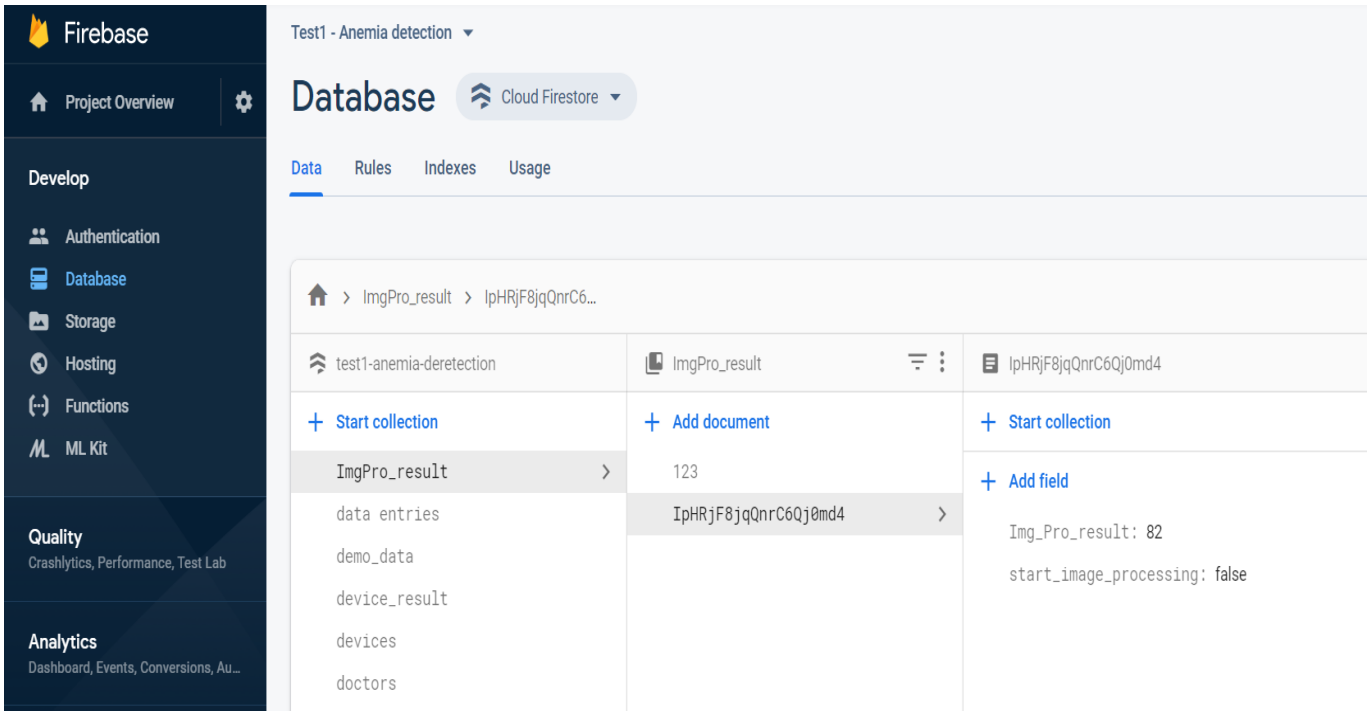

Figure 2.1.7.1: Firebase database collection image

After the image processing algorithm has been run against the image that has been inputted by the user the results would be pushed for the Img_Pro_results that has been created in the firebase server. The server is also configured so that the changes from the output after the validation step would be transferred to the Img_Pro_results field whilst deleting the exiting value in the database server.

The value sent to the server is the healthy percentage of the person. Then the information is fetched by the machine learning algorithm to detect the changes from the image processing after saved is pushed to the machine learning algorithm that is configured in the azure machine learning portal.

