MALARIA DETECTION CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK MODELS - A COMPARATIVE STUDY

DIGITAL IMAGE PROCESSING COURSE PROJECT ASSIGNMENT (SUBMISSION OF RESEARCH PAPER)

Muhammad Asif[*]
Department of Computer Science
COMSATS Univeristy Islamabad
Park Rd, Islamabad
Islamabad Capital Territory 45550
fa19-rcs-038@student.comsats.edu.pk

Dr. Usman Yaseen [†]
Department of Computer Science
COMSATS Univeristy Islamabad
Park Rd, Islamabad
Islamabad Capital Territory 45550

June 2, 2024

ABSTRACT

In conventional methods of diagnosing malaria services of qualified physicians are required by carrying out a cumbersome microscopic blood smear examination.This disease often undiagnosed due to human error. The shortcomings of traditional diagnostic techniques are being enhanced by using Machine Learning and such other computational algorithms of computer science. These novel approaches effectively diagnose malaria through blood smear image whether it is parasitized or uninfected. In this work, wellknown machine learning and deep learning classification models including random forest, AdBoost classifier, Support Vector Machine (SVM), Artificaial Neural Networks (ANNs), Convolutional Neural Networks (CNNs) models, LeNet-5, AlexNet, ResNet, ResNeXt, SE-ResNet DenseNet and a pre-trained EfficientNet were applied. Same dataset which is obtained from NIH (National Institute Health) for these models find out their performance in term of accuracy, precision, F1 score and recall. By comparing the number of parameters used in these deep learning models. It was much large in outperforming networks depending upon the number of layers used. The proposed network is 8-layered convolutional neural network architecture with three convolutional layers proceeded by three pooling layers and two fully connected dense layers. The number of parameters in this model are only 285,506. The performance of this model compete the other massive layered architectures with accuracy 95%, F1-score 95%, Precision 92.8% and recall 95%.

Keywords Malaria · Blood smear · Machine Learning · Deep Learning · Convolutional Neural Networks · Pre-trained models · Feature extraction · Screening, Computer-aided diagnosis

1 Introduction

Malaria is a life losing disease caused by the Plasmodium parasites and a large part of this glob especially, third world countries have a severe health concern due to this disease. Fever, vomiting, headaches and fatigue are its basic symptoms of and the severity of the disease cause coma or even death [1]. Generally, it happens through bite of female anopheles' mosquitoes that injects the parasite into affected person's blood which then travels to the liver to mature and reproduce [2]. South East Asian countries, Eastern Mediterranean countries, Western of Pacific countries and both the Americas are declared as high risk zones by WHO[5]. The disease, however, is curable it proper initiatives and effective measures

[*]Master of Science in Computer Science student at COMSATS University Islamabad. This paper is written as assignment for Digital Image Processing course. Latex file is available at https://www.overleaf.com/project/5ffb7a5901cc64b779090781

[†]Teacher for Digital Image Processing course

taken against it. It all depends how early the disease is diagnosis. Therefore, rapid diagnosis of the malarial parasites has a great significance.

The Malaria Indicator Survey (MIS) was carried out in Pakistan to measure the progress of interventions in 38 highly endemic districts in order to formulate future malaria elimination strategies. The survey reported that 180 million inhabitants population of Pakistan of which 177 million are at risk of malaria. 3.5 million presumed and confirmed malaria cases annually[6]. In 2016, approximately 731,000 deaths have been reported around the world due to malaria with 90% of them in Africa [7].

Generally, malaria has been diagnosed with the help of blood films in which blood cells are examined [8]. Using this microscopy technique about 167 million blood films were tested in year 2010. This technique is widely used besides its drawbacks, however, it is less costly and less complex as compared to polymerase chain reaction-based diagnosis [9]. Microscopy is very time-consuming operation, typically for a doctor a minimum of 5000 cells must be identified manually to validate, the condition. This totally depends on the skills of microscopist and hence working in an environment having limited resource setup and absence of helpful systems to improve the skills of microscopist lead to affect the diagnostic quality and finally results to wrong diagnostic decisions. Use of computer-aided diagnosis (CAD) systems can help and boost microscopist to enhance the accuracy of blood films classification and malaria detection. [10], [11].

In recent years, various classification experiments have been performed on automated malaria-infected cells in medical images, like approaches focused on machine learning and morphology, however, a lot of advanced expertise and services of trained and skilled surgeon are required for this [10]. Development of artificial intelligence (AI) based systems or computer-aided diagnosis or decision support systems have make malaria detection using blood films more efficient. Deep learning (DL), one of the most recent AI technique, which can be used to classify cell images and help to prevent wrong diagnostic decisions. Current modern computer aided systems use deep learning algorithms for medical image analysis [12]. Automated diagnostic system through several machine learning techniques enable human specialist to make correct diagnostic [10].

Machine learning, the subset of Artificial Intelligence are playing a vital role in the field of medical image processing, computer-aided diagnosis, image interpretation, image fusion, image registration, image segmentation, image retrieval and analysis and computer vision. One of the most recently use of ML in computer aided diagnosis and medical image analysis is the classification of objects such as classification based on input features like contrast, area obtained from segmented objects. The important deep-learning techniques includes Support Vector Machine (SVM), Neural Networks (NN), deep learning algorithms such as Convolutional Neural Network (CNN), Recur-rent neural Network (RNN), Long Short term Memory, Extreme Learning Model etc. The deep learning being a powerful technique will become the mainstream technology in medical image analysis in upcoming decades for its higher performance [13]. Researchers are paying attention to deep leaning and its applications are growing exponentially [14]. Deep learning is the most effective and supervised machine learning approach. It has got great interest in almost each and every field particularly in medical image analysis; Data Bridge Market Research analyses that the Medical imaging market is expected to gain market growth in the forecast period of 2020 to 2027 with a CAGR (Compound annual growth rate) of 10.4% in this period and it is expected to reach USD 265,334.23 million by 2027 from USD 120,237.71 million in 2019 [15].

