# Detection of Malaria from Blood Smear Using Deep Learning


Vignesh Chivukula
*Department of Electrical and Computer Engineering*
*Stony Brook University*
*New York, USA*
vignesh.chivukula@stonybrook.edu

Nitin Asthana
*Department of Electrical and Computer Engineering*
*Stony Brook University*
*New York, USA*
nitin.asthana@stonybrook.edu

Meghana Jayaswamy
*Department of Electrical and Computer Engineering*
*Stony Brook University*
*New York, USA*
meghana.jayaswamy@stonybrook.edu



***Abstract* – The aim of this project is to develop the detection of the malarial disease using the blood smears of the samples quickly using Deep Learning. Currently, malarial detection is done by using the blood smears of the patient by a clinician manually and this process takes up 2-3 days which is quite a long time. Rapid Diagnostic Tests (RDTs) have been developed, which are quicker but are not very accurate. So, to improve the speed and maintain the accuracy, the project detects malaria using Deep Learning models. Here the models are pre-trained and then this model is used to detect the presence of malaria in the test data, with a better accuracy of around 97% and speed.**



***Index Terms – Malaria, RDTs, RBC, Anapholes mosquito***


## 1. INTRODUCTION

Malaria is a human disease which is caused by parasite of protozoan virus and of the genus Plasmodium which is mainly transmitted by the bite of infected female Anopheles mosquitoes and that infect red blood cells (RBC). [1] It is seen that most of the deaths occur among children accounting for one death every minute from malaria, and it is also seen that malaria is a leading cause of neuro-disability in childhood. It is very important to detect the malaria disease for treating the patient on time and also to prevent further spread of infection. [2] The symptoms of malaria typically develop within 10 days to 4 weeks following the infection. The common symptom of malaria includes very high fever, profuse sweating, chills, headaches, nausea, vomiting, anemia, muscle pain.

Fig.1 Female Anopheles mosquito

It can turn out to be a life-threatening disease and it is very important and helpful to detect malaria in the initial stages. In 2017, there were an estimated 219 million cases of malaria in 87 countries with the death number of 435000 in 2017. Total funding for malaria control and elimination reached an estimated US$ 3.1 billion in 2017. Contributions from governments of endemic countries amounted to US$ 900 million, representing 28% of total funding. [3] In most cases, malaria is transmitted through the bites of female *Anopheles* mosquitoes. There are more than 400 different species of *Anopheles* mosquito; around 30 are malaria vectors of major importance. Malaria disease takes place in a cycle where an infected mosquito bites a normal human being, upon the infection the virus travels to the liver and infect the red blood cells (RBC) where the virus rapidly multiply in their number. Whenever a normal mosquito bites an infected person the virus transfers from the human to the mosquito and the cycle repeats as shown in fig. 2.

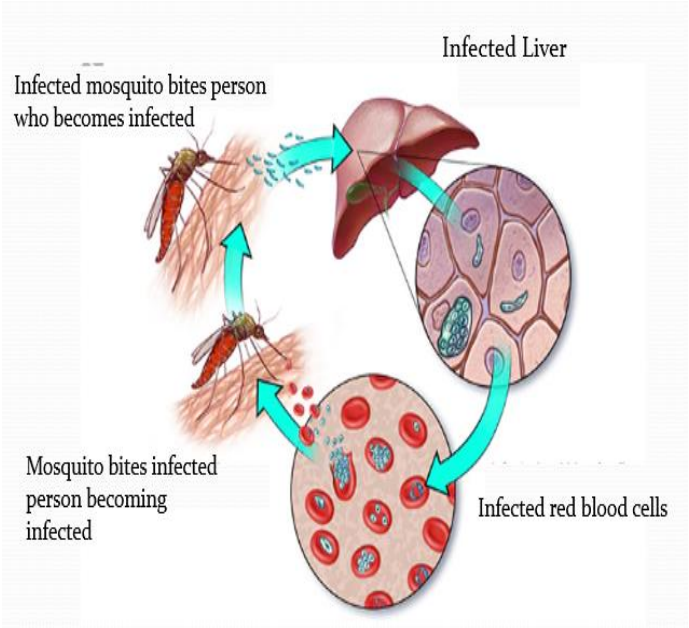



Fig. 2 Stages of Malaria

The aim of this project is to develop the detection of the malarial disease using the blood smears of the samples quickly using Deep Learning. Currently, malarial detection is done by using the blood smears of the patient by a clinician manually and this process takes up 2-3 days which is quite a long time. Rapid Diagnostic Tests (RDTs) have been developed, which are quicker but are not very accurate. So, to improve the speed and maintain the accuracy, the project detects malaria using Deep Learning models. Here the models are pre-trained and then this model is used to detect the presence of malaria in the test data, with a better accuracy of around 97% and speed.

