Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020)
IEEE Xplore Part Number: CFP20J32-ART; ISBN: 978-1-7281-5518-0

Malaria Detection using Deep Learning

Gautham Shekar
U.G Scholar
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
shekargautham1@gmail.com

S. Revathy
Associate Professor
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
revathy.it@sathyabama.ac.in

Ediga Karthick Goud
U.G Scholar
Department of Information Technology
Sathyabma Institute of Science and Technology
Chennai, India
karthikg.palvai@gmail.com

***Abstract*—Malaria is the deadliest disease in the earth and big hectic work for the health department. The traditional way of diagnosing malaria is by schematic examining blood smears of human beings for parasite-infected red blood cells under the microscope by lab or qualified technicians. This process is inefficient and the diagnosis depends on the experience and well knowledgeable person needed for the examination. Deep Learning algorithms have been applied to malaria blood smears for diagnosis before. However, practical performance has not been sufficient so far. This paper proposes a new and highly robust machine learning model based on a convolutional neural network (CNN) which automatically classifies and predicts infected cells in thin blood smears on standard microscope slides. A ten-fold cross-validation layer of the convolutional neural network on 27,558 single-cell images is used to understand the parameter of the cell. Three types of CNN models are compared based on their accuracy and select the precise accurate - Basic CNN, VGG-19 Frozen CNN, and VGG-19 Fine Tuned CNN. Then by comparing the accuracy of the three models, the model with a higher rate of accuracy is acquired.**

***Keywords*—SIANN, VGG, Softmax, Mathew correlation, F_beta, F1 Score, Specificity**

I. INTRODUCTION

The origin of malaria starts from the continent of Africa. The malaria was originated from the virus *plasmodium falcipuram* virus which is the cause for this disease. The disease has traveled through all around the world by the pathogen of mosquitoes. The virus can survive in hot and mild weather, but it cannot survive in very cold weather. The disease was 40 million years old from a very old age period. Malaria can infect all the animals to humans from infant to adult. Starting from fever to coma and death. The disease directly attacks the blood cell of the human body and breaks the white blood cell and stops the functionality of the organs of the human being. Malaria can be detected only by taking the blood samples from the human being and viewed in the microscope.

Malaria infection begins when an infected Anopheles mosquito bites a person, infecting Plasmodium parasites in the form of sporozoites. The sporozoites pass quickly into the liver of a human being and the sporozoites multiply asexually in the liver cells for 7 to 10 days. Then the parasites will form merozoites and will move through the bloodstream and settle at the lung capillaries. The merozoites are then moved through the red blood cells and multiply more, then the cell will burst. This will go through a process of a cycle of the carrier of a mosquito to another healthy person.

If a person is suffering from malaria, the person will get to know from the symptoms that his human body will emit as a warning symbol. The human body will start triggering the white blood cells to provide immune from the malarial cells. It causes high fever, headache, nausea, vomiting, abdominal pain or even coma.

Even though there were many machine learning models to predict malaria. In the proposed work, a deep learning model is used to predict malaria with high accuracy.

II. LITERATURE SURVEY

Peter et al [10] proposed a model for the new genotypic signature for the detection of the malarial cell. So, using that concept, the bloodstain concept is selected [10] for this research work.

Raghuveer et al [5] said that variability and artifacts are important for taking the microscopic images on malarial cells. The model shows that they have taken Leishman's blood smears for this project. So, understanding the concept of the Leishman blood smears and undergoing our project with the same concept.

Ratnaparkhe et al [4] showed the concept of image processing using OpenCV and the contour detection concept which is sued in the proposed work to use contour detection on the blood cell to find the attributes of the blood cell. So, once the attributes are detected, the number of dots will be counted to conclude that the given cell is a malarial cell or not.

Zhaoui et al [7] composed the introduction of the deep learning concept called the convolutional neural network and using a convolutional network finding the accuracy of the infected or not infected blood cell. So, the concept of building the scratch convolutional neural network is used in our proposed work and other convolutional neural networks.

