International Journal of Future Generation Communication and Networking
Vol.14, No. 1, (2021), pp. 4331 - 4342

Comparative Analysis of Deep learning architectures for automated diagnosis of malaria parasite.

Prof.N.N. Kulkarni[1], Prof. K. Kasture[2], Parth Waghmare[3], Pratyush Gavali[4], Nitin Bildani[5]
[1,2] Assistant Professor, Department of E&TC, SKNCOE, SPPU, Pune
[3,4,5] Student, Department of E&TC, SKNCOE, SPPU, Pune
nileshkulkarni992@gmail.com[1]
koki.thakur@gmail.com 2
wparth2099@gmail.com [3]
gavlipratyush123@gmail.com [4]
nitinbildani.2108@gmail.com [5]

Abstract

This paper proposes the study of implementation and comparative analysis of pre-trained Convolutional Neural Network (CNN) models architectures on the basis of accuracy for malaria parasite detection. Malaria caused by the Plasmodium parasites, is a blood disorder, which is transmitted through the bite of a female Anopheles mosquito. It is detected by trained microscopists who analyse microscopic blood smear images. In recent times, Machine learning technologies have been used for automated diagnosis of malaria. Manual evaluation for diagnosis requires various steps to be performed. Moreover, this process leads to overdue and misguided analysis, even when it comes to the hands of expertise. The performance of pre-trained Convolutional Neural Network (CNN) dependent DL models as feature extractors for classifying parasitized and uninfected cells is evaluated in this study to help in improved disease screening. We take advantage of transfer learning methodology by examining pre-trained VGG-16, Xception, convolutional neural network (CNN) models with adjusted, densely connected classifiers. We compare the obtained results and visualized them. The proposed models have been evaluated on a dataset containing 13789 healthy and 12188 infected images. The dataset used was taken from National Institute of Health named NIH Malaria Dataset. The use of pre-trained Convolutional Neural Network (CNN) as a promising method for feature extraction for this reason is demonstrated by statistical confirmation of the results.

Keywords— *Malaria parasite detection, convolutional neural network, Transfer Learning, deep learning, Pre-trained model.*

I. INTRODUCTION

Malaria is a life-threatening disease spread by bite from infected female Anopheles mosquitoes. In 2015, there were 214 million cases of malaria and 438,000 fatalities, according to a data released by the World Health Organization (WHO)[1]. Malaria is usually only detected through a manual examination of a microscopic slide. Four human red blood cell samples were produced using entire slide pictures in Fig. 1. Necessary training and specialised human resources are required to offer a trustworthy diagnosis. Unfortunately, these materials are in short supply and are usually inaccessible in underdeveloped areas where malaria is widespread. As a result, an automated diagnostic system can offer a quality solution to this issue. Machine learning algorithms have recently gotten a lot of interest from academics because of their capacity to construct automated diagnosis systems for malaria [2], [3].

SVM and Naive Bayes Classifier were used in [2] to obtain 84 % and 83.5% accuracy, respectively. Unsupervised learning, such as K-Nearest Neighbour (KNN), has been proposed to