## 2. RELATED WORK

Image based identification and classification of malaria parasite infection involves a sequence of operations viz. Image acquisition, Preprocessing, Erythrocyte segmentation, Feature extraction and Classification [4]. Image acquisition is the process of collecting malaria blood smear images from digital camera attached to a light microscope. Preprocessing removes unwanted noise present in the acquired malaria parasite image for better visualization and further analysis [5]. Segmentation is a process of isolating red blood cells (erythrocyte) and removing other details such as white blood cell, platelet and artifacts from the preprocessed image. Feature extraction extracts insights from the erythrocytes like color, morphology, geometry and texture. Extracted features are classified as infected and non-infected erythrocytes using machine learning algorithms. Generally images acquired from thick smear are used to identify malaria infection and thin smear is used to find the type of malaria species and its respective stages [6]. Several techniques have been discussed for preprocessing blood smear images to improve spatial quality of images. May et al. used Median filter for reducing the impulse noise and also he suggested Wiener filter to reduce the additive noise to invert blurring of images . Arco et al. employed Gaussian filter to effectively remove Gaussian noise from the images. Das et al. implemented Geometric mean filter to reduce the Gaussian noise and preserve edges in microscopic images [1]. Savkare et al. used Laplacian filter for smoothening and edge enhancement in the images. Soni et al. employed Susan filter to preserve image structure and to improve quality of the images . Diaz et al. used low pass filter to remove the high frequency intensity by averaging the intensity of the pixels in the image. Suradkar suggested adaptive local histogram equalization techniques for contrast preserving the low resolution images. Tek et al. utilized Grey World Assumption for illumination correction, used to rectify mistakes in the staining process [6]. Abbas et al. suggested the histogram matching algorithm to normalize intensity of the pixels. May et al. suggested partial contrast stretching algorithm for increasing contrast of the images. It has been observed from existing literature is that frequently used noise removal methods are median and geometric mean filter to enhance the quality of microscopic images. Furthermore contrast Multimedia Tools and Applications stretching techniques have been used popularly to improve the contrast. For illumination correction commonly used method is Grey World Assumption. Erythrocyte segmentation is a subjective analysis; most of the studies elaborate several types of segmentation techniques. Savkare et al. used watershed algorithm to find overlapping cells on each connected components. Suradkar implemented edge detection algorithm to significantly differentiate between foreground erythrocyte and background image. Soni et al. employed granulometry method to effectively identify subject-of-interest objects, which is regular in size. Ma et al. used circle hough transformation to find the radius of the erythrocyte. Das et al. implemented marker controlled watershed segmentation algorithm for effectively identifying overlapping cells. Khan et al. used K-means clustering algorithm by selecting the initial centroid and found the intensity distance to segment the Red Blood Cell (RBC). Chayadevi et al. emphasized fuzzy rule based system to segment the RBCs. Makkapati et al. suggested Otsu thresholding to get the binary image mask, it is a cluster based algorithm to decrease the intra class variance and maximize the inter class variance. Damahe et al. suggested Zack threshold to differentiate between erythrocytes by finding the difference between the highest histogram values and the lowest histogram values. Otsu thresholding in combination with morphological operations is a frequently referred technique because of its simplicity. But watershed marker controller segmentation algorithm is more suitable to segment the touching and overlapping blood cells.

