CNN Based Deep Learning Approach for Automatic Malaria Parasite Detection

Mousami Turuk, R Sreemathy, Sadhvika Kadiyala, Sakshi Kotecha, Vaishnavi Kulkarni

Abstract— **Malaria is a deadly disease spread by the bite of a female anopheles mosquito infected by the plasmodium parasite. Despite thorough research in the medical field, the pervasiveness of this deadly disease is increasing globally. More than 1.5 billion cases have been averted in the last two decades, with 7.6 million deaths. The traditional microscopic method of detecting whether a person is infected or not is time-consuming. Its accuracy depends on smear quality and individual expertise in counting and classifying parasitized and uninfected cells. Computer-aided diagnostic methods based on image processing and Machine learning (ML) use hand-generated features and need expertise in texture, morphological aspects, and analysis of a region of interest. Convolution Neural Network (CNN) based methods are superior to the traditional ML-based approach. They are highly scalable and give the best results with end-to-end feature extraction and classification. Hence, this research intends to design and develop a reliable framework for automatic malaria parasite detection using Deep learning (DL), which could serve as the best effective aid. This research paper evaluates the performance of pre-trained CNN-based DL models like AlexNet, ResNet50, and VGG19 as feature extractors for analyzing infected and non-infected cells. Statistical results show pre-trained CNN models serve as the best feature extractor tool for this purpose. Features extracted using VGG19 are proven to be more efficient than the ResNet50 and AlexNet to detect the presence of malaria parasites with a training accuracy of 95.28% and a testing accuracy of 93.89%. This transfer learning and CNN model are integrated further with a web application for accurate and automatic malaria parasite detection**.

***Index Terms*—Deep Learning, Transfer Learning, Malaria Parasite Detection**

I. Introduction

Malaria is an extensive worldwide public health issue that causes widespread suffering and misery, spreading by a female anopheles mosquito's bite infected by the plasmodium parasite.

Manuscript received July 19, 2021; revised March 28, 2022.

Mousami Turuk, Assistant Professor, Department of Electronics & Tele Communication Engineering, Pune Institute of Computer Technology, Pune, India, mpturuk@pict.edu

R. Sreemathy, Associate Professor, Department of Electronics & Tele Communication Engineering, Pune Institute of Computer Technology, Pune, India, rsreemathy@pict.edu

Sadhvika Kadiyala, Student, Department of Electronics & Tele Communication Engineering, Pune Institute of Computer Technology, Pune, India, sadhvika7599@gmail.com

Sakshi Kotecha, Student, Department of Electronics & Tele Communication Engineering, Pune Institute of Computer Technology, Pune, India, sakshikotecha12@gmail.com

Vaishnavi Kulkarni, Student, Department of Electronics & Tele Communication Engineering, Pune Institute of Computer Technology, Pune, India, vaishnavi.9.5.2000@gmail.com

These parasites travel through the bloodstream and infect the human body's Red Blood Cells (RBCs). More than 1.5 billion cases have been averted in the last two decades, with 7.6 million deaths. In 2019, 229 million people were infected by malaria, and 409,000 deaths were reported as per the WHO report [1].

Inadequate malaria diagnosis has been one of the significant roadblocks to significant mortality reduction. To identify the presence of the infection-causing parasite, a proper diagnosis of the blood smear is required.

The thin blood smears assist in determining the species of the Plasmodium parasite, namely Plasmodium falciparum (P. falciparum), Plasmodium ovale (P. ovale), Plasmodium vivax (P. vivax), Plasmodium malariae (P. malariae), or Plasmodium knowlesi (P. knowlesi) [2], whereas the thick blood smear is used just to identify its presence. The traditional method of detecting whether a person is infected or not involves manually counting the number of infected cells in the blood smear by an experienced microscopist [3],[4], which has a lot of-limitations, such as processing time and reliability. Technologies like Machine Learning and Deep Learning can be leveraged to overcome these drawbacks. Image analysis tools and machine learning approaches can improve diagnosis and estimate the presence of parasites in the blood in microscopic blood slides. The detection procedure becomes more robust, dependable, and exact when it is automated.

The first attempt at automation was made with digital image processing. Certain image transformations and segmentation techniques served to improve the features of the image so that categorization could be done easily [5]. Later, as Machine Learning progressed, many efforts were taken in the medical field. SVM, Decision Trees, Random Forests, Ada boost, KNN, Naïve Bayes [6], [7] are some of the effective classification methods in detecting malaria. Deep Learning techniques [8] such as CNN and transfer learning improved the performance much further. These models were discovered to generate significantly better findings faster than traditional diagnoses.

II. Related work

The efforts undertaken in identifying malaria parasites in blood smears were studied to overcome the shortcomings of traditional methods. Saiprasath G.B. et al. [7] focuses on the automatic detection of malaria by local action and separating infected erythrocytes from healthy individuals with a low-grade smear. They have used seven different machine learning algorithms: AlgoBoost, Random Forest, Decision Tree, KNN, Linear regression, Naive Bayes, and Extra Trees. They aimed to evaluate the performance based on the presence of the virus at the patch level and not at the overall