ISSN: 2233-7853 IJFGCN

recognise malaria infected cells in [3], in contrasts to supervised learning. A three-layer Neural Network (NN) was built as a classifier in [4], with an accuracy of 85% in detecting malaria infected cells. Even though these supervised learning algorithms have achieved some success in terms of virus detection accuracies, their results are highly dependent on the feature extraction method utilised. As a result, constructing a discriminatory feature vector with low redundancy is critical. There has been a lot of work done to extract features for malaria-infected cells [2], [3], [5].A detailed discussion of feature extraction and optimization for malaria cells may be found in [6]. Although a good feature extraction method can enhance detection accuracy, this sort of virus detection cannot achieve entirely automated diagnosis because trained specialists must still manually extract feature vectors based on specific datasets. A deep convolutional neural network (CNN) was used to identify malaria in a thick blood smear in [7]. Pathologists, on the other hand, find it difficult to separate infected and non-infected samples in thick films since the distinction is not as evident as it is in thin films, where individual red blood cells can be cropped from complete slide images. The spatial local correlation among neighbouring pixels/voxels is an essential source of information for photos. Convolutional Neural Networks (CNN) are a type of deep learning model that uses methods like local receptive fields, shared weights, and pooling to utilise this information[31]. Alex Krizhevsky introduced AlexNet, a CNN-based deep learning model that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 and significantly improved CNN classification performance [31] Several sample CNNs, including VGGNet [24] and ResNet [30], showed considerable improvements in ILSVRC annual challenges. DL tools are currently being used by researchers all around the world, with promising findings in a wide range of medical picture analysis/understanding applications (Rajaraman et al., 2017; Suzuki, 2017). Studies on using DL techniques to detect malaria parasites have also been found in the literature. SVM and pre-trained DL models such as LeNet (LeCun et al., 1998), AlexNet, and GoogLeNet were compared to classify parasitized and uninfected cells by Dong et al. (2017). The authors randomly separated the red blood cells (RBCs) into train and test sets after segmenting them from thin blood smear pictures.We suggest using deep learning algorithms for malaria cell detection in this paper, with the ultimate goal of creating a fully automated diagnostic platform that does not require manual feature extraction. Deep machine learning approaches may be a suitable fit in this case. Deep learning algorithms can extract a systematic representation of the data from the input, with higher layers representing increasingly abstract notions that are less susceptible to transformations and scaling. The inherent properties of malaria infected cells and non-infected cells were learned using three well-known deep convolutional neural networks, namely VGG-16, VGG-19 and ResNet-50.We tested the effectiveness of pre-trained CNN-based DL models as feature extractors in identifying parasitized and uninfected cells to aid in disease screening. The following are some of the work's major contributions: A comparison of the performance of pre-trained DL models as feature extractors for distinguishing parasitized and uninfected cells is presented.The following paper is organised as follows: Materials and Methods delves into the data collection, it’s pre-processing, simulation platform, architectures of pre trained models, and then Result and Discussion concluded with conclusion and Future Scope.

II. MATERIAL AND METHODS

Data Collection

The data for this study was obtained from the National Institute of Health [23]. It's made up of segmented cells from the Malaria Screener research activity's thin blood smear slide photographs. At Chittagong Medical College Hospital in Bangladesh, Giemsa-stained thin blood smear slides from

ISSN: 2233-7853 IJFGCN

150 P. falciparum-infected and 50 healthy patients were collected and photographed. A professional slide reader at the Mahidol-Oxford Tropical Medicine Research Unit manually labelled the photographs. There are 25,977 segmented cell photos in the dataset, with

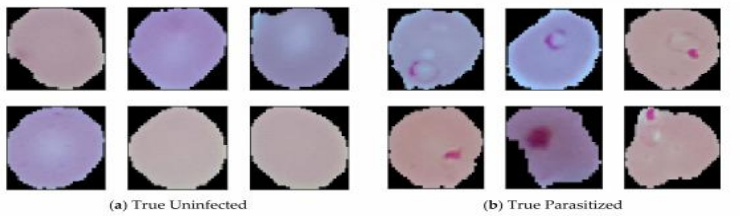

FIG 1: Parasitized and Uninfected cell mages

12,188 parasitized and 13,789 uninfected segmented red blood cell photos in equal numbers.
Positive samples had plasmodium, while negative samples did not have plasmodium but could have other things such as staining artefacts or contaminants. The patches of segmented red blood cells are 3-channels (RGB) with sizes ranging from 110 to 150 pixels

.