## 3. MOTIVATION & CHALLENGES

It can turn out to be a life-threatening disease and it is very important and hence it is very helpful to detect malaria in the initial stages. Existing diagnosis for the malaria disease includes multiple techniques. As mentioned earlier the initial symptoms of malaria is high fever, chills, sweating, nausea which are often not specific since it can be found in other diseases too. Additionally the physical findings are also not specific. Clinical findings should always be confirmed by a laboratory test for malaria. The gold standard test for malaria disease detection is the microscopic diagnosis where the malaria parasites will be examined with the help of patient's blood samples under a

microscope. Once when the blood samples are obtained from the patient it is stained with a reagent so as to give a unique appearance for the parasite. However this gold standard technique requires atleast 2-3 days for the results to be out. On the other hand various test kits are available to detect antigen which is derived out of the malarial parasites. These Rapid Diagnostic Tests (RDTs) are very helpful in cases where a reliable microscope is not available. Though this test provides the result in 2-15 minutes, the obtained results are not very accurate and sensitive. Meanwhile the nucleic acids present in the parasites are detected with the use of Polymerase Chain Reaction (PCR). Although this technique may be slightly more sensitive than smear microscopy, it is of limited utility for the diagnosis of acutely ill patients in the standard healthcare setting. Also the results from the use of PCR techniques are often not available quickly in order to establish the diagnosis of the malaria disease.

The main challenge for the proposed work is to identify the exact species which causes malaria disease, there are several other species as shown in fig.3.

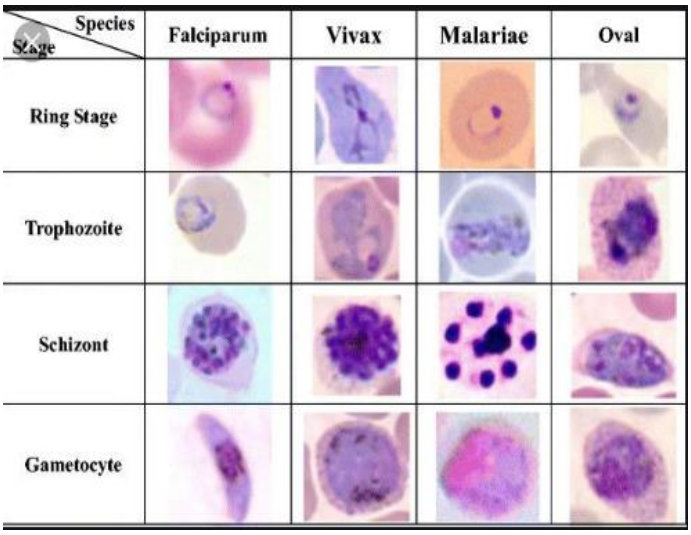



Fig.3 Different species of malaria parasite

Secondly, the gold standard test which includes the blood samples from patients requires only the red blood cells out of the blood samples. The human blood mainly contains four different components which are named as: red blood cells (RBC), white blood cells (WBC), platelets and plasma.

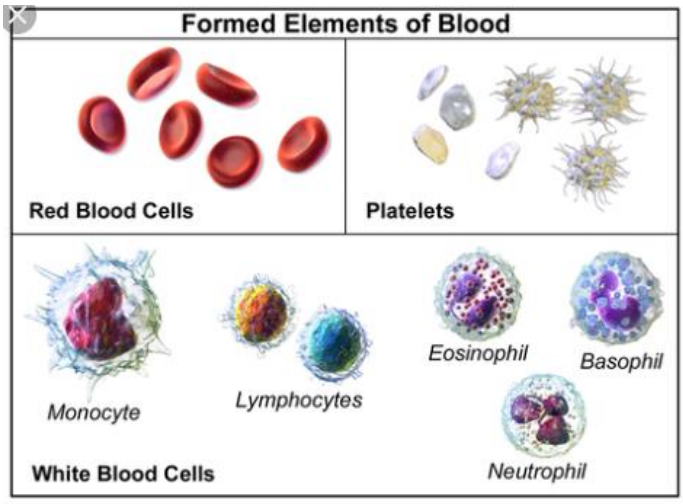



Fig.4. Components present in blood

Separating red blood cells from all the other components of blood can only be done with help a trained medical-practitioner. Also the staining of the blood requires a quality reagent for the samples to give a unique and distinctive appearance under a microscope.

## 4. SYSTEM OVERVIEW:

In this project, we obtain a labeled dataset of images containing both affected and unaffected samples to train a Deep Learning Model. Based on this trained model, the test images (real time images) are processed and prediction is done, providing the final result.

### 4.1 Proposal:

The basic idea of this project is to detect the presence of malaria infection in the blood as quickly as possible in an efficient manner. The RBCs, from which the detection is made primarily, is processed to determine the presence of the parasite.