The term deep learning means the use of a deep neural network model and neuron is the basic computational unit in it. The concept is taken by the study of the human brain, which takes multiple signals as inputs, combines them linearly using weights, and generate output signals after passing the combined signals through nonlinear operations. Introduction of GPUs have favoured the research in this field and since the introduction of challenge, a sudden rapid growth in development of such models will be seen. The accurate and the most useful type of deep learning models for analysis of images are convolutional neural networks (CNN) and it is used widely in the field of computer vision for the diagnosis and classification of medical disease. There is no need for any feature extraction before the training process [17], [16]. Generally, CNN's are designed to minimize or eliminate data preprocessing steps. They are compatible to deal with raw images or data [18], [19]. Due to outperformance over other network architectures especially on visual data, CNN gain popularity since 2010. However, the concept behind the functionality of CNN is not new and obtained from human visual system. Neural Network is the domain of AI which aims at mimicking the brain structure of human body and Artificial Neural Network (ANN) is the basic type which is efficient in finding the hidden pattern in the provided data and give out the results. Since image is nothing more than collection of numerical pixel values. ANNs are capable to well classify it, however, it lacks the consideration of spatial relationships while processing the image. For example, while processing the list of images, if ANN finds that an object occurs in the top right corner of the image it will assume that it always appear there. Hence no matter where the object in is placed in the image. If not found in the top right corner of the image, the result will be negative. On the other hand Convolutional Neural Network, performs exceptionally well in finding the object anywhere in the image. Convolution takes two functions and gives a third function showing how the shape of one function is affected by the other one. Normally in photoshop applying a

filter to an image (say highlighting the shadows). This will activate certain features of the image and produce an output image. Actually, filters are nothing but 3D matrix having some values that get excited when they found desired pattern in neighborhood. They are considerably smaller in size of the input image and slides over its whole area. The size of output image is bound to reduce after the convolution operation and it depends upon filter size, stride (movement steps) and padding (insignificant cover around the image). To introduce non-linearity we use Sigmoid and ReLU (Rectified Linear Unit) are activation function to transforms the numbers to range between 0 and 1 and thresholding input at 0 respectively. Pooling is applied to downsample the image whose main purpose is for dimensionality reduction. Fully connected (FC) layer used to learn through the non-linear combinations of high level features extracted by the convolutional and pooling layers. These layers accepts the input in 1D vector form so they are first flatten the output received from previous convolutional or pooling layer to feed to FC layer. The output of these layers are the real values of each class (in case of classification problem) and to check these values softmax function is used which converts these real values into class probabilities. The class with the highest probability is then extracted as the predicted class. These layers are the basic building blocks of CNN. Depending on the nature of the problem, multiple other types of layers such as DropOut Layer and Normalization Layer can be added. The former is used to prevent a model from overfitting and later for training very deep neural networks that standardizes the inputs to a layer for each mini-batch.

2 Problem Statement

Malaria is amongst the major diseases causes the highest mortality worldwide and can turn fatal if not taken seriously. Rapid and quick diagnosis of the disease is the key to survive. However, manual diagnosis is time consuming and tedious due to the large amount of image data and non-availability of required number of qualified technicians who examine malarial disease through the examination of blood smears under the microscope for parasite-infected red blood cells.

The inadequacy of this traditional method is enhanced using advanced computer vision and deep learning methods to automatically classify the malarial parasite from the microscope's blood smear image as parasitized (infected) or uninfected. Convolutional Neural Networks (CNN), a class of deep learning (DL) model with end-to-end feature extraction and classification promises highly scalable and superior results helps to get a high classification accuracy on a wide variety of image datasets as compared to machine learning architectures like SVM, RandomForest and AdaBoost etc. However, these Deep Learning models have a very high computational complexity and so incur a high computational cost of running these algorithms as well as make it hard to interpret the results. This can be managed by hyper parameter tuning adding dropout and normalization layers to these network and necessary preprocessing.

3 Related Work

Liang et al. have proposed a deep learning based approach for the classification of malaria infected cells from red blood smears. Their transfer learning-based model using the AlexNet architecture [20] pre-trained on the CIFAR-100 dataset [21] that outperforms, however, the proposed method is based on a 16-layer convolutional neural network. Dong et al. [26] have used a data comprising around 1000 training and testing samples only, and hence have employed transfer learning and reported the results on LeNet [22, 23], AlexNet [24] and GoogleNet [25] architectures were used to learn the inherent features of the malaria infected cells and the non-infected cells. For comparison, a support vector machine (SVM) was trained on pre-selected features extracted from the same dataset. A classic approach has been proposed by Bibin et. al [27] on deep belief network (DBN) consisting of 4 hidden layers pretrained by stacking restricted Boltzmann machines [28] using contrastive divergence method [29] for pre-training. Rajaraman et, al [10] evaluate the performance of custom and pretrained CNNs and construct an optimal model ensemble toward the challenge of classifying parasitized and normal cells in thin-blood smear images the ensemble model constructed with VGG-19 and SqueezeNet outperformed the state-of-the-art in several performance metrics toward classifying the parasitized and uninfected cells to aid in improved disease screening.

4 Data Description

Malaria datasets which is being used is available online which is released by NIH (National Institute of Health). Dataset consists of thin blood smear slides images of segmented cells obtained through Malaria Screener research activity. Quantity of data is 27610 images, out of which 13779 images are parasitized (infected) and 13831 are uninfected images. Which are almost equal in number, however, make the dataset fully balanced 27558 images are taken (i.e. 13779 images each for parasitized and uninfected class). Now both the classes of the dataset have equal instances, this contribute towards solving the overfitting of model during training.