image level of each patient. The random forest algorithm provided excellent results in detecting malaria with 96.5% accuracy. Rajaraman S. et al. [9] provide feature extractors using pre-trained CNN-based deep learning models for uninfected and infected blood cell classification for disease identification in their study. The methodology uses six pre-trained models on the dataset to obtain a 95.9% accuracy in detecting infected and non-infected samples. It also used a test method to figure out the ideal model layers. Nakasi et.al. [10] analyzed three pre-trained deep learning architectures faster regional convolutional neural network ,single-shot multi-box detector (SSD) and RetinaNet using a Tensorfow object detection API. The likelihood for mobile phone detector deployment is investigated with the applicable solution for class and location of pathogens with degree of detection confidence. The authors claim faster R-CNN was the best trained model with a mean average precision of over 0.94 and SSD, was the best model for mobile deployment. Masud M. et al. [11] developed a mobile-based application using a customized CNN model that may be used in real-world malaria detection and reduce manual labor. The custom CNN model produced an accuracy of 97.30 % with a high degree of precision and sensitivity. Kalkan S.C. and Sahingoz O.K. [12] used a deep learning approach to classify blood smears as healthy or parasitized. A simple CNN network with 10 layers was built, and the parameters were fine-tuned by using 5-fold cross-validation targeting the identification of plasmodium falciparum. This experiment led to a training accuracy of 97% and test accuracy of 95% with the help of a strong GPU. Dong Y. et al. [13] performed a comparative analysis on the available deep learning neural networks LeNet, GoogLeNet, and AlexNet and compared their results with a non-neural network classifier SVM. The study resulted in an accuracy of 98.13% for GoogLeNet compared to SVM with only 91.66%. Liang Z. et al. [14] applied deep learning to build a 17 layered CNN model with 10-fold cross-validation. Also, a transfer learning model was built by using a pre-trained AlexNet model, which was further linked with an SVM classifier for comparison. The results exhibited comprehensively that the 17 layered CNN model outperformed the transfer learning model in all parameters with an accuracy of 97.37% vs. 91.99%, the sensitivity of 96.99% vs. 89%, specificity of 97.75% vs. 94.98%, the precision of 97.73% vs. 95.12% and F1 score of 97.36% vs. 90.24%. In addition to the many papers that claim to detect malaria parasites and the type of malaria, Das D.K. et al.[15] also provided a solution to see at what stage the parasite is. They focused on developing a machine learning system that discriminates five different categories, three of P. Vivax and two of P. Falciparum. They found 94 factors to be statistically significant in determining against 6 classes. The results show that the Besese method provides high accuracy of 84% for malaria classification by selecting the 19 most important features while SVM provides high accuracy, i.e., 83.5% with 9 most essential features. Fuhad et al. [23] implemented multiple classification models comprising knowledge distillation, data augmentation, and autoencoder. Feature extraction is performed using the CNN model, and classification techniques like Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are applied to detect infected malaria parasites. Model accuracy, inference performance, and optimization are improved using three training procedures: general, distillation, and autoencoder. Zongo, P. et al.[24] proposed a strategy is to create a computational tool that is as efficient as possible. It illustrates how to calculate the minimum percentage of immunocompromised individuals that should be safeguarded to eradicate malaria from the entire population over time.

III. PROPOSED METHODOLOGY

This work aims to design and develop a reliable framework using a transfer learning approach to differentiate between infected and uninfected malaria parasite cells in thick blood smear. The approach should be more accurate and precise than the traditional method, which requires expertise and is time-consuming. In establishing efficient feature extraction and classification of malaria, CNN brings down domain expertise. This method incorporates image analysis filters applied in multi-stage processing layers. Implementing and comparing transfer learning methods can improve diagnosis. The following methodology is proposed to accomplish this aim, as shown in Fig. 1., which depicts the sequential steps for implementing this work.

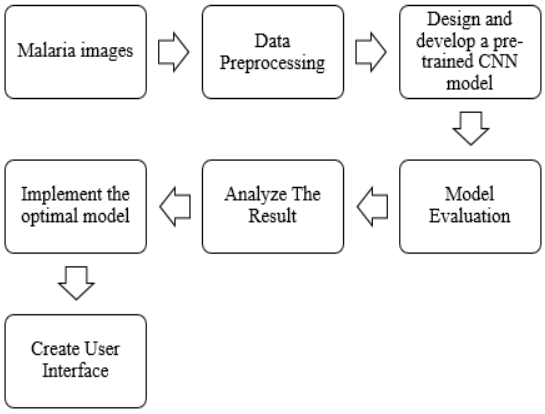


Fig. 1. Block diagram of the proposed research methodology

The first stage is to collect images of thick blood smears to build a dataset for further processing. The second data pre-processing block involves dividing the dataset into training, validation, and testing datasets, followed by resizing the images. In the third stage, the transfer learning technique is applied using pre-trained CNN models such as AlexNet, ResNet50, and VGG19, from which the optimum model is selected for analyzing the result. A User Interface is also developed, which takes a thick blood smear image as the input from the user and predicts the output as infected or not infected.

A. Implementation

The dataset used in this experimentation is collected from publically available data for research. The link for the dataset is https://lhncbc.nlm.nih.gov/LHC-downloads/dataset.html [16]. This dataset includes Giemsa-stained thick blood smear images from 150 malaria infected patients, with 12 images per patient on average. It comprises 27,558 cell images with 13,779 instances of infected and 13,779 instances of uninfected cells. Images are captured with 100x magnification in RGB color space. A thick blood smear is used to detect the presence of malaria parasites in a drop of blood. It offers more efficient detection of parasites than a thin blood smear, with about 11 times higher sensitivity

[17,18]. From Figure 2, it is evident that in the images of thick blood smears containing a spot are categorized as infected (Fig. 2 (a), (c), (d)) and the others are uninfected (Fig. 2 (b), (e)).

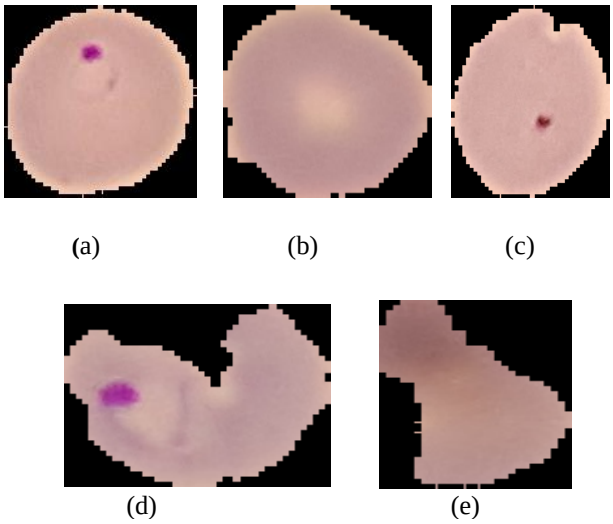


Fig. 2. Sample images taken from the dataset

Experimentation is carried out on the set of 3275 images from the dataset[16]. Training data is required to build deep learning models, and the testing data is the unseen data is needed to check the model's performance. The dataset is divided into train, validation, and test, respectively. The training dataset and validation dataset help during training the model, and the execution of the trained model are examined on the test dataset. The train, validation, and test dataset are divided in the ratio of 70:15:15, respectively, as mentioned in Table I.