Weihong et al [1] proposed and introduced the advanced concept of a convolutional neural network called the VGG which is Visual Geometry Graphics. The VGG-16 is used in their model. The concept of the VGG-16 model is considered and developed the proposed research work in the VGG-19 model.

Zhuocheng et al [8] portrayed the automatic classification of blood cells using the deep convolutional neural network. The concept uses the database of the malarial cell and used it with the LeNet and AlexNet and GoogleNet. So, understanding the concept of the three convolutional neural networks and using it on our project for building the three convolutional neural networks.

Ross et al [3] introduced the backpropagation feedforward neural network concept. The learning rate of this project was higher than the basic convolutional neural network. So, understanding the concept of the neural network used in that model.

Gavet et al [2] developed a model featuring the time series classification algorithm and recurrence plot. Both these concepts were used to build a unified network using a convolutional neural network. By using the method understanding the working of the Time Series Classification algorithm.

Dallet et al [6] developed a real-time identification of malarial cells in the human body using a mobile phone. By understanding the concept of building a tracking watch, which will monitor our pulse rate and calorie rate.

Gopalakrishna et al [9] introduced the concept of forming the artificial microscopic slide of *plasmodium falcipuram* cell, which causes malaria. Using the concept of the cell structure and cell attributes are studied to understand the morphology of the malarial cell.

Vadavalli et al [11] focused on the concept of deep learning and the use of deep learning techniques to define the truth of a given corpus. Subashini et al [12] proposed the image processing on diabetics.

Madhukeerthana et al [13] introduced medical and pharmaceutical image processing. Revathy et al [14] evaluated a rough fuzzy clustering structure using Decision Theory. Revathy et al [15] derived a suitable machine learning model to predict kidney disease.

III. PROPOSED WORK

The problem starts with the decision taking whether the particular cell is infected or healthy. Start training the machine by giving all the attributes of the images. So, by collecting as much as images through the internet, where totally 27,558 images were collected.

Once all the images are acquired, they started the process of training, validation, and testing. Provided 17,361 to the training set, 1,929 to the validation set, and 8,268 to the testing set. *Fig. 1* will give a detailed explanation of how the project works in a block diagram and the step by step procedure is given in the block diagram. Then imported the OpenCV library to the process.

By using OpenCV, the property will use contour detect on the particular cell. Contour can be described simply as a curve joining all the continuous points having the same color or same intensity. A contour is a useful tool for shape analysis and object detection. When seeing the images, it can be concluded that it has some black dark spots inside the cell. So, these contours will draw the curve near the dark spot forming a circle around.

When the process is finished with the contour detection, then it has been moved on to the most important process called threading. Threading is a separate flow of execution. That means the program will have two things happening at once. Threading can also be called as multiprocessing. So in this project, the threading process happens with an attribute named Thread Pool Executor. Thread Pool Executor creates a context manager, telling it how many worker threads it wants in the pool. It uses .map() to step through an iterable of things. The library will produce the minimum dimension, average dimension, median dimension, and the maximum dimension in the format of an array.

Then loading and resizing of images will take place through Thread Pool Executor for each training set, validation set, and testing set. Once running the images into the machine, the XYZ points will be acquired. Images will move to the setup configuration settings, scaling of images, and label encoding. In this stage, encoding and scaling will take place, where the images are converted into binary code of 1's or 0's and the new term will be used called epochs. Epoch is a point where the time starts.

Epoch = (number of iterations * batch size) / total number of images in training

A. Basic Convolutional Neural Network

The Basic Convolutional Neural Network is done from scratch. So, by using the Tensor Flow, Python library releases the Keras. Keras is the open-source neural network library in Python. By doing so, the Keras has an attribute named Conv2D. It is also Conv1D but in this project, Conv2D is being used because the image being used here is a two-dimensional image. Then the max-pooling part comes there, to maximize the cluster of neurons at the prior layers. Max pooling is a downsampling strategy on CNN.