Table 1: Summary of the Related Work

Reference	Model	Size	Accuracy	Sensitivity	Specificity	Precision	F1 score	Matthews correlation coefficient (MCC)
Liang et al. [20]	16-layer AlexNet pre-trained	(227x227)	97.37%	96.99%	97.75%	97.73%	97.36%	94.75%
Dong et al. [26]	LeNet	(60x60)	96.18%	95.36%	96.74%	95.17%	95.27%	92.7%
	AlexNet	(60x60)	95.79%	97.10%	94.91%	92.79%	94.90%	91.37%
	GoogLeNet	(256x256)	98.13%	97.29%	98.69%	98.05%	97.67%	96.11%
	SVM	7 features out of 76	91.66%	96.71%	88.25%	84.75%	90.33%	83.62%
Bibin et al. [27]	4 Layered Model 200 nodes	-	96.30%	97.91%	95.93%	84.84%	90.91%	88.96%
Rajaraman et, al [10]	VGG-19	47 (224x224)	99.32%	-	-	99.71%	99.31%	98.62%
	SqeezNet	68 (227x227)	98.66%	-	-	99.44%	98.64%	97.32%
	Inception ResNet-V2	825 (229x229)	99.79%	-	-	99.56%	98.77%	97.359%

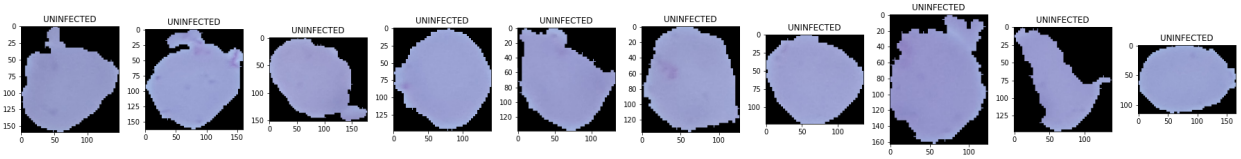


Figure 1: Uninfected sample red blood cells images of NIH dataset

The size of these patches of segmented red blood cells varied from 110 to 150 pixel and color images (3-channel RGB). Figure 1 and Figure 2 shows first 10 samples each from uninfected and parasitized images respectively from the dataset.

5 Pre-processing the Data

Various preprocessed techniques applied on the dataset to make it suitable for classification models and further analytical operations.

- **Resizing**. As the images size of the NIH dataset are ranging from 110 to 150 pixel in both the direction height and width. Equalize the size of all the images which is fixed as 50x50 in our case.Lorem ipsum dolor sit amet
- **Numeric Labels**. The classes of data are uninfected and parasitized which are strings. For convenience these classed are replaced with '0' and '1' respectively.

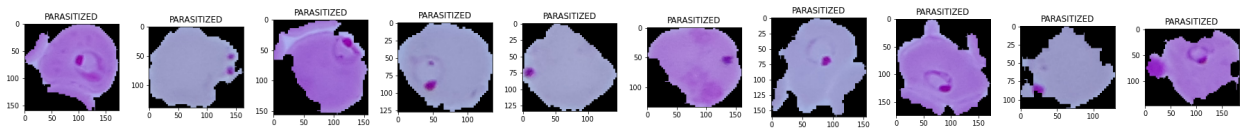


Figure 2: Parasitized samples images taken from NIH dataset. Different forms of the parasite can be seen in these red blood cells.

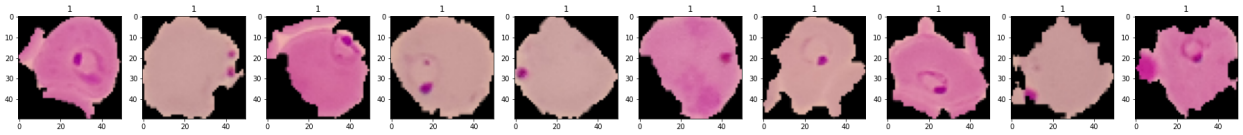


Figure 3: Figure shows the resized images of dataset with numeric labels. However, first ten samples are of the same class.

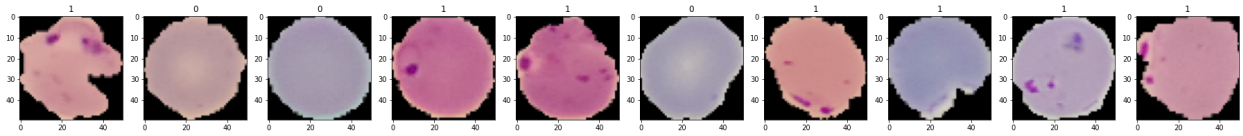


Figure 4: Figure shows preprocessed images of dataset with numeric labels. Samples of both the classes are seen.

- **Merging**. Since dataset is available in two separate folders parasitized and uninfected and to make. Data is merged in a single array of images.
- **Shuffling**. After merge operation data is stacked in the array in the order in which data is merged. Imaged of both parasitized and uninfected are lying in the group form. So data is shuffled so that data may be scattered throughout the dataset. Figure 3 and Figure 4 clearly differentiate it.
- **Min-Max Normalization (Image Re-scaling)**. Red blood cell patches are 8-bit RGB color images (i. maximum pixel value in each channel is 255 and minimum value is 0. Image patches are rescaled to map the features ranges from 0 to 1. This can easily be done by dividing each pixel to 255. Thus resultant image pixel values now ranges from 0 to 1 rather 0 to 255. It helps faster convergence.
- **Standardization**. Another technique of feature standardization is applied to in which values of each feature in the data is rescaled in such a way that it has mean 0 and standard deviation 1.
- **Data Enhancement (Data Augmentation)**[30]. In order to achieve better robustness, increase the diversity of data like spatial diversity of the image by means of rotation, mirroring and cropping.

5.1 Data Splitting

Data is randomly splitted into two sets with 80:20 ratio. The bigger one termed as training set and the smaller as testing or validation set. Now training set has 22046 instances whereas test set has 5512 instances.