TABLE I
DATASET DIVISION

	Healthy	Infected
Total - 3275	1639	1636
Training - 2292	1161	1131
Validation - 492	251	241
Test - 491	264	227

B. Transfer learning

Deep Learning is a popular subclass of Machine Learning where multiple layered architectures extract the higher-level features. DL model that has been pre-trained can deal with the diversity of datasets and respond to instances when data is limited, such as in the medical field. The transfer learning approach has ascended as it allows the reuse of pre-trained models for feature extraction. It is a method of reusing models trained with large amounts of data to perform a training process using small amounts of data and still achieve a high level of accuracy. Deep learning enables high performance with a neural network with enormous input data. Practically, it is not easy to collect an enormous amount of data quickly enough when tackling a completely new task. However, obtaining an acceptable performance using only a small amount of data for training is also difficult. In these situations, transfer learning can be used where neural networks are trained with a large amount of data. Transfer learning is flexible, uses pre-trained models directly as feature extractors, and is integrated into new models. Deep learning can solve complex problems with good results but at the cost of training time and more data samples. Hence, pre-trained networks are used, which form the basis of transfer learning in deep learning. A learned feature map on a large data set can be used for other tasks without developing the model from scratch. Initial layers in deep networks extract generic features, whereas higher layers focus on task-specific features. In some cases, everything can be reused rather than the final layer or the first half of the neural network. The second half of the neural network or sometimes just the final layer is configured to solve the actual task. The reuse of a convolutional neural network to classify 1000 different images for another image-related task will have 1000 output neurons. To apply this to a binary image classification problem, the final layer has 2 output neurons. The first half can be kept the same for semantic segmentation, and the latter can be modified per requirement. The similar weights of the first half layers are used by freezing those layers. The advantage of this is an extremely fast training process. Some pre-trained networks for image classification are AlexNet, Visual Geometry Group (VGG-16), VGG-19, GoogleNet, Residual Network (ResNet50), etc. The final layers in the classification can be replaced to develop a new model successfully, or selective retraining of the previous layers can be done in a pre-trained network.

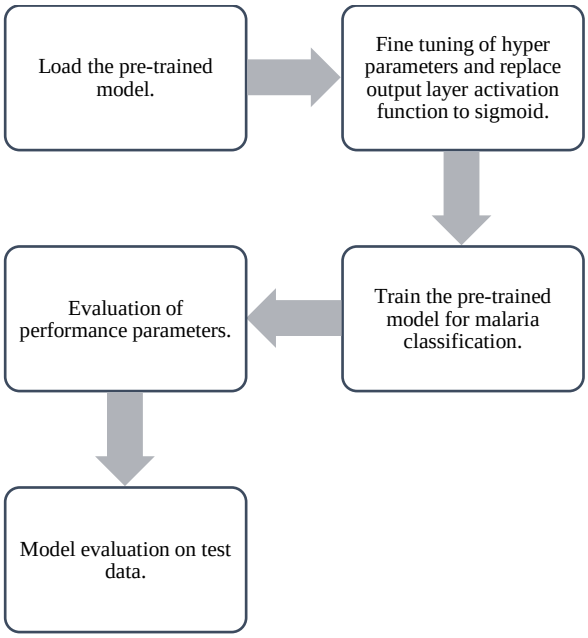


Fig. 3. Reuse of the pre-trained network

This study was worked upon 3 pre-trained models VGG19, AlexNet, and ResNet50. All the neural networks were provided the same dataset of 3275 images. The final layers of each model were changed to a dense layer having a sigmoid activation function instead of softmax for binary classification. A threshold found by the ROC curve determines the value between 0 and 1, and the final classification is based on this threshold value. Fig. 3 shows the reuse of the pre-trained network. It involves importing the

Keras application library containing the VGG19 and ResNet50 functions. AlexNet layers have been written explicitly as there is no readily available function. The top layer weights are frozen, and the final layers are replaced, followed by training, analysis, and result deployment. Using the SVM classifier, a model is built to test how a non-neural network responds to the same dataset. For the SVM classifier, the dataset is divided in the ratio of 7:3, where 70 percent is the training dataset, and 30 percent is the test dataset.

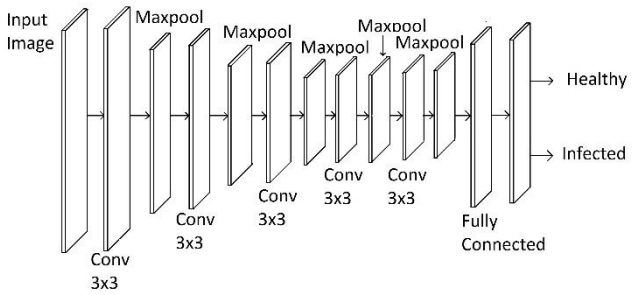


Fig. 4. Implementation of VGG19 Architecture

The Visual Geometry Group developed the VGG19 architecture [19]. As depicted in Fig. 4, this architecture contains 16 layers of convolution network and 3 fully connected layers with 138 million trainable parameters. A fixed size of (125x125) image is given as input to the network; this means the shape of the matrix is (125,125,3). The kernel size is (3x3) with a stride of 1 pixel. This helps to cover the whole image. Over a (2x2) pixel window with a stride size of 2, max pooling is performed, followed by a Rectified linear unit (ReLU) to make the model classify better. There is a final layer of softmax function. The softmax function turns a vector of k real values to the output to k real values that sum to 1. The values obtained from the softmax equation are then used to classify the image in the appropriate category according to the output weights of that category. As malaria parasite detection is a binary classification, the softmax function layer has been replaced with the sigmoid function layer. The sigmoid function maps the output value in the range of 0 to 1, and the classification is done based on the threshold value calculated by the ROC curve.