Initially, the blood smears of the patient are taken and are stained using one of the stains such as Giemsa (most commonly used) or Leishman's stain. Once the stain is done, the smear is then placed in the microscope, zoomed to very high level of 100x for clear visuals. These visuals are processed by image processing techniques to obtain the images of single RBC cells, which are classified by our model to detect the presence or absence of parasites. This classification is done in a very quick time, compared to few days taken by normal lab analysis, which helps to notify the patient regarding the diagnosis and inform them of the immediate action required. This quick response helps prevent the spread of the disease and the effect of the infection getting intense. Also, the local health authorities

are informed so that they can assess the situation and prevent the further spread of the disease.

### 4.2 Dataset:

The dataset used for this project was a collection of 27k images, containing both affected and unaffected samples, which were labelled. Knowing the ground truth of the images helps in easy training of the model. These images were taken from an open repository, https://lhncbc.nlm.nih.gov/publication/pub9932 and then processed.

### 4.3 Dataset Processing:

Once the dataset was obtained, next step was to process the data into different datasets, such that each of them can be utilized for a certain purpose: Training, Validation or Testing. Of the entire dataset, a major part, 80% of the data was primarily used as Training data. This split included the 80% of both the Parasitized images and the uninfected images. Remaining 20% of the images was used as testing dataset. Of the training data, 10% was used to validation dataset, which was used to check the accuracy of the model during the training process. This calculation of accuracy during the training made sure that there was some regular updates during the Training process, thereby helping in determining the efficiency of the training model.

### 4.4 Deep Learning:

Deep Learning is a subset of Machine Learning and is based on the concept of Artificial Neural Networks. It includes various architectures like Deep Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, etc. and was mainly used for applications like Image Processing, Computer Vision, and Speech Recognition. All these architectures are modelled similar to that of a human brain containing neurons that are connected to one another. Here, there are many layers of neurons that are connected with one another, starting from the Input layer to the Output layer. These layers are used to process the input data and provide the possible output as predicted the network, in the output layer. There might several hidden layers in the middle, but at least one layer must be present.

In this project, we have used the Deep Neural Network for training the model to process the input image and determine the output. We have used the TensorFlow Framework as a lower level framework for backend and Keras for building the model and training it.

### 4.5 Training Model:

Training the model does not involve just loading the training data (images) but pre-processing them into a standard format and then training them. So, the initial step in pre-processing is to augment the data, so as to make it more robust while training. Augmentation refers to shifting the image in different angles and using each of them as a training data, thereby creating new training data and increasing the amount of training data. This makes sure that different variations of the model would be seen by the model. It is performed using the Keras framework ImageDataGenerator class and produces various version of the image. The same method is applied even to the validation data, since the accuracy during the training is measured using the validation data and the model is robust enough to predict the augmented images too.

The next step is to create the training generator, which is used to feed the data into the model. The images are processed into RGB format and then resized into a standard size of 64 * 64 pixels. Also, the number of images are too high to be sent in a single iteration and so they are split up into a number of batches. The Batch Size used here was 32 and so, for the training dataset, there was 620 steps in total for a single training iteration. It just refers to the fact that, exactly 32 images are sent for training at once and to complete the training of all the images present in the dataset, 620 times the iteration is needed to be performed. Similar to that of the training generator, validation and testing generator are developed for running the validation and testing steps respectively.

Once the data pre-processing is done on the images, the next step is to develop a training model and compile it. Generally there are quite a number of training models present in Keras to peform the training. For image processing, there is an efficient neural network called ResNet (Residual Neural Network), which provides high amount of training efficiency for image related models. The main advantage of using ResNet is that, it avoids the negative outcomes that occur during training, making sure that there is a high accuracy and quicker training of the models, when compared to other traditional neural networks like CNN.

### 4.6 ResNet Architecture:

The ResNet model used in this application uses a 64*64*3 images (3-channel RGB images). The output obtained here is binary, as either affected or unaffected. In the model, there are 3 layers involved and in each layer, there is a stack of 3 Residual modules. For each layer, there are some filters present, after which the spatial dimensions are reduced for the images. The dimensions of the images, which might be 299*299 or 274*274 are reduced to a common size of 64*64 pixels.

There is also a BatchNormalization that is being done on the images, wherein the images are standardized such that they have a zero mean and a unit standard deviation. Then there is an Activation function that is used to obtain the response of the layer. This is used since the output is binary. After the activation function, Max

Pooling is performed to reduce the spatial dimensions of the image, as required. These steps are performed for each stack and the final model is obtained, which can now be compiled.