Image data pre-processing and Data Compilation:

To build a deep learning model data need to be trained and also is essential to test the performance of the model on unseen data. These models use 80:10:10 splitting to train the data, validate as well as test dataset correspondingly.
The Training set comprises of 20781 image samples while the Testing set has 5196 image samples.
The dataset consisted of image samples of irregular dimensions. In order to train our models we upsampled the dataset to 224x224x3,each axis in the dimension representing the height, width and the colour channel, which in our case is RGB, of the images respectively. To upscale all the images without having to lose relevant information we used geometric interpolation in image processing. The technique employed was bicubic interpolation to interpolate data points on a two dimensional grid. The images resampled using this technique were found to be smoother and having less interpolation artifacts.

Simulation Platform

The models were trained and tested on a, platform called Google's Colaboratory, a web based IDE for Python programming, which enabled us to utilize high end GPUs for free. Colaboratory comes with excellent cloud support by Google's cloud platform which allowed us to use upto 15 gigabytes of memory. Colaboratory is embedded with 3 runtime options users can choose from, CPU, GPU and TPU. For the proposed project, each model was trained on the Tesla T4 16GB GPU which is based on the Turing architecture and is solely targeted to enhance deep learning model inference acceleration.
PythonR 3.6.9, KerasR 2.1.6 with TensorflowR 1.4.0 backend.

The Network Architectures

1. CNN -

Convolutional neural networks are one of the many neural networks out there that processes image data using relevant filters, The layer is the basic data structure of any neural network, In our case the Convolutional layers which perform mathematical filtering on the input image tensors and return an

ISSN: 2233-7853 IJFGCN

output. A Convolutional network proves to be more efficient than primitive methods when it comes to pre-processing, because in primitive classification algorithms the filters are manually generated whereas Convolutional networks have the ability to learn these filters based on the input.Convolutional networks can capture spatial and temporal dependencies in an image. Which is why convolutional networks are preferred over feed forward networks.[31]There are certain parameters of Convolutional neural networks that are discussed in detail below

a) **Convolutional layers: -** Are the basic building blocks of any CNN, It performs the linear function of convolution on input image data. This starts activation, on performing this process over and over again produces a feature map which is derived from the input image itself.

b) **Feature Maps: -** The feature maps of a CNN capture the result of applying the filters to an input image. They are the output obtained at each layer. The feature maps extracted depend upon the amount of data fed into the network. In our case we use around 20,000 images, each image of 224x224x3 dimension. The amount of features extracted would be huge. Therefore in order to reduce the dimensions of these feature maps, we employ special layers called Pooling layers

c) **Pooling layers: -** Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

d) **Stride: -** For any filter in the Convolutional neural network, after specifying the size of the filter, we specify the stride. It controls the filter which convolves around the given input image. For various neural networks, there are either present strides or can be implicitly set. For instance, in our case the pre-trained model VGG16 uses a maxpool layer which has a filter size of 2x2 and a stride of 2 and a Convolutional layer with a stride of 1.

e) **Activation functions: -** Activation functions act as deciding factors. They decide whether a neuron or node in our neural network be activated or not. by calculating the weighted sum and adding bias with it. The main purpose of any activation function is to add non-linearity into the output of any neuron.

f) **Dense Layers/Fully-connected layers**: - Fully Connected layers in a neural network are those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are a full connected layer which compiles the data extracted by previous layers to form the final output.

2.) Architecture of Pre-trained models –

ISSN: 2233-7853 IJFGCN

We have discussed the basic Convolutional neural network in the section above. In our case we have employed a popular technique in Deep learning called "Transfer learning". In deep learning, transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved. There are various models put forth by leading institutions in the world, in our case we have chosen 3 prominent models namely,

1) Oxford's VGG16
2) Oxford's VGG19
3) Microsoft's ResNet50

As described in the previous section, we used three different experimental settings and leveraged transfer learning by fine-tuning three pre-trained models, VGG16, VGG19, and ResNet50 architecture. The model architecture and weights for these mentioned CNN pre trained models were downloaded from GitHub repositories [28]. We'll go over each of these in detail.