Normally max pooling is used to converge any type of matrix to the smallest possible value. For Example, by taking the 4X4 matrix in the matrix, there are four corner values. By using the max-pooling effect, the determinants of the four values are detected and the resultant will be a 2X2 matrix. In the Keras layers, there is another layer called Flatten which is used to prepare to become a vector for the fully connected layers by using the flatten command with respective to Keras. So once finished with the attributes of the Keras, then the activation of sigmoid will occur by calculating the accuracy. Once the attributes of the Keras are acquired, the model is built using Keras.model().

Then printing the model summary takes place to identify the given parameters full-fledged with both training and testing. There are a total of 15,102,529 trainable parameters that are acquired and zero Non-trainable parameters. To finish it with the model summary and the training of the model will take place by giving the details of the epochs, batch size, callbacks, verbose, and validation data. Then, fitting the model will start the process of 25 epoch processes. By doing so, the loss percentage will start decreasing slower and slower and comes to the halt after the 25th epoch. At 25th epoch, the readings are, loss: 0.0054, accuracy: 0.9985, validation accuracy: 0.9430 and validation loss: 0.4244.

By seeing these values, it is not convincing because the loss value is way higher of what the expectation was. To get lesser loss value, going to the next version of CNN may produce accurate value. In the graph depicting the Accuracy vs Epoch and Loss vs Epoch graph states that there is a huge gap between the dark line with a grey line that shows the validation percentage of both accuracy and loss graph. Whereas the loss graph as the most deviation between the dark line and the grey line this will give us the graphical representation of the CNN process of the project. So, once the model is trained and acquired the accuracy, it is saved in the extension of basic.h5. The h5 extension is a hierarchical data format used to store scientific data.

B. Frozen Convolutional Neural Network

The order of presentation is same for the both basic and frozen CNN but the given attributes will change as the frozen CNN takes place. In this CNN same, but using the Keras with VGG. VGG comes in two types VGG16 and VGG19. The main difference between the VGG16 and VGG19 is that VGG16 uses 16 layers and VGG19 uses 19 layers in the deep neural network. In this project, VGG19 is used to make it effective run on the images and model for training.

VGG is built under 16 layers the main layers are convolutions layer and max-pooling layer. They are fully connected at the end of the model. The model size is 528MB. The first layer is the convolution layer with 64 filters goes where the image goes through the first layer. Then the second layer is the convolution + max-pooling layer with 64 filters. These layers have a 3X3 matrix this example already given above. Next is the third layer that is convolution layer with 128 filters when by doing this the pool is divide by 2 when it goes from one filter to another filter. Then the next layer is the convolutional layer + max-pooling layer with 128 filters. From this layer, the images will move from images to the pixel unit through the filtering method and produce the matrix. Then comes the 256 filter, 512 filter, at last, the fully connected node layer where it has 4096 nodes where the images will be encoded and then at the output side the softmax activation will be activated with the 1000 nodes. The 224 x 224 x 3 image will further transform into 1 x 1000 when the image comes out from the softmax activation.

So once the model is fitted and the model will be using the history value so from that the model summary can be retrieved. From the model summary, the total parameter is 22,647,367 in that trainable parameter is 2,622,977 and the non-trainable parameter is 20,024,384. From here, it has been concluded that these non-trainable parameters are somehow escaped from the Basic CNN resulting in the zero non-trainable parameters.

The model fitting process is over the training of the model that will be occurred by using the 25 epoch. It is the same as the basic CNN. 25 Epoch will start with 17361 samples then the loss and accuracy values will be generated. Once the process is generated and moves to the 25th Epoch the value will be also generated. Loss: 0.1017, accuracy: 0.9956, validation accuracy: 0.9430 and validation loss: 0.1751.

Once the values are acquired by concluding that the value which was generated is not enough as the loss is less but not accurate. The Epoch vs Loss graph shows the loss percentage which is very low when compared with the basic CNN but not very accurate so it is good to go with the next high certain and pre-trained convolutional neural network.