After the compilation is done, the successive step is to make the model ready for training. To start the training, two main parameters, namely the Number of epochs and Learning rate, have to be determined. The Number of Epochs refer to number of times the training needs to be performed on the model so that we can obtain a high accuracy trained model in the end. The Learning rate is kind of an hyperparameter, which determines how quickly the model gets adapted to the problem. Usually the learning rate is kept at a optimal level, not too small or large because higher learning rates make the model to train sub-optimally and lower rates might make it get stuck at one place and the time period of training goes very high. Thus we used a learning rate of 1e-1 and the number of epochs as 20. Also, there is a polynomial decay function devised to determine the different learning rate for every epoch, so that the learning rate is kept optimal for every increasing epoch.

Once all the parameters necessary for training the model are set for optimal values, the model is now made to start its training process. This training process continues for the total number of epochs provided in the application. For every epoch, there is an accuracy and loss determined for that particular epoch. The accuracy values provides the information on how much the model is accurate after the training of that epoch and also provides the information on the loss faced during that epoch.

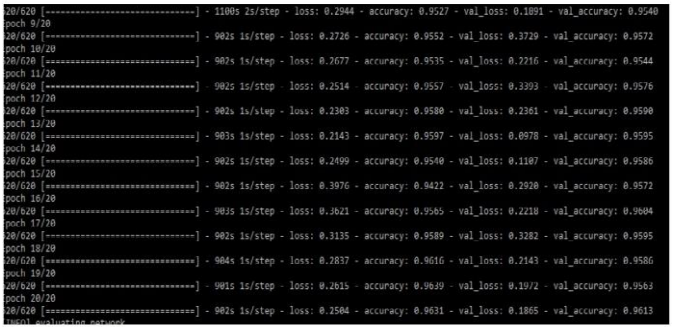



Fig. 5 loss and validation accuracy

It is seen that the accuracy is low in the initial epochs but it goes on increasing for the higher epochs. This is suitable for both the training and validation accuracy values. Also, it is seen that the training and validation losses are high during the initial epochs but they gradually reduce to a small minimum value as the number of epochs increase. Thus we are able to train the model to a good amount of accuracy of 96.5% on the Training data and 96.7% accuracy on the Validation data.

The entire training process is a computationally heavy one and it depends upon a very high capacity of hardware like GPUs to complete quicker. Here the model took around 5 hours to complete the entire training of 20 epochs.

This training is not practically feasible during the real time diagnosis because the person testing for the disease cannot be made to wait for a few hours before every detection. To avoid this problem, we store the trained data as a pickle model. The pickle is a library that is used to store the array data in a serialized way. Using this kind of models would positively increase the speed of the application by a very huge margin. This is because, everytime there is a need to diagnose the images, and the trained and stored model can be loaded again to perform the testing. This significantly reduces the time required for training, thereby helping in performing the detection in a very small span of time.

## 5. IMPLEMENTATION

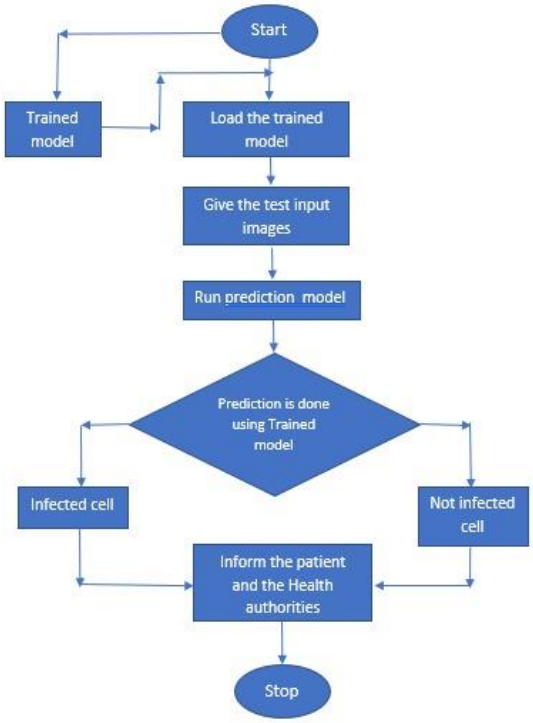
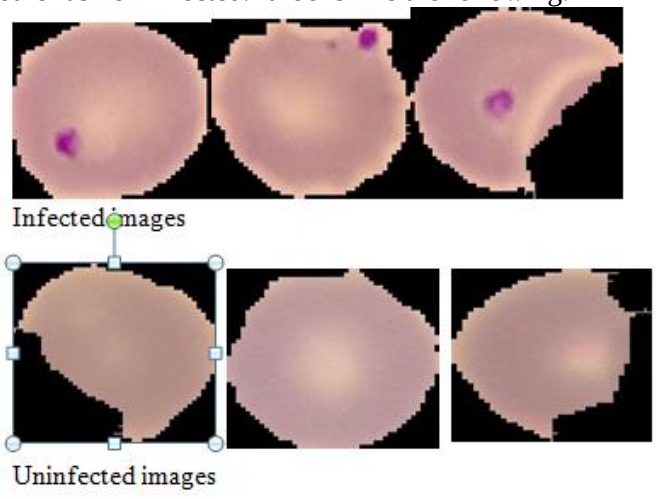