6 Proposed Model

The proposed model consists of eight layers in total out of them three convolutional layers, three pooling layer each after convolutional layer for down-sampling and batch normalization is applied to after each pooling layer. Rest of the two layers are fully connected desnse layers. Input image size is 50x50 and all three channels are use being color image.This 3 channelled 50x50 image is processed by first convolutional layer with 3x3 sized 32 filters with stride 1 and activation Relu. This produce 32 channelled 48x48 images. Max pooling of 2x2 with stride 2 applied to this layer for down sampling. The resultant 32 channelled 24x24 image is input for further convolution after applying batch normalization. Similar sequence is adopted with similar parameters for second and third convolutional and pooling layer and batch normalization also. The resulted image size becomes 4x4 with 32 channels. Here flatten the data and fully connected layer is applied which has (4x4x32=512) nodes and they act as input to the next fully connected layer which is output layer. Figure 5 and Figure 6 illustrates the architecture of the proposed model.

The number of parameters in the model depends upon the neuron used in the layers of the model more the deeper network and more the channels in the layers, more will the the parameters which effect the computational cost of the network and thus performance is affected. The total number of learnable parameters used to train our model are 285,506. Input layer has nothing to learn so it has zero parameter also pooling layer just down sample the data and thus has zero learnable parameter. Convolutional layer and fully connected layers are where CNN learns. To calculate parameters, just multiply by the filter shape of width w, height h, previous layers' number of filters p to the number of filters k in the current layer before adding 1 as bias term attaches with each filter. Equation 1 illustrates the compute the number of

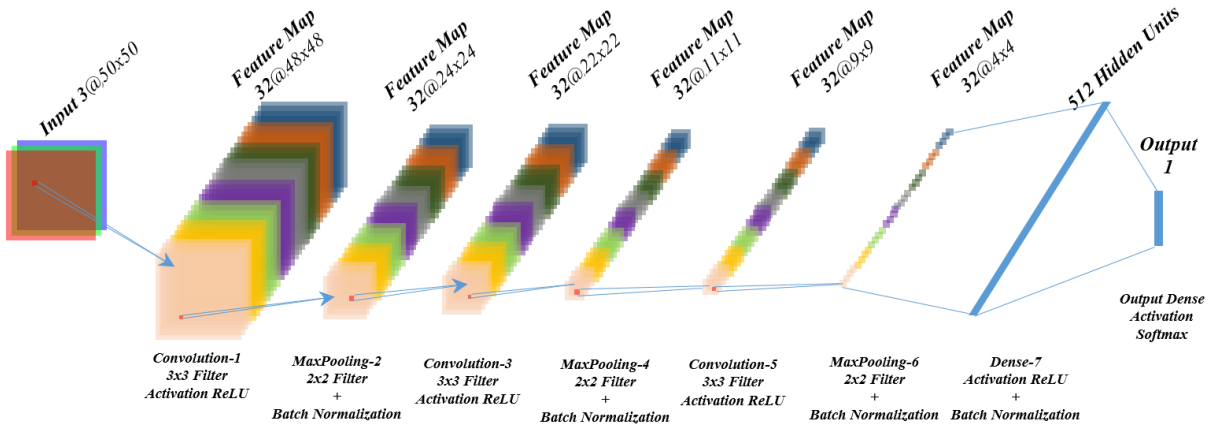


Figure 5: Proposed Optimal network structure overview.

parameters *P* convolutional layers.

$$P_{Conv} = ((w \times h \times p) + 1) \times k \tag{1}$$

There is a slight difference in fully connected layers parameters calculations which is the product of the number of neurons in the current layer *c* and the number of neurons on the previous layer *p* and as always, do not forget the bias term. Thus number of parameters here are:

$$P_{FC} = ((c \times p) + 1 \times c) \tag{2}$$

Batch normalization work during training, they need to keep track of the distributions of each normalized dimensions. To do so, since you are in *mode=0* by default, they compute 4 parameters per feature on the previous layer. Those parameters are making sure that you properly propagate and backpropagate the information. If there are *n* number of filters in the layer then:

$$P_{BN} = 4 \times n \tag{3}$$

Therefore the total number of parameters in the network can be calculate as. If there are *l* convolutional layers and *m* fully connected layers and *n* batch normalization in the network then the total parameters canbe calculated as:

$$P = \sum_{i=1}^{l} P_{Conv}^{i} + \sum_{j=1}^{m} P_{FC}^{j} + \sum_{k=1}^{n} P_{BN}^{k} \tag{4}$$

Figure 7 is the summary of the proposed model in which trainable parameters are calculated.

7 Experimental Setup

The hyperparameters setting is listed below.

- Physical calculation environment: All of these experiments were performed on Google Colab, Google's free cloud service for AI developers, which provides free GPU in Google Chrome browser. The configuration of machine used with Windows 10 operating system with Intel® Core(TM) i7-3540M CPU @3.00GHz (4CPUs) processor, 512 GB SSD, 8 GB RAM, a Google Colab graphical processing unit (GPU), Python® 3.6.7, Keras® 2.2.4 with TensorFlow 1.12.0. GPU speedup over CPU: 80x.
- Number of filters in 2D convolution - (10 to 100)
- Filter size for 2D convolution - (2 to 6)
- Network layer number range:
- Kernel size for 2D max pooling: - (2 to 6)
- Number of units in fully connected layers: (32 to 256)