CNN	VGG - 16	VGG - 19	ResNet - 50
Year Proposed	2014	2014	2015
Number of Layers	16	19	50
Top 5 errors on ILSVRC	9.33%	9.9%	6.71%
Number of Convolutional Layers	13	16	48
Kernel Size	3 , 3	3 , 3	7 , 7
Number of Parameters	15,242,050	20,551,746	24,637,826
Dropout	No	No	No
Batch Normalization	No	No	Yes

TABLE 1: Comparison of VGG - 16, VGG - 19 and ResNet - 50.
ILSVRC stands for "ImageNet Large Scale Visual Recognition Competition".

We will discuss the architecture of each one in detail below.

1) Oxford's VGG16 -

VGG16 is a convolutional neural network model proposed by [24] K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet,which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014.The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers.

ISSN: 2233-7853 IJFGCN

Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2. Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

2) Oxford's VGG19 –

A fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3). The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set. Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image. spatial padding was used to preserve the spatial resolution of the image. max pooling was performed over a 2 * 2 pixel windows with sride 2. this was followed by Rectified linear unit(ReLu) to introduce non-linearity to make the model classify better and to improve computational time as the previous models used tanh or sigmoid functions this proved much better than those. implemented three fully connected layers from which first two were of size 4096 and after that a layer with 1000 channels for 1000-way ILSVRC classification and the final layer is a softmax function

.

3) Microsoft's ResNet –

ResNet [50], short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients.AlexNet, the winner of ImageNet 2012 and the model that apparently kick started the focus on deep learning had only 8 convolutional layers, the VGG network had 19 and Inception or GoogleNet had 22 layers and ResNet 152 had 152 layers. In this blog we will code a ResNet-50 that is a smaller version of ResNet 152 and frequently used as a starting point for transfer learning. ResNet first introduced the concept of skip connection. The diagram below illustrates skip connection. The figure on the left is stacking convolution layers together one after the other. On the right we still stack convolution layers as before but we now also add the original input to the output of the convolution block. This is called skip connection. The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

III. **RESULTS AND DISCUSSION**

All three above mentioned pre trained CNN models were trained and tested using the below given Experiment setting TABLE- 2.

ISSN: 2233-7853 IJFGCN

Model	Input Size	Activation function	Optimizer	Batch Size/Epoch/Dropout	Learning rate
VGG– 16	224x224	Relu & Softmax	SGD	32/74/0.5	1e-5
VGG– 19	224x224	Relu & Softmax	SGD	32/100/0.5	1e-5
ResNet- 50	224x224	Relu & Softmax	SGD	32/100/0.5	1e-5

TABLE 2: Experimental setting of CNN models for Training and Testing

As the above table suggests, we were able to achieve good results in the form of fast converging objective function. FIG 2. Although we had to fine-tuned the pre-trained models. The top most layers were frozen and were added with relevant layers such as dropout, to avoid Overfitting of the model. Freezing the layers reduces the number of trainable weights which results in decreased model computational time and complexity. VGGnets have over 138 million parameters and ResNet50 has over 11 million parameters, which makes ResNet model comparatively computationally inexpensive.
The optimizer used was [26]SGD + momentum for all the three models, out of a plethora of optimizer choices,the traditional SGD + momentum proves to be the most efficient. Adaptive optimizers such as RMSprop,Adaptive Moment estimator(Adam) would often result in slow convergence because of the size of the parameters present in the model. The loss function employed was categorical cross entropy because our input tensors were one-hot-encoded before feeding into the network.
For dropout [25], 0.5 was set so half the inputs to the dropout layer were initialised to 0 to overcome the problem of over fitting. Batch-size was specified to be 32, which ensured the model's stability and speed. In addition, the activation function includes ReLU which proves to be fast and helped in avoiding overfitting of the model , and the Softmax function which creates a probability distribution of the classes.
This training set was used to train all three neural networks, and the trained models were then tested on the testing set. Validation was carried out after training, with the loss of each method given in FIG 2.