Fig. 6 System implementation

So, the first step in the system implementation is the load the test images of the data set. The loading of the images in the are done using the python library import pickle. The pickle mainly has the two methods. One is to dump the objects into the file objects and the other is to load the objects into the file object. In the previous section we have already told how we have stored the trained data model. So, basically using this library we retrieve the trained data

model. Next, we access all image path as each image in the data set has separate distinct image path and it is necessary for us to be able to access all the images stored. So, when we give the test image input, it will access all the image paths and gives us the output. Now once we have given the test image input, it will access every image path. In this section we do certain actions as well. First, we pre-process our image by converting it from BGR to RGB channel reason being as our Keras model was trained on RGB ordering only. Second, we must resize the image into the 64 X 64 pixels and finally we scale the pixel intensities. It mainly means that intensity is 0 we will ignore the image and if it is 1 we will consider that image. Finally using all these we predict the test image as infected and non-infected image. We have assigned the indexes to the predicted output as index 0 and index 1. The index of 0 is the infected label while the index of 1 represents the non-infected image. Finally, we put the prediction in the results, and we display the result in the application. Now the based on the detection output a proper feedback is given to the user. The feedback is given in the form of an email to the user. To give out this feedback in the form of an email we import the import smtplib library. This library allows us to send the email from difference messengers like g-mail, yahoo mail etc depending on the mail id we are using. We have also attached the text files in this to what message must be given to the user who has been detected with infected cell and who have been detected with non-infected cell.

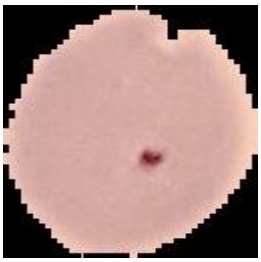
There are two types of cell inputs we have one infected and other as non infected. It looks like the following:




### 5.1 Detection and response:

A test image is provided, and the detection is performed using the above method. Based on the detected value a feedback was sent to the patient and the concerned authorities. The feedback is sent to the patient to let him know whether he is infected with malaria or not so that he can take the necessary future steps. And the feedback is given to the local authorities so that if the maximum number of people in their locality is being affected by the malaria, so necessary precautionary steps can be done in order to reduce it from spreading it to the other people living in the area.

One of the test image for which we tested the application is given below following which the feedback to the user is also shown.