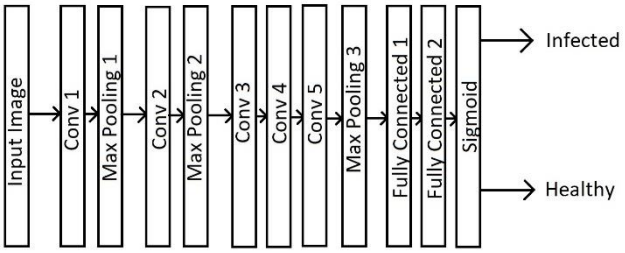


Fig. 5. Implementation of AlexNet architecture

The AlexNet architecture, as shown in Fig. 5, comprises eight layers, five convolutional layers, and three fully connected layers with 61 million trainable parameters. The AlexNet authors [20] also used overlap in the case of a pooling layer and found a reduction in error. Followed by a convolution layer with 96 kernels of size 11x11 and stride 4, there is an overlapping max-pooling layer with a mask size of 3x3 and stride 2. The stride size is less than the kernel size; hence, overlapping occurs. Max-pooling is used to lower the height and width of the image array by maintaining its depth. Similarly, there are 4 more convolutions and 2 more max-pooling layers. Then there is the fully connected layer. The AlexNet layer consists of the Softmax function at the last layer. As this is a binary classification task, the Softmax function is changed to sigmoid. After every fully connected and convolutional layer, the ReLU function is applied to speed up the model. Instead of regularization, AlexNet uses the dropout function to deal with over-fitting.

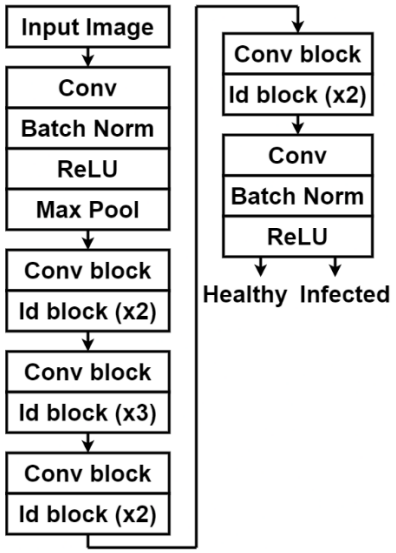


Fig. 6. Implementation of ResNet50 architecture

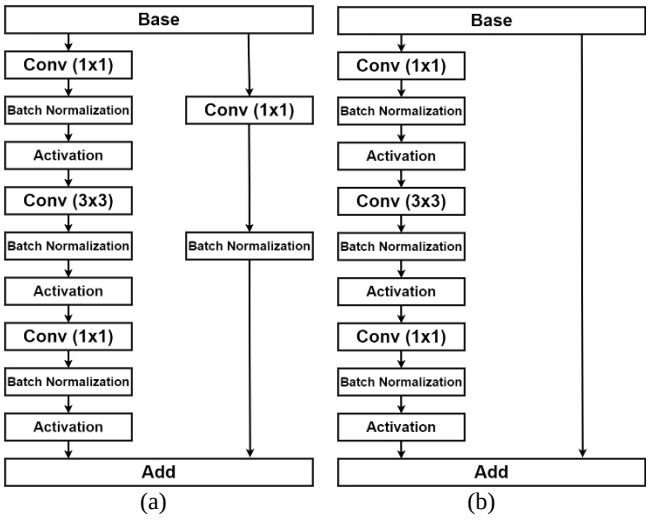


Fig. 7. (a) Conv Block and (b) Id block of ResNet50 architecture

ResNet50 architecture shown in Fig. 6 and Fig. 7, also known as Residual Network [21], has 48 convolution layers, 1 max-pooling layer, and 1 average-pooling layer with 23 million trainable parameters. ResNet is the structure that delivers the idea of skip connection. ResNet lessens the trouble of vanishing gradients by introducing the alternate shortcut path for the gradient to flow through. It comprises 5 steps, each with a convolution and identity block. Each convolution block has 3 convolution layers, and each identity block too has 3 convolution layers.

Stage 1: In the first convolution layer, there are 64 different kernels, all with a stride of size 2 and kernel size of (7x7). A max-pooling layer is followed by a stride size of 2. In the next convolution layer, there are 64 kernels of size (1*1), following 64 kernels of size (3x3), and at last a 256 kernel of size(1x1). These layers are repeated thrice, and 9 layers are obtained at this stage.

Stage 2: This has 128 kernels, each with a kernel size of (1x1). After this, there are 128 kernels of kernel size (3x3), and at last 512 kernels of kernel size (1x1). This step is repeated 4 times. So, at the end of this stage, we get 12 layers.

Stage 3: There are 256 kernels of size (1x1) after the second stage. After this, there are 256 kernels with a kernel size of (3x3) and at last 1024 kernels with size (1x1). This step is repeated 6 times, and a total of 18 layers is obtained at this stage.

Stage 4: This has 512 different kernels with size (1x1). After this, there are 512 kernels of size (3x3) and (1x1), 2048 at the end. This step is repeated thrice, and 9 layers are obtained at this stage.

Stage 5: The above layer is Average pooled and ends with a fully connected layer containing 1000 nodes. In the end, the sigmoid activation function is applied.

These five stages result in the 50 layers of the Deep Convolutional network.

C. SVM

Support vector machine (SVM) [22], a supervised learning model, is used for classification. It is used for its excellent generalization ability. A considerable nonlinear classification line can be achieved with good accuracy and time efficiency by varying the tuning parameters like kernel, regularization, and gamma. To transform the data into a suitable form, various kernel trick functions like polynomial, linear, nonlinear, radial basis function, etc., are used to convert lower dimensional input space to higher-dimensional space. A Radial Basis Function (RBF) kernel trains the SVM model. For this, the process of cross-validation selects a regularization parameter 'C' with value 1. This model helps compare models of non-neural networks and models of neural networks using transfer learning. SVM is a simple Machine Learning model; hence, feature extraction is an extra task that must be implemented separately before classifying the image. Unlike the CNNs, which inherently act as feature extractors, firstly, various dominant features of the image must be found and then served to the SVM classifier. For this task, a single feature, i.e., the standard deviation of all the image matrices, is provided to the classifier.

This work is implemented using Python with Tensorflow, Numpy, Pandas Matplotlib, Sci-Kit Learn, and Flask. This experiment was performed on a CPU system with 8 GB RAM and Intel i5 processor configuration.