C. Fine-Tuned Convolutional Neural Network

This is the last stage of the convolutional neural network. Fine-tuning means taking weights of a trained neural network and use it as initialization for a new model being trained on data from the same domain. It is often used to speed up the training and overcome small dataset sizes.

The fine-tuned convolutional neural network has some techniques to accomplish the tasks. They remove the final layer of the past trained network and change it with our new softmax layer that is used in our problem. They use a low learning rate to train the network. The finely tuned practice will make an initial learning rate 10 times smaller than the one used for the basic convolutional neural network. The next practice of the fine-tuned convolutional neural network is locking up the weights of the first layers of the past trained network. This is because the first layers capture unique features like curves and edges that are also relevant to our new problem.

In this convolutional neural network, the imagenet is used as a weight and VGG 19 is used with the Keras. So the layers are getting ready to freeze by training the VGG layers. Then by using the for loop the layers are extracted and checked with the processed weights of imagenet. Then the output of Vgg is analyzed. In Keras, the flattening process will be executed. The flattening is converting the data into a 1-dimensional array for inputting it to the next layer. The output of the convolutional layers will create a single long feature vector and it is connected to the final classification model, which will become a fully connected layer. Then it moves to the next layer called the dense layer. A dense layer is a classical fully connected neural network layer where each input node is connected to each output node. Normally the dense layer will act as a matrix-vector multiplication. The next layer in the Keras is the dropout layer.

The dropout layer is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

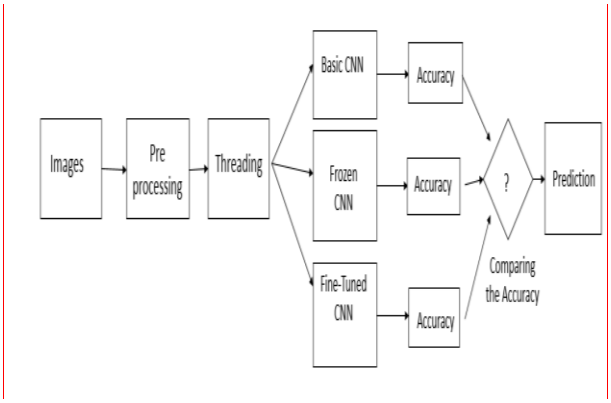

Fig 1: Block Diagram of Proposed Work

Then the model is ready to be fitted with the activation of the sigmoid and then producing metrics with the model. Once the model is trained, the total layers are 28 and total trainable layers are 16. So the numbers are achieved by running the network. The next step is to train the model. The callback response is used during training because they provide specific functionality with a set of methods called testing and prediction. After 25th Epoch the loss: 0.0792, accuracy: 0.9989 validation loss is 0.1127 and validation accuracy is 0.9641. So once the model is fully trained and tested, it is then represented by a graph and check the accuracy vs loss in the epoch. So once the model is trained and acquired the accuracy, it is saved in the extension of frozen.h5. The h5 extension is a hierarchical data format used to store scientific data.

D. Proposed Algorithm

Step 1: Collect the preprocessed images and merge them under one file for easy transportation
Step 2: Split the images according to train and test using sklearn
Step 3: Use OpenCV at the images and understand the parameters of the images and do the process of contour detection
Step 4: Process the image using thread pool executor for not to face time straining.
Step 5: Create the Basic CNN model from scratch and fit the model
Step 6: Now insert the images into the model and run the model by using the tensor flow and Keras package.
Step 7: Use Epoch = (number of iterations * batch size) / total number of images in training
Step 8: Check the accuracy if it is not sufficient then move to the next CNN model
Step 9: Create the Frozen CNN model and fit the model and repeat step 6-7
Step 10: Check the accuracy if it is not sufficient then move to the next CNN model
Step 11: Create the Fine-Tuned CNN model and fit the model and repeat step 6-7
Step 12: If the accuracy is sufficient to stop here and get the accuracy rate.