Test image

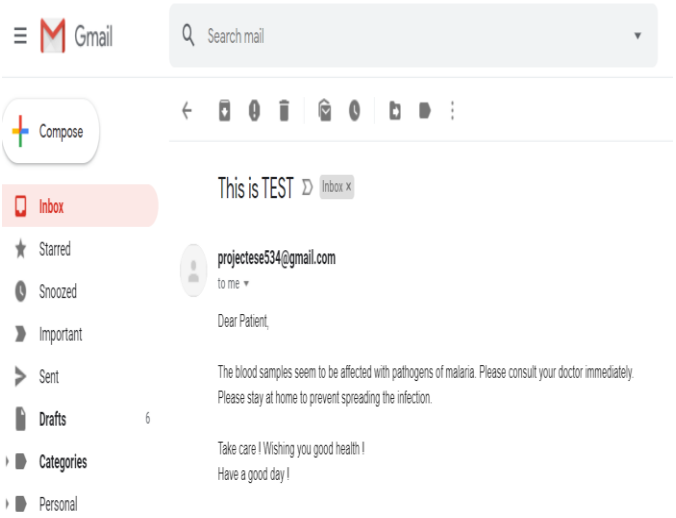



Fig. 7 Feedback mail

## 6. SYSTEM EVALUATION

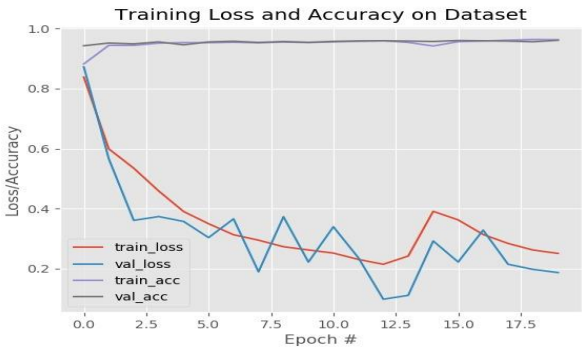
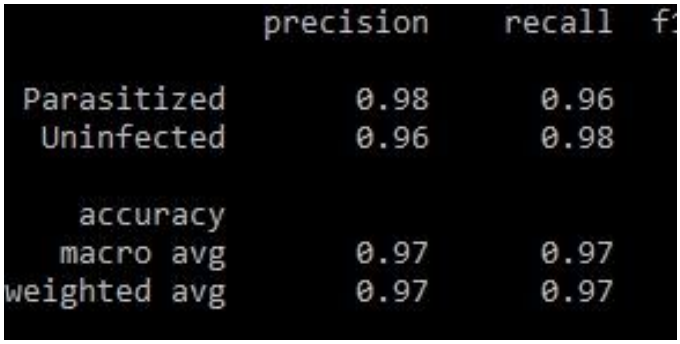



Fig. 8 Training loss and accuracy

The graph represents the loss / accuracy verses number of epochs. We can easily see from the graph that that initially when we start training the model the number of epochs is very less. Currently the loss is very high. But the training is repeated i.e. as the number of epochs increases the loss decreases over the course. Next, we can also see that our system gives us the very high accuracy from starting till the end. So, we can say that overall application gives the correct results maximum number of times.



| | precision | recall | f1 |
|---|---|---|---|
| Parasitized | 0.98 | 0.96 | |
| Uninfected | 0.96 | 0.98 | |
| accuracy | | | |
| macro avg | 0.97 | 0.97 | |
| weighted avg | 0.97 | 0.97 | |

Fig.9 accuracy

from the fig. 9, we can see that the application is able to detect the infected image for 98% times correctly and non-infected image correctly for 96% times with the overall average accuracy of 97% for this model.

## 7. RESULTS

Firstly, the model provides us a malaria detection application with the overall accuracy of 97%. Next The accuracy of the model increases as the number of epochs increases. For the initial epochs till number of epochs equal to 2, the accuracy is low (around 80%) and it gradually increases till 97%. We have loaded the model and saved it in a file. So, the detection is very quick (around 2 minutes) for each test input.

## 8. CONCLUSIONS AND FUTURE WORK

The project shows how the Deep learning can be used to detect the infection of malaria in blood cells. The images are taken from a high definition smartphone camera and microscope. Prediction is performed on these images at a very high speed as quick as around 2 minutes for each test input. The results are obtained within a few minutes, which is very much quicker compared to traditional methods used currently

- Running the deep learning applications are computationally heavy and requires the huge number of processors and gpu's, so we can implement the application using lighter frameworks so this model can be directly implemented in the smartphones applications directly.
- Determination of the exact species of the malaria causing parasite can be done. Human malaria is caused by four different species of Plasmodium: P. falciparum, P. malariae, P. ovale and P. vivax. So exact information of all the four type can be done by knowing more information of the images.
- We can also improve the efficiency to obtain even higher accuracy by improving the quality of the camera and the microscope and the hardware (laptop) used to run the application.
- Finally, we can also implement this model for determination of similar diseases like typhoid etc.


## ACKNOWLEDGEMENT

We would like to thank our course instructor Professor Shan Lin for his valuable guidance during the course and presenting us an opportunity to work on such a research project. We would also like to thank our fellow-mates, their valuables and insights helped us a lot during our project.


