Malaria Detection using Deep Learning

Akansha Singh[1], Neetika Porwal [2],Aditya Kumar[3]

[1]Department of Computer Science and Engineering, Galgotias University Greater Noida, India
akanshasingh172000@gmail.com
[2]Department of Computer Science and Engineering, Galgotias University Greater Noida, India
neetikaporwalhdi@gmail.com
[3]Department of Computer Science and Engineering, Galgotias University Greater Noida, India
aditya9599215021@gmail.com

Article History: Received: 5 April 2021; Accepted: 14 May 2021; Published online: 22 June 2021

Abstract— Malaria is a life-threatening disease that is spread by the Plasmodium parasites. It is detected by trained microscopists who analyze microscopic blood smear images. Modern deep learning techniques may be used to do this analysis automatically. The need for the trained personnel can be greatly reduced with the development of an automatic accurate and efficient model. In this article, we propose an entirely automated Convolutional Neural Network (CNN) based model for the diagnosis of malaria from the microscopic blood smear images. Our deep learning-based model can detect malarial parasites from microscopic images with an accuracy of 96.52%. For practical validation of model efficiency, we have deployed the miniaturized model in a server-backed web application. Data gathered from this environment show that the model can be used to perform inference under 1's per sample in online (web application) mode, thus engendering confidence that such models may be deployed for efficient practical inferential systems.

Keywords—VGG, Softmax, CNN, Mathew correlation

I. INTRODUCTION

Malaria is one of the leading causes of fatalities in the developing world. Effective malaria diagnosis contributes towards effective disease management, control and surveillance. High-quality diagnosis is essential in all settings as misdiagnosis can result in drug resistance coupled with the economic burden of buying unnecessary drugs and in some cases significant morbidity and mortality. The standard system for diagnosis of Malaria plasmodium is microscopy of blood smear slides. Un-like Rapid Diagnostic Tests (RDTs) that detect specific anti-gens derived from malaria parasites in red blood cells, microscopy supports direct parasite detection. The thick blood smear microscopy method is the most com-mon method for screening malaria parasites. It facilitates the separating of plasmodium parasites from the Red blood cells associated with parasites in a sample of blood. To achieve a negative or positive test, thick blood smear slides are usually used while the thin smears are used for species determination of the malaria infection. Laboratory diagnosis of Malaria mainly uses this method because of its simplicity, relatively low cost and the possibility of determining that magnitude of the parasitaemia in a sample. These traditional microscopy related disadvantages have led to a pressing need for quality microscopy improvements. Automation eliminates human error such as missed parasites in low parasitaemia samples or fatigue that can occur during repeated viewing of stained slides. The emergence of web technologies and computer vision techniques that surpass human ability in detection of disease pathogens have the potential to improve public health. These can be used to bridge the diagnostic gaps in Malaria diagnosis. Web-based technology is used as a platform to enable easy and quick dissemination and interpretation of diagnostic in-formation by users (lab technologists) appropriately. The integration of the web-based system and computer vision detection model will benefit laboratory technicians who wish to achieve decision-supported detection results. This mode of use can help laboratory staff to achieve consistency in diagnosis, and by focusing concentration on parts of the images likely to contain pathogens, may also help to relieve operator fatigue and improve throughput rates. A web-based system was integrated with a computer vision algorithm based on a pre-trained Faster R-CNN model to enhance diagnosis.

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

Raghuveer et al [2] 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 [3] 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.

Weihong et al [4] 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 [5] 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.

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.

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.

B. 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.

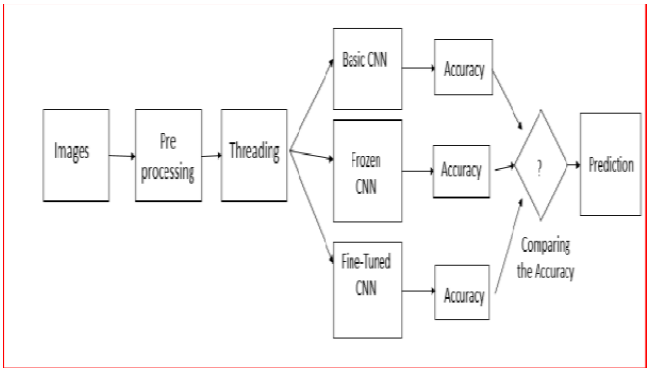

Fig 1: Block Diagram of Proposed Work

C. 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

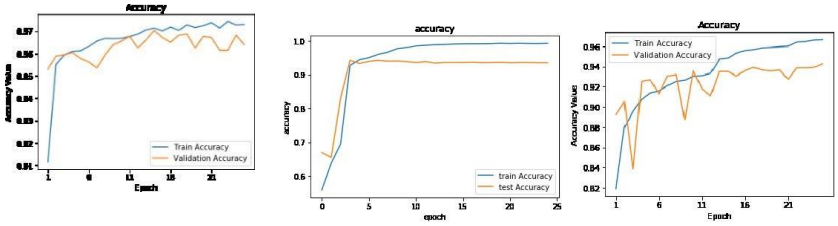

Fig 3: The Accuracy vs Epoch graph starts from the top left is the Basic CNN model, frozen CNN Model, and fine-tuned CNN model

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.

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. Peter Gascoyne., Jutamaad Satayavivad and Mathuros Ruchirawat "Microfludic approaches to malaria detection" ActaTropica, Vol. 89 No.3.
2. 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.
3. K.M. Khatri., V.R. Ratnaparkhe., S.S. Agarwal and A.S. Bhalchandra " Image processing approach for malarial parasite identification"
4. International Journal of computer Application (IJCA).
5. 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).
6. 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 Inform atics (BHI).