IV. RESULT

Evaluation of the models built in the training phase is done by making predictions by fitting the model on the data from the test dataset. The model potential is examined by its performance metrics with parameters like accuracy, f1 score, recall, and precision. Analysis of the result is done with the help of graphs and a confusion matrix. Finally, the optimal model is implemented on the test dataset.

Performance of all models was recorded with the confusion matrix primarily. The rows in the confusion matrix correspond to what the machine learning algorithm predicted, and the columns correspond to the actual labels, as shown in Table II.

TABLE II
CONFUSION MATRIX

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

The threshold for classification was calculated separately for each of the models using the Receiver Operating Characteristic (ROC) curve**.** The y-axis of the ROC curve shows the True Positive Rate (1), and the x-axis contains the False Positive Rate (2). The ROC curve summarizes all the confusion matrices that each threshold produced. Hence, the optimal threshold can be located.

$$TPR = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (1)$$

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives} \quad (2)$$

All the models have been trained for 15 epochs with a batch size of 64. In the graphs of loss vs. epoch and accuracy vs. epoch as depicted in Fig. 8, Fig. 9, and Fig.10, it can be observed that after the 11^{th} epoch, both the training and validation curves become constant; hence 15 epochs were chosen for implementation. Performance metric parameters such as Accuracy (3), Precision (4), Sensitivity (5), Specificity (6), and F1 Score (7) are specified for all the models as shown in Table IV. All these performances are calculated as:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ samples} \quad (3)$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (4)$$

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ \ Negatives} \quad (5)$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ positives} \quad (6)$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (7)$$

The accuracy and loss curve is also plotted for each model to understand the rate of improvement in model performance epoch by epoch.

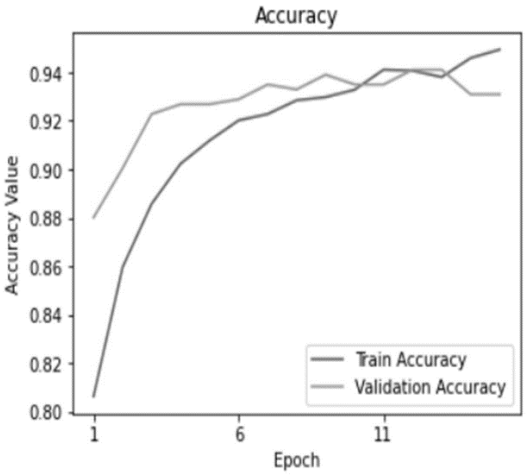
A. Transfer Learning Models and SVM Model.

In transfer learning, three pre-trained models, VGG-19, AlexNet, ResNet50, and an additional non-neural network algorithm SVM, were used for the comparative study. The train and test accuracy for all four models is shown in Table III. Graphs of accuracy vs. epoch and loss vs. epoch for the models are shown in Fig. 8, Fig. 9, and Fig. 10 of VGG-19, ResNet50, and AlexNet, respectively.

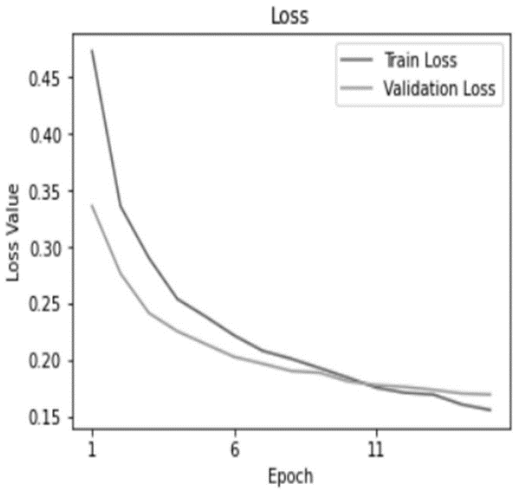
The ROC curve for the VGG-19 model suggests a threshold of 0.51 for the classification. Similarly, a threshold of 0.45 was suggested for AlexNet, and 0.64 for ResNet50.

TABLE III
RESULT

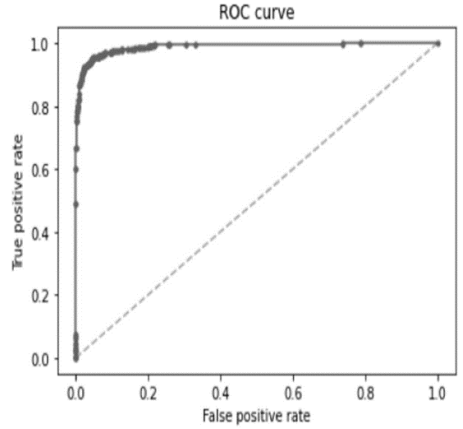
Model	Threshold	Train Accuracy	Test Accuracy
VGG-19	0.51	95.28%	93.89%
AlexNet	0.45	83.94%	82.68%
ResNet50	0.64	82.24%	80.24%
SVM	-	90.57%	80.46%



(a)

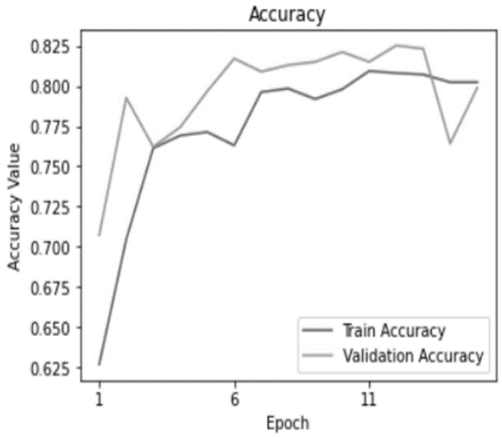


(b)

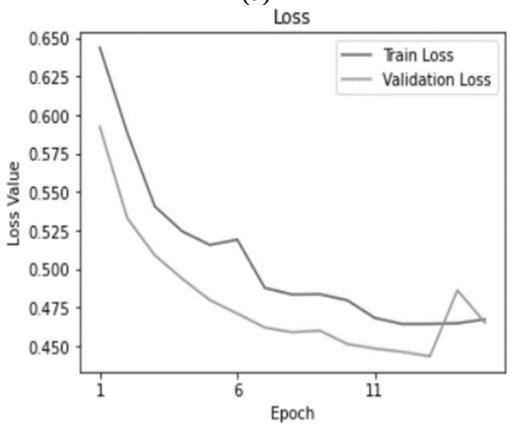


(c)