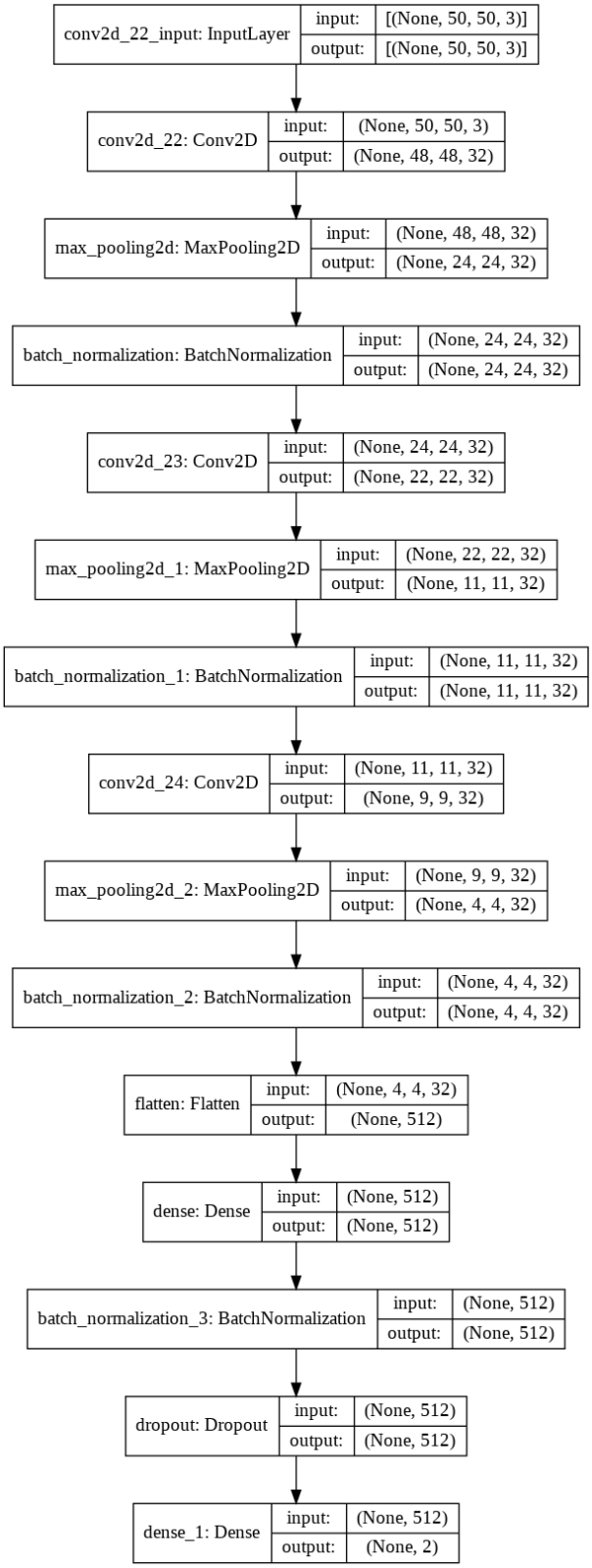


Figure 6: Flow chart diagram of the proposed model.