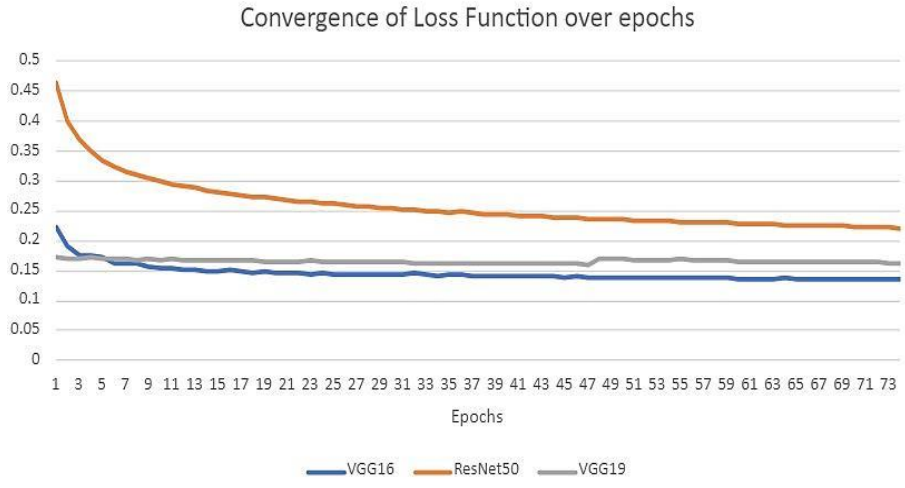

FIG 2: Validation loss for VGG-16, VGG-19 and ResNet-50.

ISSN: 2233-7853 IJFGCN

The above figure FIG 2 shows the convergence of Loss Function over number of epochs run for the above mentioned three models. It was observed that VGG-16 proved to be the most efficient model among the three in terms of convergence and ResNet-50 was observed to be the slowest for our input.

Model	Accuracy	Precision	Recall	F-1 score	AUC	Avg. precision score	Training Time	Log Loss
VGG- 16	95.55%	0.95	0.95	0.95	98.65%	98.655%	104min 83 s	0.1395
VGG- 19	94.59%	0.94	0.94	0.94	98.40%	98.434%	177min 6s	0.1551
ResNet- 50	91.48%	0.93	0.93	0.93	97.78%	97.809%	105min 20s	0.1960

Table 3: Performance Metrics

Table 3 shows the performance metrics that were obtained from the Sklearn's classification report library. These values were obtained by using the predict method on the test set consisting of 5196 images divided into two classes. Accuracy of the models was noted to be 95.55%, 94.59% and 91.48% for VGG16, VGG19 and ResNet50 respectively. Accuracy is simply the ratio of predicted to actual values.

Accuracy = (TN + TP) / (TN + TP + FN + FP)

In our comparative analysis, on the basis of performance evaluation metrics consisting of Precision score, Recall and F-1 score [27], VGG16 outperformed the other two models with observed trends of 0.95as an average for each metric. The model with the lowest metric score was ResNet with 0.93 as an average for all the metrics The precision is the ratio of all positive samples that are actually positive.

Precision = (TP) / (FP + TP)

Similarly, the recall is the ratio of positive predictions among all positive

Recall = (TP) / (FN + TP)

And, the F1 measure is a metric employed to describe the classification performance of the system. It is calculated through the recall and precision rate

F-1measure = (2TP) / (2TP + FP + FN)

In the comparative study we also observed that the highest training time was taken by VGG-19, 177min 6sec, on the other hand VGG-16 and ResNet-50 took much lower training time than the earlier. These results were obtained as training time is basically proportional to number of trainable parameters.

Also the best AUC and log loss was observed and obtained from VGG-16, 98.655% and 0.1395 respectively. Whereas the least results were observed from ResNet-50, 97.809 and 0.1960 respectively.