## REFERENCES


1. https://www.cdc.gov/malaria/diagnosis_treatment/diagnosis.html
2. https://www.healthline.com/health/malaria#symptoms
3. https://www.who.int/news-room/fact-sheets/detail/malaria
4. Das DK, Maiti AK, Chakraborty C (2015) Automated system for characterization and classification of malaria-infected stages using light microscopic images of thin blood smears. J Microsc 257(3):238–252
5. Poostchi M, Silamut K, Maude RJ, Jaeger S, Thoma G (2018) Image analysis and machine learning for detecting malaria. Trans Res: J Lab Clin Med 194:36–55
6. May Z, Mohd Aziz SSA, Salamat R (2013) Automated quantification and classification of malaria parasites in thin blood smears. IEEE International Conference on Signal and Image Processing Applications, Melaka, pp 369–373
7. World Health Organization, "Disease burden of malaria," http://www.who.int/mediacentre/factsheets/fs094/en/.

**8.** D. K. Das, et al. , “Machine learning approach for automated screeningof malaria parasite using light microscopic images,” Journal of Micron,vol. 45, pp. 97–106, Feb 2013.

**9.** F. B. Tek, A. G. Dempster and I. Kale, “Parasite detection and identification for automated thin blood film malaria diagnosis,” Journal of Computer Vision and Image Understanding, vol. 114, no. 1, pp. 21–32, January 2010.

**10.** N. E. Ross, et al., “Automated image processing method for the diagnosis and classification of malaria on thin blood smears,” Medical and Biological Engineering and Computing, vol. 44, no. 5, pp. 427–436, May 2005.

**11.** F. B. Tek, “Computerised diagnosis of malaria,” Ph.D thesis, University of Westminster, 2007.

**12.** V. Muralidharan, Y. Dong, and W. D. Pan, “A comparison of feature selection methods for machine learning based automatic malarial cell recognition in wholeslide images,” in 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), Feb 2016, pp. 216–219.

**13.** J. A. Quinn, R. Nakasi, P. K. Mugagga, P. Byanyima, W. Lubega, and A. Andama, “Deep convolutional neural networks for microscopybased point of care diagnostics,” arXiv preprint arXiv:1608.02989,2016.

**14.** Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998

**15.** Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venu-gopalan S, Widner K, Madams T, Cuadros J et al (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316(22):2402–2410

**16.** Kathirvel CTR (2016) Classifying diabetic retinopathy using deep learning architecture. Int J Eng Res Tech 5(6)

**17.** Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y (2016) Convolutional neural networks for diabetic retinopathy. Procedia Comput Sci 90:200–205

**18.** Haloi M (2015) Improved microaneurysm detection using deep neural networks. In: arXiv preprint arXiv:1505.04424

**19.** Alban M, Gilligan T (2016) Automated detection of diabetic retinopathy using fluorescein angiography photographs. In: Report of standford education

**20.** Lim G, Lee ML, Hsu W, Wong TY (2014) Transformed representations for convolutional neural networks in diabetic retinopathy screening. Modern Artif Intell Health Anal 55:21–25

**21.** San GLY, Lee ML, Hsu W (2012) Constrained-MSER detection of retinal pathology. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE, pp 2059–2062

**22.** Razzak MI, Alhaqbani B (2015) Automatic detection of malarial parasite using microscopic blood images. J Med Imaging Health Inform 5(3):591–598

**23.** F. B. Tek, A. G. Dempster, and I. Kale, “Malaria parasite detection in peripheral blood images,” in Proc. British Machine Vision Conference, Edinburgh, September 2006.

**24.** N.E. Ross, C.J. Pritchard, D.M. Rubin, and A.G. Dus, “Automated image processing method for the diagnosis and classiſcation of malaria on thin blood smears,” Medical and Biological Engineering and Computing, vol. 44, no. 5, pp. 427 – 436, May 2006.

**25.** S. Halim, T. Bretschneider, Y. Li, P. Preiser, and C. Kuss, “Estimating malaria parasitaemia from blood smear images,” in Proc. ICCARV. IEEE, 2006, pp. 648 – 653.

**26.** “CDC DPDx - malaria image library,” http://www.dpd.cdc.gov/ DPDx/HTML/ImageLibrary/Malaria il.htm.

**27.** A.H. Munsell, A Color Notation, Munsell Color Corporation, Baltimore, MD, 1946.

**28.** N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Syst., Man, Cybern., vol. 9, no.1, pp. 62 – 66, 1979.