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_37 (Conv2D)           (None, 48, 48, 32)        896
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 24, 24, 32)        0
_________________________________________________________________
batch_normalization_20 (Batc (None, 24, 24, 32)        128
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 22, 22, 32)        9248
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 11, 11, 32)        0
_________________________________________________________________
batch_normalization_21 (Batc (None, 11, 11, 32)        128
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 9, 9, 32)          9248
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 4, 4, 32)          0
_________________________________________________________________
batch_normalization_22 (Batc (None, 4, 4, 32)          128
_________________________________________________________________
flatten_6 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_13 (Dense)             (None, 512)               262656
_________________________________________________________________
batch_normalization_23 (Batc (None, 512)               2048
_________________________________________________________________
dropout_20 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_14 (Dense)             (None, 2)                 1026
=================================================================
Total params: 285,506
Trainable params: 284,290
Non-trainable params: 1,216
_________________________________________________________________
```

Figure 7: Summary of the model showing the parameters.

- Dropout rate: (0,1)
- The selection for activation function of 2D convolution layer: (ReLU)
- The selection for activation function of fully connected dense layer: (linear, sigmoid, softmax, relu)
- The selection for activation function of last fully connected output layer: (sigmoid, softmax)

8 Evaluation Metrics

Proposed model evaluated using the following standard metrics: Accuracy, Loss, Precision, Recall, F1 score. In our experimental setup, we use accuracy as our optimizing metric and the others are used as satisfying metrics and number of parameters (use of hidden layers) was also took consideration.

9 Methodology

Both machine learning and deep learning models were applies to the image dataset which is preprocessed and dataset was prepared separately both machine learning and deep learning models. For machine learning models first best five features extracted from images and then different machine learning algorithms were applied. Model is compiled with categorical crossentropy loss, adam optimiser and evaluation of accuracy matrix. model trained for 20 epochs with 32 batch size.The histor is shown in Figure 8

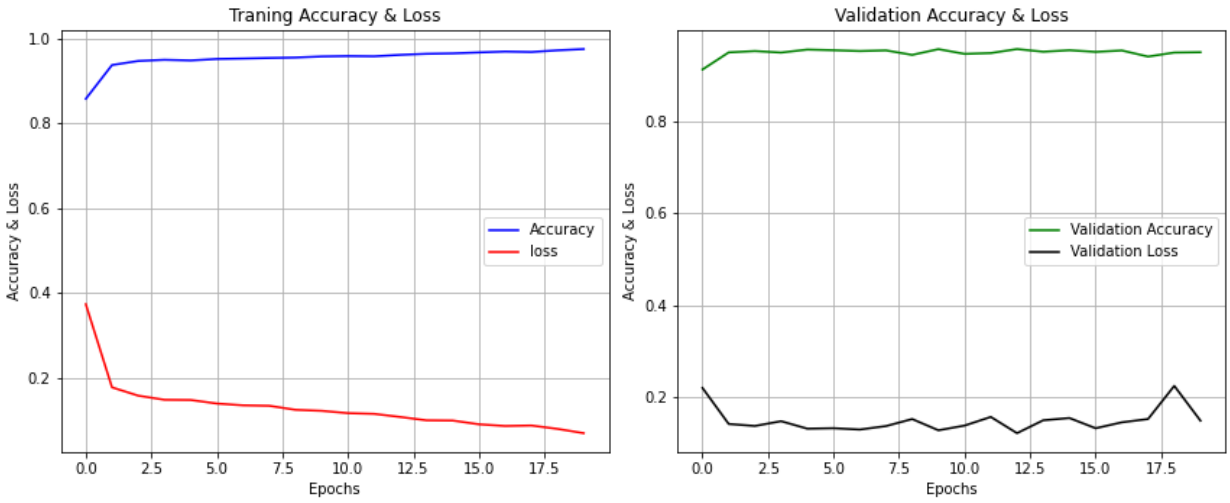


Figure 8: Proposed Model Accuracy and Loss showing the parameters.

Table 2: Accuracy Score of Machine Learning Models trained on best five contour features

Model,	Accuracy,	F1 Score,	Precision,	Recall
Random Forest Model	90.28%	90%	90%	90%
AdaBoost	90.29%	90%	90%	90%
SVM	88.55%	89%	89%	89%
Fine Tuned SVM	90%	90%	90%	90%

10 Results & Discussion

In this paper, the self-organizing network structure is shown in Figure 5. Through a lot of iteration and training, we finally get a model with a 8-layer network structure. All experiments were performed on the training data to obtain an optimized CNN architecture, and the best performing architecture was utilized to calculate the accuracy of the test data. Figure 7 the parameter settings and summary of the model.

10.1 Results of Machine Learning Algorithms

The table 2 clearly shows that the accuracy does not exceed more than 90%. Even though SVM was fine tuned by hyperparameters tuning.

10.2 Results of Deep Learning Algorithms

Deep learning Models incliuds: Artificaial Neural Networks (ANNs), Convolutional Neural Networks (CNNs) models, LeNet-5, AlexNet, ResNet, ResNeXt, SE-ResNet DenseNet and a pre-trained EfficientNet were applied. Same dataset which is obtained from NIH (National Institute Health) for these models find out their performance in term of accuracy, precision, F1 score and recall. By comparing the number of parameters used in these deep learning models. It was much large in outperforming networks depending upon the number of layers used. The proposed network is 8-layered convolutional neural network architecture with three convolutional layers proceeded by three pooling layers and two fully connected dense layers. The number of parameters in this model are only 285,506. The performance of this model compete the other massive layered architectures with accuracy 95%, F1-score 95%, Precision 92.8% and recall 95%. Table 3 show the results:

The following figures show the results of the proposed dataset with the designed CNN architecture. The accuracy and the loss of the training and validation data are shown in Figure 5. The testing confusion matrix and ROC are shown in Figure 6 A and B, respectively. Based on these figures, it can be noticed that the proposed system has classified the images (red blood cells) with an accuracy rate of 98.85of 98.7998.90

Table 3: Accuracy Score of Machine Learning Models trained on best five contour features

Model	Accuracy	F1 Score	Precision	Recall	Parameters
SE-ResNet	**95.9%**	**95.9%**	**95.1%**	**95.9%**	26,108,530
EfficientNet	**95.5%**	**95.5%**	**94.4%**	**95.5%**	4,052,133
CNN	**95.1%**	**95.1%**	**93.8%**	**95.1%**	**515,202**
Proposed CNN	**95%**	**95%**	**92.8%**	**95%**	**285,506**
DenseNet	94.4%	94.4%	92.3%	94.4%	7,045,442
ResNet	94.1%	94.1%	**93.3%**	94.1%	23,591,810
ResNeXt	93.9%	93.9%	92.4%	93.9%	35,228,418
AlexNet	92.4%	92.4%	91.9%	92.5%	926,218,498
LeNet5	91.1%	91.1%	88.1%	91.1%	**205,326**
ANN	68.3%	68.3%	63.5%	68.3%	25,506,002

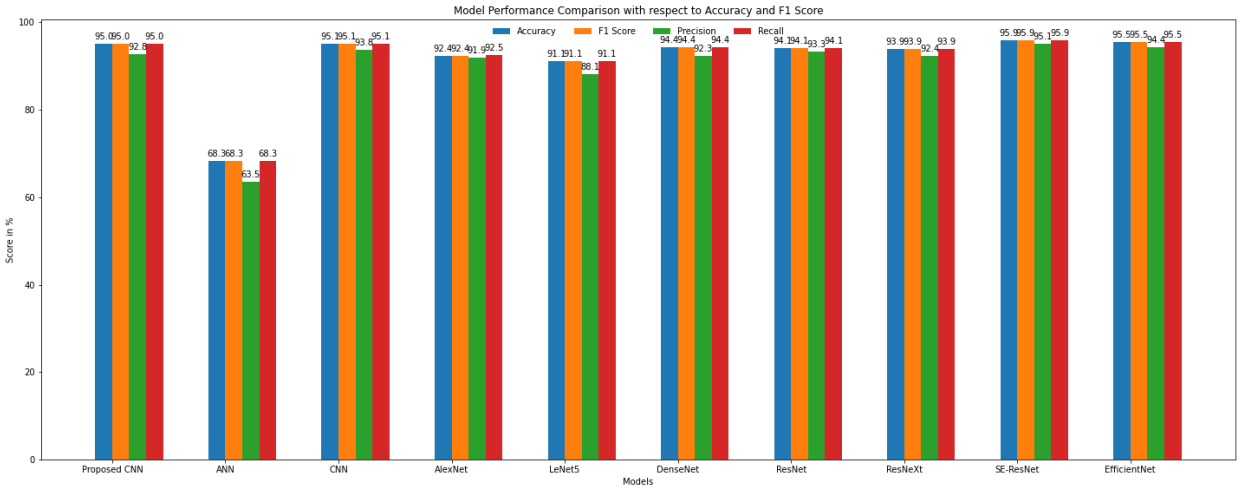


Figure 9: Comparison of Performance of Different Models.

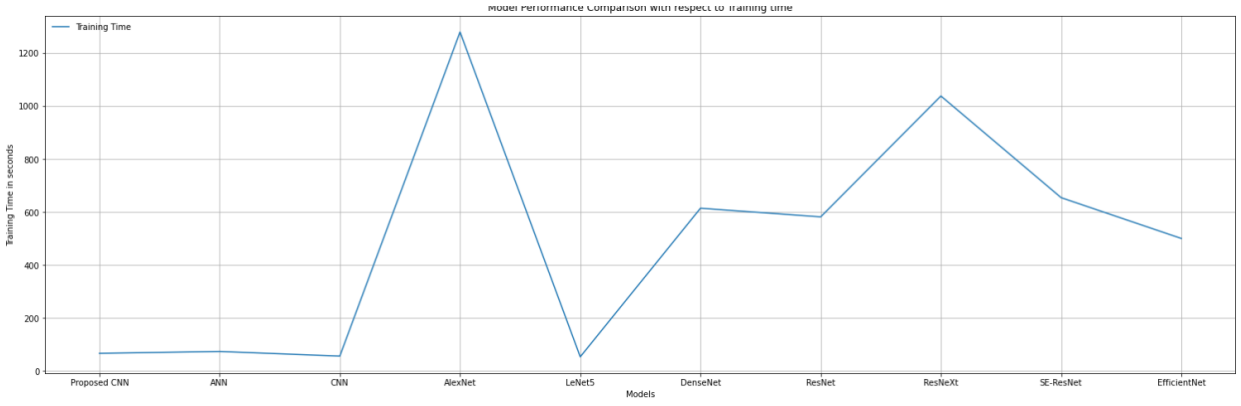


Figure 10: Comparison of training time of the models.

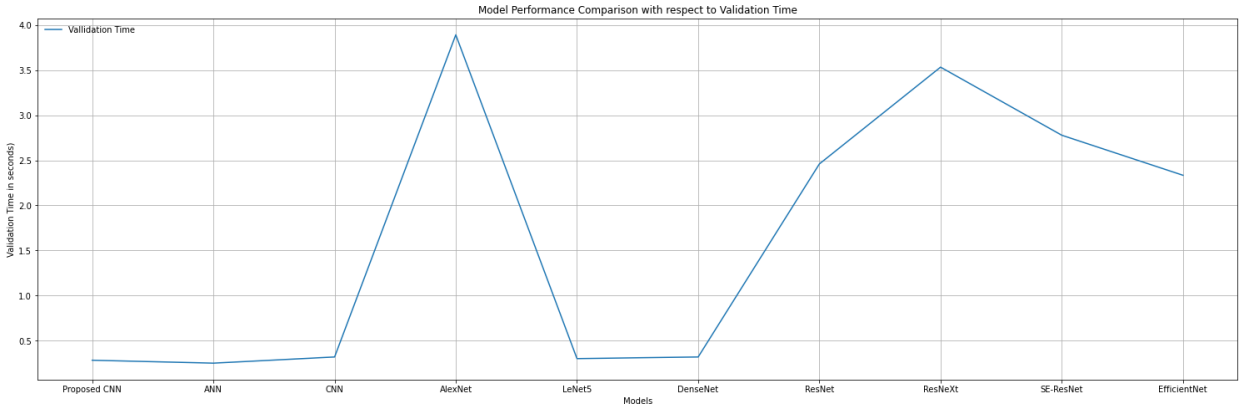
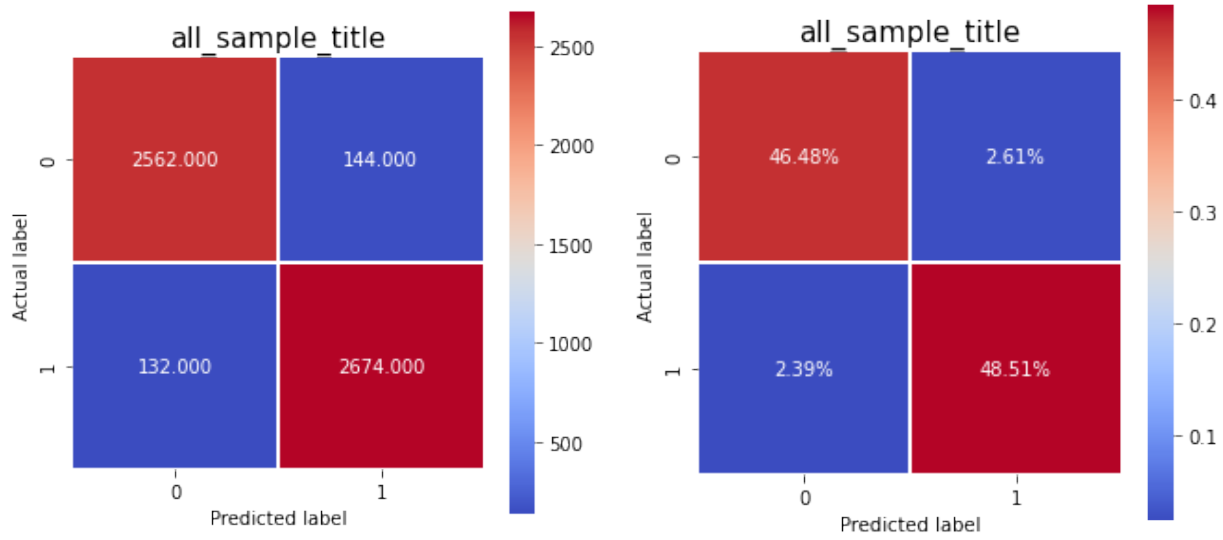


Figure 11: Comparison of Validation time of the models.



(a) Number of values

(b) Values in Percentage

Figure 12: Confusion Matrix

11 Conclusion

In this work, we conduct a series of experiments based on end-to-end deep learning to improve malaria classification from segmented red blood cell smears. We show that, using very deep networks with billions of parameters does not contribute to the model having overall performance. Instead, simple convolutional networks gave better accuracy 95.1%, Accuracy of SE-ResNet and EfficientNet is 95.9% and 95.5% respectively seems better however, the number of parameters are 26,108,530 for the former and 4,052,133 for the later which is pre-trained model, In our case we use a 8 layer customized CNN model which by just add subsampling which cause reduction of parameters which is just 285,506. This low quantity of neurons also cause better training time. Accuracy of our model is 95%, F1 score 95%, precision is 92.8% and recall is 95%.

12 Future Work

Achieved accuracy is likely to be improved by applying some other pre-processing algorithm like augmentation and image enhancement. Fine tuning of the model by changing hyper-parameters can also make it more efficient.

References

[1] World Health Organization. Malaria: fact sheet. No. WHO-EM/MAC/035/E. World Health Organization. Regional Office for the Eastern Mediterranean, 2014

[2] Caraballo, Hector, and Kevin King.. *Emergency department management of mosquitoborne illness: malaria, dengue, and West Nile virus.* Emergency medicine practice 16, no. 5 (2014): 1-23

[3] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.

[4] Fact Sheet about Malaria. 2019. Available online: https://www.who.int/news-room/fact-sheets/detail/malaria (accessed on 29 December 2020). Geneva: World Health Organization. World malaria re- port 2019. Technical report, 2019.

[5] Fact Sheet about MALARIA. 2020. Available online: https://www.who.int/news-room/fact-sheets/detail/ malaria (accessed on 12 December 2020)

[6] Malaria Indicators Survey In 38 High Risk Districts of Pakistan 2013-14

[7] Wang, Haidong, Mohsen Naghavi, Christine Allen, Ryan M. Barber, Zulfiqar A. Bhutta, Austin Carter, Daniel C. Casey et al. *"Global, regional, and national life expectancy, allcause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015.* The lancet 388, no. 10053 (2016): 1459-1544.

[8] Wilson, Michael L. *Malaria rapid diagnostic tests.* Clinical infectious diseases 54, no. 11 (2012): 1637-1641