Here, the log-loss represents how close the forecast probability is to the actual/true number (0 or 1 in case of binary classification). The higher the log-loss number, the more the anticipated probability differs from the actual value.

ISSN: 2233-7853 IJFGCN

Furthermore, we plotted something called a binary confusion matrix[29] using a function. True positives, true negatives, false positives, and false negatives will all be listed in a confusion matrix. It informs us not only about the errors made by a classifier, but also about the kind of errors made while identifying Parasitized and uninfected malaria cell. The graphic in TABLE 4 depicts detailed comparison of confusion matrix of VGG16, VGG-19, and ResNet-50.

Positive in the confusion matrix is Uninfected blood cell and Negative is Parasitized blood cells. By this annotation we made a comparative matrix where True Positive [TP] is truly uninfected similarly True Negative [TN] is truly Negative, whereas False Positive [FP] is falsely predicted as Uninfected and similarly False Negative [FN] is falsely predicted as Parasitized which were observed in our comparative study and are given in TABLE 4.

ISSN: 2233-7853 IJFGCN

In addition to this TABLE 4 shows the confusion matrix for all three models

(Predicted Label)

(True Label)

		Positive	Negative	Accuracy
VGG- 16	Positive	2726	82	95.55%
	Negative	173	2215	
VGG- 19	Positive	2707	101	94.59%
	Negative	193	2195	
ResNet- 50	Positive	2649	159	91.48%
	Negative	220	2168	

TABLE 4: Comparison table of Confusion matrix of VGG16, VGG19 and ResNet50

From the confusion matrix [29] it is evident that the number of false positives and false negatives is highest in the model ResNet-50 with 220 and 159 images respectively, that is 220 falsely predicted as Uninfected out of 2388 parasitized truly labelled images, which comes to 9.2% parasitized images were classified to Uninfected. Similarly, 159 falsely predicted as Parasitized out of 2808 uninfected truly labelled images, which then comes to 5.6% uninfected images were classified to Parasitized. And here the best performed model was VGG-16 with highest accuracy and least false positive and false negative predicted label images TABLE 4.

IV. CONCLUSION AND FUTURE SCOPE

This study aimed to apply a deep learning pre trained model for the detection of parasitized and uninfected malaria cell and compare them on various performance evaluation metrics.Malaria causes death of lot of people every year, and it's especially dangerous for children and people with chronic illness. The proposed work aimed at providing a solution to the question, “which model would perform better on my dataset?”. Based on the results obtained it was clear that Oxford’s VGGnets performed better as feature extractors for dataset, unlike Microsoft’s Residual network. Although the ResNet was the first model that introduced the concept of skip-connection. The idea of skip connection revolutionized convolutional architectures. If the current problem were to be solved by anyone in the year 2014, they would’ve stumbled upon the infamous vanishing gradient problem. In residual networks, The skip blocks are stacked together the fundamental idea one should keep in mind here is that skip connection is used to preserve the gradient value that would often get very small even zero at times, for the first few layers while backpropogating using the chain rule. Techniques such as skip-connection pave the path for the future methods of deep learning. The VGGnets despite of being complex models with a large number of parameters are used more often because of the ease with which one can implement them.

This study has one major limitation which is the unavailability of the recommended hardware, the computationally expensive process that is deep learning often requires state of the art hardware. The ResNet model was first run on a personal computer that had an NVIDIA GTX 1650 GPU, yet the training time of the model for 10 epochs was noted to be 8 hours. It is highly recommended to use cloud based instances that are compatible with the deep learning frameworks or employ techniques such as parallel processing for utilising one or more CPU cores on your system to perform the task at

ISSN: 2233-7853 IJFGCN

hand. There are many cloud based platforms that provide virtual instances which have fully fledged high end GPUs such as AWS's p2.xlarge instance or the EC2 instance that provides popular deep learning frameworks for us to explore. One can also opt for platforms such as FloydHub and Kaggle.