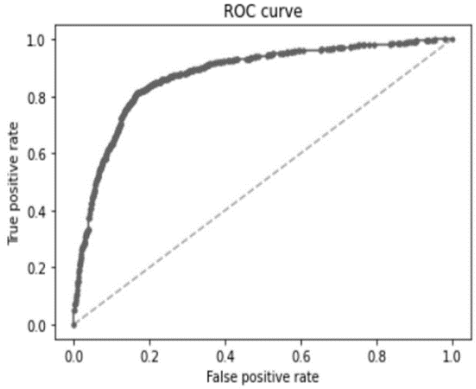
Fig. 8. (a)Accuracy vs. Number of epochs (b) Loss vs. Number of epochs (c) ROC curve for VGG19



(a)

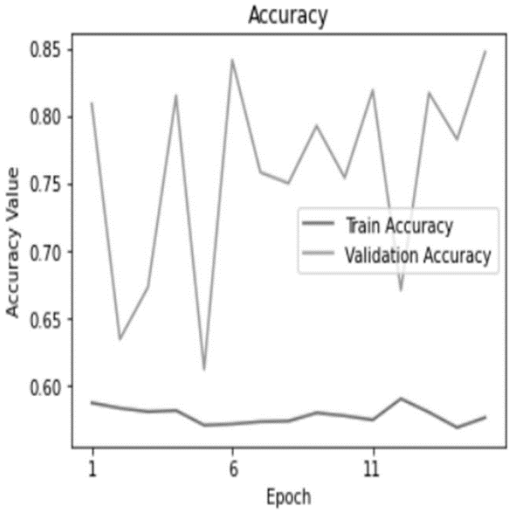


(b)

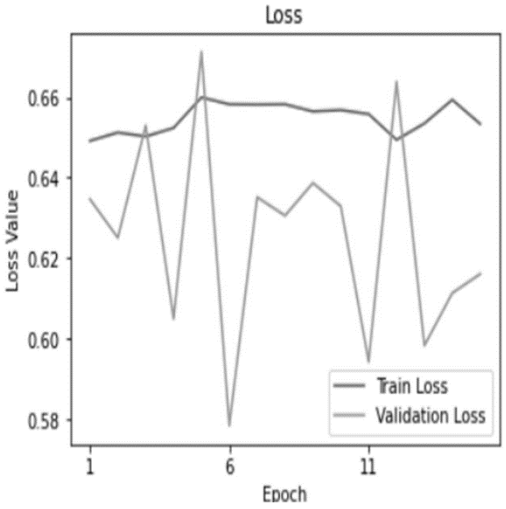


(c)

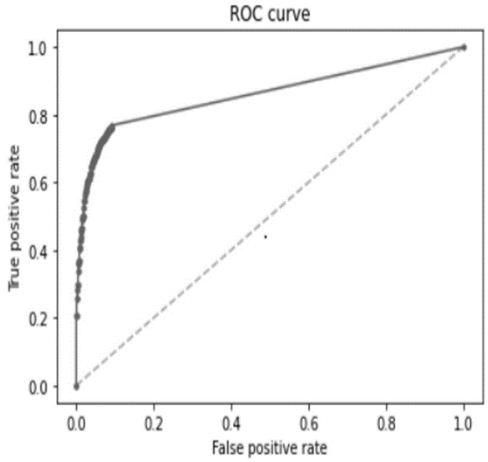
Fig. 9. (a)Accuracy vs. Number of epochs (b) Loss vs. Number of epochs (c) ROC curve for ResNet50



(a)



(b)



(c)

Fig. 10. (a)Accuracy vs. Number of epochs (b) Loss vs. Number of epochs (c) ROC curve for AlexNet

A comparative analysis for all the models is shown in Table IV and V. Sensitivity determines what percent of patients infected by malaria were correctly identified. This parameter is the most crucial performance indicator as identifying infection correctly, i.ie, if the cell is infected or not, is extremely important. The sensitivity of VGG-19 is 92.95%, as shown in Table IV and V and Fig.11, which is the highest in comparison to all other transfer learning models, and hence, this depicts that the model is reliable. Floating-point operations per second (FLOPS, flops, or flop/s) measure computer performance relevant in scientific computations where floating-point calculations are required. The number of FLOPs in the three DL architectures are VGG-19 - 19.6 billion FLOPs, Resnet50 - 3.8 billion FLOPs, Alexnet - 0.72 billion FLOPs. Transfer learning involves several layers, each performing various operations and filters. The convolution operation requires high processing power to compute floating-point operations. Hence, the time taken by each

model is directly related to the number of computations. Also, the time taken by the models is highly dependent on trainable parameters. The execution time increases with an increase in the number of FLOPs.

As mentioned in the above architectures, VGG19 has a higher number of trainable parameters than ResNet50 and AlexNet. Hence, the total training time for VGG-19 is greater. But on the other hand, more parameters extract more features, thus providing better results. A comparative study of accuracy is shown in Figure 11. It suggests that VGG-19 gave the best result among all other models with a training accuracy of 95.28% and test accuracy of 93.89%. The proposed method has been compared with the results from Vijayalakshmi A. et al. [8] as depicted in Fig. 12 and Fig. 13. The proposed method gives better results for all the performance measures for the VGG-19 architecture as shown in Fig. 12. Similarly, the proposed method outperforms the accuracy, sensitivity, and F1-score for the AlexNet model, as shown in Fig. 13. VGG-19 model is preferred for the deployment of a web-based application to provide a dependable framework for the automated detection of malaria parasites.