IV. RESULTS AND DISCUSSIONS

The main aim of this proposed work is to develop an efficient deep learning model to predict Malaria disease. In our proposed work, the dataset *Fig. 2* used is the collection of 25,000 images both infected and uninfected blood cells. The images are retrieved from the websites, kaggle.com and National Medical Science Organization. The dataset was already preprocessed and the images are labeled with healthy or unhealthy.

Fig 2: Sample of blood cell images of both healthy and infected

V. PERFORMANCE ANALYSIS

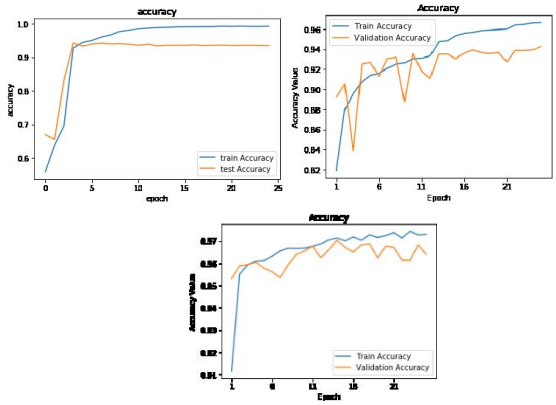

Fig 3: The Accuracy vs Epoch graph starts from the top left is the Basic CNN model, frozen CNN Model, and fine-tuned CNN model

The *Fig. 3* shows the accuracy Vs epoch graph of the three CNN model where it portrays that the fine-tuned CNN has the high accuracy than the other two CNN model because the train and validation accuracy line meets with each other which gives an accurate result from the model.

As discussed that the three convolutional neural networks. The fine-tuned convolutional network shows the highest accuracy among the three convolutional neural networks. The only disadvantage of the fine-tuned convolutional neural network is that it is a very cost-effective process and it takes more time.

For the prediction, the confusion matrix is formed for each convolutional neural network for better understanding. The confusion matrix is a performance measurement for machine learning classification. A confusion matrix is a two-dimensional matrix with four attributes true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The confusion matrix is used to calculate the precision values, accuracy, recall, and F-measure.

Table 1: The confusion matrix of the basic CNN

Actual/Predicted	Healthy	Malaria
Healthy	3884	191
Malaria	225	3968

Table 1 shows that the basic convolutional neural network's confusion matrix states that there are 3884 cells are healthy and 3968 cells are unhealthy or infected but the accuracy is 94% which is not enough to predict the values.

Table 2: The confusion matrix of the Frozen CNN

Actual/Predicted	Healthy	Malaria
Healthy	3871	204
Malaria	212	3881

Table 2 shows that the frozen convolutional neural network's confusion matrix states that there are 3871 cells are healthy and 3881 cells are unhealthy or infected but the accuracy is 92% which is less than Basic CNN which cannot be used to predict the values.

Table 3: The confusion matrix of the Fine-Tuned CNN

Actual/Predicted	Healthy	Malaria
Healthy	4004	71
Malaria	260	3933

Table 3 shows that the fine-tuned convolutional network's confusion matrix states that there are 4004 cells are healthy and 3933 cells unhealthy or infected have a higher accuracy of 96% which will be used to predict accurate values.

Table 4: The calculation values of the three convolutional neural network

CNN	**Acc**	**Class.**	**Sen**	**Pre**	**F1**	**Fb**	**Spe**	**Fal**	**Mat**
Basic CNN	0.94	0.05	0.94	0.95	0.95	0.95	0.95	0.04	0.89
Frozen CNN	0.92	0.07	0.90	0.93	0.91	0.91	0.94	0.05	0.85
Fine-Tuned CNN	0.96	0.03	0.93	0.98	0.96	0.96	0.98	0.01	0.92

Table 4 shows that the calculation of different parameters for three different convolutional neural networks, where Acc means Accuracy, Class means Classification Error, Sen means Sensitivity, Pre means Precision, F1 means F1 Score, Fb means F_beta, Spe means Specificity, Fal means False positive rate and Mat means Mathew correlation.