2.2 Commercialization aspect of the component

This project was built on a commercialization concept. The sugar machine although requires an invasive method it has been able to capture a larger market due it's accuracy rate. If the accuracy rate is high, then the accuracy of the product can be increased. In order to increase the accuracy rate of image processing the team has used the following concepts where the keras deep learning method is used where the accuracy rate is higher than usual method. The symptoms that is also taken during the construction of the device also needs to be taken into account when it comes to the commercialization of the product. The device only inputs one symptom while the other component of the total system which this the app inputs some information regarding the anemia patient is, where the input is taken from the questionnaire where the user has to answer the relevant questions. This also increases the accuracy of the device further in the process of detection of anemia.

2.3 Testing and Implementation

In this stage after the complete design of the device it should be tested to make sure the accuracy rate of the device as well as ensure the functionality of the device. Therefore, the device should be taken to patient where the user is suffering from anemia which is proven by a physician through a proper medical report and test the accuracy rate of determining anemia. During the implementation there can be drawbacks due to the bugs that can occur. But the fixes should also be done as soon as possible and make sure the device is functional and also be guarantee that the information that is obtained taken trough the device is also kept securely because the medical information are sensitive information therefore the proper security need s to be provided.

The data during implementation should be stored in the server and stored so the accuracy can be increased as well as security.

3. Results and Discussion

3.1 Results

In order to test for the results a sample from the healthy patients as well as anemia patient should be tested and looked for the accuracy as well as make sure those samples can be identified against the trained model and the model that has been trained has a lot of data from the healthy patients being used. Therefore, the accuracy rate of the healthy patient should be high compared to the accuracy of the anemia patient. The following sample shown in image 3.1.1 is used to test for the healthy patients and the following results were obtained as shown in image 3.1.2.

Figure 3.1.1: Healthy Patient Sample

In the above put into testing we get the following results as accuracy rate. This is mainly due to the amount of training data sets used for healthy patients. The result that is obtained is 98% healthy as shown on figure 3.1.2.

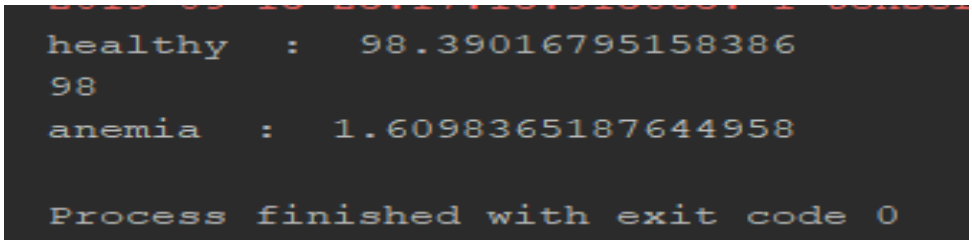

Figure 3.1.2: Results after image processing

Next the sample of anemia patient is tested against the developed algorithm. Then the output accuracy can be checked. The sample used to test is shown in figure 3.1.3.

Figure 3.1.3: Figure of Anemia patient sample

When this sample in figure 3.1.3 is tested against the developed model and the accuracy is 90%. The results are shown in figure 3.1.4 where the accuracy is 90% and this is achieved by training similar data samples of anemia patients and usage of deep neutral network libraries.

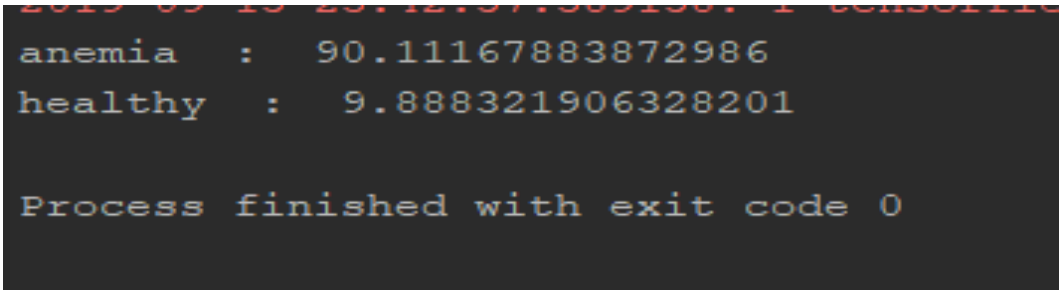

Figure 3.1.4: Output of anemia patient sample

3.2 Research Findings

During this research there were certain constraints that can be faced as mainly it is the lack of data availability that would limit the image processing from having a perfectly trained model. As the amount of data availability is low, it should be manually collected by visiting to the hospitals and making sure the appropriate patients are selected and the device developed for the project is used to collect the data samples that is used to design the model or else the accuracy level would differ if the samples are taken using some other resource which could alter the accuracy of the final results. Therefore, the data collection is an important aspect to be noted during the research.