TABLE IV
COMPARATIVE RESULT OF ALL MODELS IN TERMS OF ACCURACY, PRECISION & SENSITIVITY

Model	Accuracy	Precision	Sensitivity
VGG19	93.89%	93.77%	92.95%
ResNet50	80.24%	77.31%	81.05%
AlexNet	82.68%	75.72%	92.07%
SVM	80.46%	80.29%	79.2%

TABLE V
COMPARATIVE RESULT OF ALL MODELS IN TERMS OF SPECIFICITY, F1 SCORE & TRAINING TIME

Model	Specificity	F1 Score	Training Time
VGG19	94.69%	93.33%	4023 sec
ResNet50	79.5%	79.10%	1594 sec
AlexNet	74.6%	83.06%	2535 sec
SVM	81.5%	79.74%	1056 sec

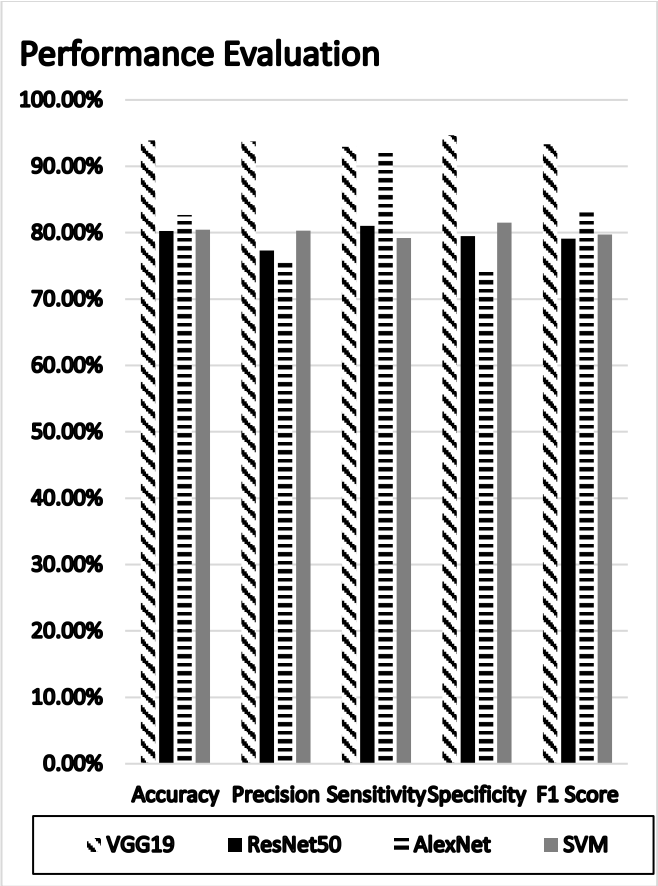


Fig. 11. Comparative Study of Performance Evaluation Metrics for all the models

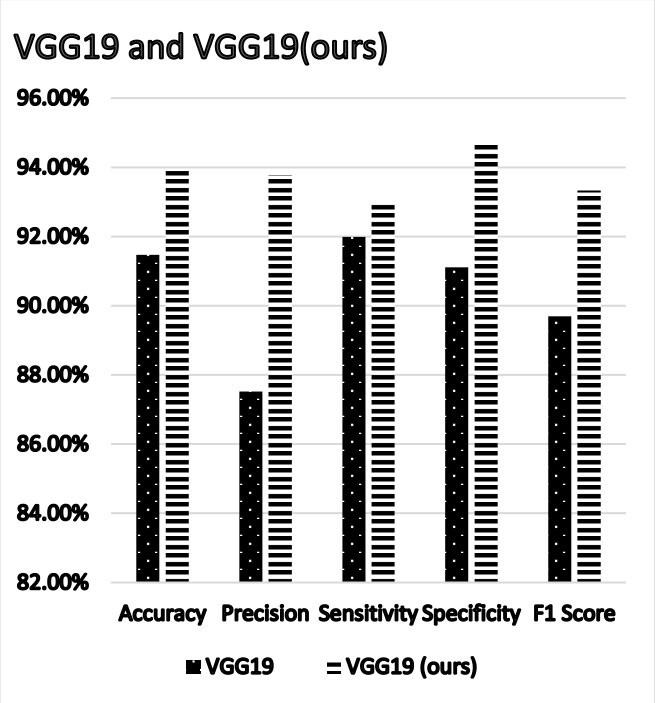


Fig. 12. Comparative Study of Performance Evaluation Metrics using VGG19

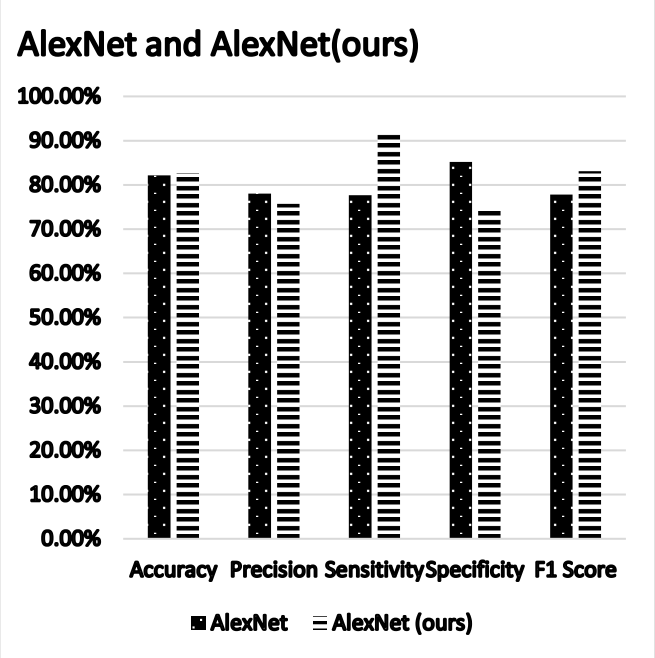


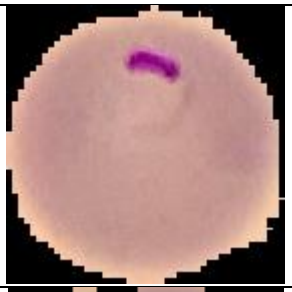
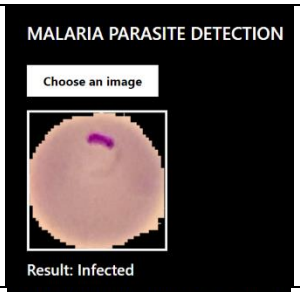
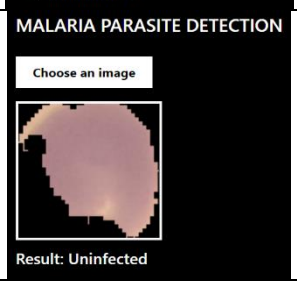
Fig. 13. Comparative Study of Performance Evaluation Metrics using AlexNet