VI. CONCLUSION

Malaria detected from the traditional method that is bringing the samples and analyzing cell growth requires more time. So in the proposed work, a deep learning model has been constructed to predict Malaria with a high accuracy rate and low time duration. Three CNN models were constructed and identified as the highest accuracy model. The Fine-Tuned CNN provided a high accuracy rate compared to the other CNN models. The future work will be, working on disease detection like pneumonia, breast cancer using CNN, and planning for the detection of COVID-19 smears in the lungs of the human body.

REFERENCES

[1] Shuying Liu and Weihong Deng., "Very deep convolutional neural network based image classification using small training sample size" *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR).*

[2] NimaHatami., Yann Gavet and Johan Debayale., "Classification of time-series images using deep convolutional neural network" *Tenth International Conference on Machine Vision (ICMV 2017)* Vol.10696.

[3] Nicholas E. Ross., Charles J. Pitchard., David M. Rubin and Adriano G. Duse "Automated image processing method for the diagnosis and classification of malaria on thin blood smears" *IEEE* Vol.44.

[4] K.M. Khatri., V.R. Ratnaparkhe., S.S. Agarwal and A.S. Bhalchandra "Image processing approach for malarial parasite identification" *International Journal of computer Application (IJCA).*

[5] Vishnu V. Makkapati and Raghuveer M. Rao., "Segmentation of malarial parasites in peripheral blood smear images" *IEEE International Conference on Acoustics, Speech and Signal Processing*.

[6] CorentinDallet., Saumya Kareem and Izzet Kale., "Real time blood image processing application for malaria diagnosis using mobile phones" *IEEE International Symposium on Circuits and Systems (ISCAS).*

[7] Zhaoui Liang., Andrew Powell., IlkerErsoy.,Mahdieh Poostchi., Kamolraut Silmaut and Kannapan Palani.,"CNN-based image analysis for malaria diagnosis" *IEEE International Conference on Bioinformatics and Biomedicine (BIBM).*

[8] Yuhang Dong., Zhuocheng Jiang., Hongda Shen., W. David Pan., Lance A. Williams., Vishnu V. B. Reddy., "Evaluations of deep convolutional neural network for automatic identification of malaria infected cells" *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI).*

[9] Gopalakrishna Pillai Gopakumar., Murali Swetha and Gorthi Sai Siva "Convolutional neural network-based malaria diagnosis from stack of blood smear images acquired using custom-built slide scanner" *Journal of Biophontics,* Vol. 11 No.3.

[10] Peter Gascoyne., Jutamaad Satayavivad and Mathuros Ruchirawat "Microfludic approaches to malaria detection" *ActaTropica,* Vol. 89 No.3.

[11] Vadavalli, A., Subhashini, R., "Deep Learning based truth discovery algorithm for research the genuineness of given text corpus*", International Journal of Recent Technology and Engineering*, 2019.

[12] Subhashini, R., Nithin, T.N.R., Koushik, U.M.S., "Diabetic Retinopathy Detection using Image Processing (GUI)", *International Journal of Recent Technology and Engineering*, 2019.

[13] Madhu Keerthana, Y., Bevish Jinila., (2016), "A Review on Rough Set Theory in Medical Images", *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, Vol.7, Issue 1, pp.815-822, ISSN: 0975-8585.

[14] S.Revathy., B.Bharathi., P.Jeyanthi., M.Ramesh.,"Chronic Kidney Disease Prediction using Machine Learning Models", *International Journal of Engineering and Advanced Technology (IJEAT)*, ISSN: 2249 – 8958, Vol.9, Issue-1, October 2019.

[15] Subramanion, Revathy., Parvathavarthini, Balasubramanion., "Decision Theoretic Evaluation of Rough Fuzzy Clustering", *Arab Gulf Journal of Scientific Research*, Vol.32, (2014).