Through this comparative study we will be able to extract vital information that discriminates a model from the other. In future works, the network architectures will be further analysed to understand the various performance metrics yielded on the basis of series of experiments which include computing Performance metrics by using different optimizers, activation functions and also finding optimal layers [31] which contain the best trained weights of the model and storing those weights which can be efficiently accessed for transfer-learning applications.

REFERENCES

1. World Health Organization, "Disease burden of malaria," http://www. who.int/mediacentre/factsheets/fs094/en/.
2. D. K. Das, et al. , "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Journal of Micron*, vol. 45, pp. 97–106, Feb 2013.
3. F. B. Tek, A. G. Dempster and I. Kale, "Parasite detection and identification for automated thin blood film malaria diagnosis," *Journal of Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 21– 32, January 2010.
4. N. E. Ross, et al., "Automated image processing method for the diagnosis and classification of malaria on thin blood smears," *Medical and Biological Engineering and Computing*, vol. 44, no. 5, pp. 427– 436, May 2005.
5. F. B. Tek, "Computerised diagnosis of malaria," Ph.D thesis, Univer- sity of Westminster, 2007.
6. V. Muralidharan, Y. Dong, and W. D. Pan, "A comparison of feature selection methods for machine learning based automatic malarial cell recognition in wholeslide images," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Feb 2016, pp. 216–219.
7. J. A. Quinn, R. Nakasi, P. K. Mugagga, P. Byanyima, W. Lubega, and A. Andama, "Deep convolutional neural networks for microscopy- based point of care diagnostics," *arXiv preprint arXiv:1608.02989*, 2016.
8. 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.
9. 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.
10. 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.
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov,
12. 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.
13. D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.
14. "Visualizing the images and annotations," http://grand-challenge.org/ site/camelyon16/data/.
15. "Whole slide image for malaria infected red blood cells," http:// peir-

ISSN: 2233-7853 IJFGCN

vm.path.uab.edu/debug.php?slide=IPLab11Malaria.
16. University of Alabama at Birmingham, "PEIR-VM," available: http://peir-vm.path.uab.edu/about.php.
17. R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
18. Univ. of Alabama in Huntsville, "Link to the dataset used," http:// www.ece.uah.edu/ ~dwpan/malaria dataset/.
19. V. P. Plagianakos and G. D. Magoulas, "Stochastic gradient descent," *Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873–1950)*, vol. 54, p. 433, 2013.
20. M.-K. Hu, "Visual pattern recognition by moment invariants," *infor- mation Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.
21. [N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and
22. R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
23. WHO. 2016. World malaria report. Available at http:// apps.who.int/ iris/ bitstream/ 10665/ 252038/ 1/ 9789241511711-eng.pdf?ua=1 (accessed on 4 January 2017)
24. Simonyan K, Zisserman A. 2015. Very deep convolutional networks for large-scale image recognition. ArXiv preprint. arXiv:1409.1556.
25. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15:1929_1958.
26. LeCun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86:2278_2323 DOI 10.1109/5.726791.
27. Lipton ZC, Elkan C, Naryanaswamy B. 2014. Optimal thresholding of classifiers to maximize F1 measure. Machine Learning and Knowledge Discovery in Databases 8725:225_239 DOI 10.1007/978-3-662-44851-9_15.
28. Chollet F. 2017. Deep learning models. Available at https:// github.com/ fchollet/ deep-learning-models (accessed on 2 February 2017).
29. Chicco D. 2017. Ten quick tips for machine learning in computational biology. BioData Mining 10 DOI 10.1186/s13040-017-0155-3.
30. He K, Zhang X, Ren S, Sun J. 2016. Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), 770_778 DOI 10.1109/CVPR.2016.90.
31. Rajaraman et al. (2018), Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ 6:e4568; DOI 10.7717/peerj.4568

ISSN: 2233-7853 IJFGCN