B. Web Application

A front-end was created by using a Flask framework built in Python. It allows easy creation of web pages using a simple Python file. Saved trained model is imported using an app.py file. The file was linked to an HTML, CSS, and JavaScript file for building a simple webpage. The webpage contains a button to choose an image from the device. After selecting an image, it displays the image and inquires for prediction with the help of

a predict button. After clicking on the predict button, it runs through the saved model, predicts the output node value, runs through the app file logic, and displays whether the cell is infected or uninfected. The results from the web page created can be seen below in Table VI. After clicking the predict button, it took less than 2 seconds to load the image, and the result was displayed within 5 seconds.

TABLE VI
PREDICTION OF CELL

ORIGINAL IMAGE	PREDICTION
	
	

IV. CONCLUSION

The introduction of new deep learning methodologies, which have already made an impact, has ushered in an exciting new development. Many of the handcrafted features employed so far may become obsolete when deep learning takes up the onerous process of developing features for categorization from the user. If enough training data is supplied to Neural Networks, they can learn how to process varied staining. With all these advancements, automated detection is well on its way towards becoming a low-cost, quick, and reliable approach for diagnosing malaria. This research work evaluated pre-trained CNN-based deep learning models AlexNet, ResNet50, VGG19, and a comparative study has been performed. Feature extraction using VGG19 outperforms ResNet50 and AlexNet with a test accuracy of 93.89%. The study also compares the transfer learning approach with a non-neural network SVM model. The analysis shows that the transfer learning approach is best suited for automatic detection of malaria parasites than the SVM approach. This VGG19 model was deployed into a web-based application to facilitate a reliable framework for the automated detection of malaria parasites.

Further, the aim is to explore hybrid models for detecting malaria parasites and their types in the infected blood cells.

REFERENCES

[1] WHO, “World Malaria Report”, World Health Organization, Online: https://www.who.int/publications/i/item/9789240015791, 2020.

[2] Cox-Singh, Janet, et al., "Plasmodium knowlesi malaria in humans is widely distributed and potentially life threatening," Clinical infectious diseases, vol. 46, no. 2, pp. 165-171,2008.

[3] Mehrjou, Arash, Tooraj Abbasian, and Morteza Izadi, "Automatic malaria diagnosis system." In 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), pp. 205-211. IEEE, 2013.

[4] McKENZIE, F. ELLIS, et al., "Dependence of malaria detection and species diagnosis by microscopy on parasite density," The American journal of tropical medicine and hygiene , vol. 69,no. 4, pp. 372,2003.

[5] Roy, Kishor, et al.,"Detection of malaria parasite in giemsa blood sample using image processing," AIRCC's International Journal of Computer Science and Information Technology, vol.10, no.1,pp. 55-65,2018.

[6] Suryawanshi, S., and V. V. Dixit, "Comparative study of Malaria parasite detection using euclidean distance classifier & SVM," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol.2, no.11,pp. 2994-2997,2013.

[7] Saiprasath, G., et al.,"Performance comparison of machine learning algorithms for malaria detection using microscopic images," IJRAR19RP014 Int. J. Res. Anal. Rev. (IJRAR), Vol.6, no.1, pp.86-90, 2019.

[8] Vijayalakshmi, A., "Deep learning approach to detect malaria from microscopic images," Multimedia Tools and Applications,vol. 79, no. 21 , pp.15297-15317,2020.

[9] Rajaraman, Sivaramakrishnan, et al., "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," PeerJ, vol.6, e4568,2018.

[10] Nakasi, Rose, et al., "A new approach for microscopic diagnosis of malaria parasites in thick blood smears using pre-trained deep learning models," SN Applied Sciences, vol. 2, no. 7, pp. 1-7, 2020.

[11] Masud, Mehedi, et al., "Leveraging deep learning techniques for malaria parasite detection using mobile application," Wireless Communications and Mobile Computing 2020, pp. 1-15, 2020.

[12] Kalkan, Soner Can, and Ozgur Koray Sahingoz, "Deep learning-based classification of malaria from slide images," 2019 scientific meeting on electrical-electronics & biomedical engineering and computer science (EBBT). IEEE, pp. 1-4, 2019.

[13] Dong, Yuhang, et al., "Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells," 2017 IEEE EMBS international conference on biomedical & health informatics (BHI). IEEE, 2017.

[14] Liang, Zhaohui, et al., "CNN-based image analysis for malaria diagnosis," 2016 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp. 101-104, 2016.

[15] Das, Dev Kumar, et al., "Machine learning approach for automated screening of malaria parasite using light microscopic images," Micron 45, pp. 97-106, 2013.

[16] Feng Yang et al., "Malaria Thick Blood Smears." Web, 2018, doi: https://dx.doi.org/10.21227/qsqw-a673.

[17] Yang, Feng, et al., "Deep learning for smartphone-based malaria parasite detection in thick blood smears," IEEE journal of biomedical and health informatics, vol. 24, no. 5, pp. 1427-1438, 2019.

[18] Poostchi, Mahdieh, et al., "Image analysis and machine learning for detecting malaria." Translational Research 194, pp. 36-55, 2018.

[19] Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, pp. 1409-1556, 2014.

[20] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 84-90, 2012.

[21] He, Kaiming, et al., "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[22] Cortes, Corinna,and Vladimir Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[23] Fuhad, K. M., et al., "Deep learning based automatic malaria parasite detection from blood smear and its smartphone-based application," Diagnostics, vol. 10, no. 5, p 329, 2020.

[24] P. Zongo, R. Dorville, and E. Gouba, “Method for identifying spatial reservoirs of malaria infection and control strategies,” IAENG International Journal of Applied Mathematics, vol. 48, no. 1, pp. 33-39, 2018.