[9] Nadjm, Behzad, and Ron H. Behrens. *Malaria: An update for physicians.* Infectious Disease Clinics 26, no. 2 (2012): 243-259.

[10] Rajaraman, S., Jaeger, S., Antani, S.K. (2019). Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images. PeerJ, 7: e6977. https://doi.org/10.7717/peerj.6977

[11] Reddy, A.S.B., Juliet, D.S. (2019). Transfer learning with ResNet-50 for malaria cell-image classification. 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, pp. 0945- 0949. https://doi.org/10.1109/ICCSP.2019.8697909

[12] Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez. *A survey on deep learning in medical image analysis.* Medical image analysis 42 (2017): 60-88.

[13] Shital. Patil., Surendra. Bhosale. (2019). *Machine learning Applications in Medical Image Analysis.* International Conference on Trends in Electronics Communication (ICTEC), Thakur College of Engineering and Technology, Mumbai, India. Volume 6, Issue 6, June 2019

[14] Poostchi, M., Silamut, K., Maude, R.J., Jaeger, S., Thoma, G. (2018). analysis and machine learning for detecting malaria. Translational Research, 194: 36-55. https://doi.org/10.1016/j.trsl.2017.12.004

[15] https://www.pharmiweb.com/press-release/2021-01-07/medical-imaging-market-size-covid-19-impact-analysis-forecast-to-2027-spectrum-health-ramsof (accessed on 9 Jan 2021)

[16] Pan, W.D., Dong, Y., Wu, D. (2018). *Classification of malaria-infected cells using deep convolutional neural networks.* Machine Learning: Advanced Techniques and Emerging Applications, 159. https://doi.org/10.5772/intechopen.72426

[17] Alqudah, A.M., Alquraan, H., Qasmieh, I.A. (2019). *Segmented and non-segmented skin lesions classification using transfer learning and adaptive moment learning rate technique using pretrained convolutional neural network.* Journal of Biomimetics, Biomaterials and Biomedical Engineering, 42: 67-78. https://doi.org/10.4028/www.scientific.net/JBBBE.42.6 7

[18] Vijayalakshmi, A., Rajesh Kanna, B (2019). *Deep learning approach to detect malaria from microscopic images.* Multimedia Tools and Applications, 79: 15297- 15317. https://doi.org/10.1007/s11042-019-7162-y

[19] Rahman, A., Zunair, H., Rahman, M.S., Yuki, J.Q., Biswas, S., Alam, M.A., Alam, N.B., Mahdy, M.R. (2019). *Improving malaria parasite detection from red blood cell using deep convolutional neural networks.* arXiv preprint arXiv:1907.10418. 2019 Jul 23.

[20] Liang, Zhaohui, Andrew Powell, Ilker Ersoy, Mahdieh Poostchi, Kamolrat Silamut, Kannappan Palaniappan, Peng Guo et al. *CNN-based image analysis for malaria diagnosis.* In 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 493-496. IEEE, 2016.

[21] Krizhevsky, Alex, and Geoff Hinton. *Convolutional deep belief networks on cifar10.* Unpublished manuscript 40, no. 7 (2010).

[22] 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.

[23] Y. LeCun, K. Kavukvuoglu, and C. Farabet. *Convolutional networks and applications in vision,* in Proc. International Symposium on Circuits and Systems (ISCAS'10). IEEE, 2010.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Imagenet classification with deep convolutional neural networks.* In Advances in neural information processing systems, 2012, pp. 1097–1105.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions,* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9

[26] Dong, Yuhang, Zhuocheng Jiang, Hongda Shen, W. David Pan, Lance A. Williams, Vishnu VB Reddy, William H. Benjamin, and Allen W. Bryan. *Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells.* In 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), pp. 101-104. IEEE, 2017.

[27] Bibin, Dhanya, Madhu S. Nair, and P. Punitha. *Malaria parasite detection from peripheral blood smear images using deep belief networks.* IEEE Access 5 (2017): 9099- 9108

[28] Salakhutdinov, Ruslan, and Geoffrey Hinton. *Deep boltzmann machines.* In Artificial intelligence and statistics, pp. 448-455. 2009.

[29] Carreira-Perpinan, Miguel A., and Geoffrey E. Hinton. *On contrastive divergence learning.* In Aistats, vol. 10, pp. 33-40. 2005

[30] Yan, Zhicheng. *Automatic photo adjustment using deep neural networks.* ACM Transactions on Graphics (TOG), vol. 35, pp. 11-15, May. 2016.

[31] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.

[32] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.