The proper organization of the data collected is also an important factor to note during the image processing component is designed. Because during the training of the model's proper naming/labels are looked into as the model used is keras it also mainly learns the image as well divides it based on the labels that are assigned in the structure designed. The images that are used for the training and testing purposes are two different set of data samples and the same should not be used when designing the model. During the data collection, the information collected should also be kept safe as the data are sensitive data and that information should be kept confidentially.
If the server is hosted in the cloud, then a cost factor would be attached and in order to reduce it the services can be implemented locally in a server instance.

3.3 Discussions

In the image processing algorithms designed the technologies used are mainly keras and TensorFlow which are integrated to a single library called Image Ai. In this library the following technologies are also integrated which are OpenCV, h5py, NumPy, matplotlib and keras with TensorFlow. Here the TensorFlow plays a major role in the validation part of the image. At the learning stage of the image the image is classified based on the color differences that can be seen in the images that are shown in the figure 2.1.1.1 which indicates healthy patient and figure 2.1.1.2 which indicates the anemic sample. The most distinguish factor in the image is the yellowish in the image. This is caused due to the lack of Fe3+ ions in the blood.

After learning the image, a model file is created, and the model file is used to compare with the sample image that is taken for validation where the sample image indicates the image inserted by the user of the device to get the input from the patient. After the processing has been done the output is pushed to the firebase database collection where then the output is fetched by the machine learning algorithm.

4. Conclusion

The data sets that has been collected needs to be properly clustered into proper groups as anemia patients and healthy and they should be trained eventually. Keras and TensorFlow used to design the model has deep learning which increases the accuracy rate of detecting anemia using image processing. Therefore, the image that should be taken for image processing is needed to be in sequence with the quality of the image that has to be with same quality of the device.

The image can be identified using the above-mentioned processes whether it’s an anemic image or healthy image based on the color that can be detected using the designed model and the output is sent to the server for further processing. As it’s a prediction related image processing the most accurate method of training the sample is using keras and TensorFlow.

References

[1] WebMd, "Understanding Anemia -- Symptoms," WebMd, 22 July 2017. [Online]. Available: https://www.webmd.com/a-to-z-guides/understanding-anemia-symptoms#1. [Accessed 8 August 2019].

[2] N. H. M. S. M. Mazalan, "Automated Red Blood Cells Counting in Peripheral Blood Smear Image Using Circular Hough Transform," IEEE, Johor, 2013.

[3] WebMD, "Understanding Anemia -- the Basics," WebMD, 14 June 2018. [Online]. Available: https://www.webmd.com/a-to-z-guides/understanding-anemia-basics#1. [Accessed 8 August 2019].

[4] C. S. J. M. S. S. S. U. Z. M. M. I. A. I. S. A. F. C. S. Azwad Tamir, "Detection of Anemia from Image of the Anterior," IEEE, Dhaka, Bangladesh, 2017.

[5] M. U. A. S. A. S. M. K. Attia Irum, "Anemia Detection using Image Processing," DIPEWC2016, Dubai, 2016.

[6] WHO, "Global anaemia prevalence and number of individuals affected," WHO, [Online]. Available: https://www.who.int/vmnis/anaemia/prevalence/summary/anaemia_data_status_t2/en/. [Accessed 08 August 2019].

[7] R. Khadka, "Machine Learning Types #2," Towards Data Science , 8 September 2017. [Online]. Available: https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1. [Accessed 8 August 2019].

[8] tensorflow, "Keras," tensorflow.org, 2019. [Online]. Available: https://www.tensorflow.org/guide/keras. [Accessed 08 August 2019].