Visualizing Deep Learning Activations for Improved Malaria Cell Classification

Sivarama Krishnan R
National Library of Medicine
National Institutes of Health,
Bethesda, MD, USA
rajaramans2@mail.nih.gov

Sameer Antani
National Library of Medicine
National Institutes of Health,
Bethesda, MD, USA
sameer.antani@nih.gov

Stefan Jaeger
National Library of Medicine
National Institutes of Health,
Bethesda, MD, USA
stefan.jaeger@nih.gov

ABSTRACT

Malaria is a life-threatening disease caused by the parasites transmitted through the bite of the female Anopheles mosquito. Thick and thin film microscopic examinations of blood smears are the most commonly used and reliable methods for diagnosis, however, its accuracy depends on the smear quality and human expertise in classifying the normal and parasitemic cells. Manual examination can be burdensome for large-scale diagnoses in endemic regions resulting in poor quality, unnecessary medication, leading to severe economic impact to the individual health program. Automated malaria screening using machine learning techniques, such as deep learning, offers the promise of serving as an effective diagnostic aid. In this study, we propose the advantages offered through visualizing the features and activations in a simple, customized deep learning model. We apply it to the challenge of malaria cell classification, and as a result the model achieves 98.61% classification accuracy with lower model complexity and computation time. It is found to considerably outperform the state of the art including other pre-trained deep learning models.

Keywords

Deep learning, machine learning, malaria, classification, pre-trained models, visualizing activations, deep networks

1. INTRODUCTION

Malaria is a life-threatening disease caused by the parasites transmitted through the bite of the female Anopheles mosquito. Different kinds of malaria parasite including *P. falciparum*, *P. ovale*, *P. vivax* and *P. malariae* can infect humans, of which *P. falciparum* is the deadliest. According to the 2016 World Health Organization (WHO) report, there were 212 million instances of the disease worldwide and the African region accounted for the majority of the disease cases, followed by south-east Asian and eastern Mediterranean regions [26]. Microscopic examination of thick and thin blood smears for infected erythrocytes is a commonly used method for malaria diagnosis. Depending on the local protocol, the examination includes: (i) classifying and counting the normal and infected erythrocytes in the thin smear images; and/or (ii) counting parasites in thick smear images as specified in the WHO guidelines [3]. Thus, the diagnostic accuracy is heavily dependent on manual expertise and can be adversely impacted by the burden posed by large scale analyses that are common in malaria endemic regions. Alternative techniques such as polymerase chain reaction (PCR) and rapid diagnostic tests (RDT) are also widely used. However, PCR tests are limited in their performance [12] while RDTs are less cost-effective in zones with high disease prevalence [10].

Machine learning (ML) techniques have been previously applied to detect the degree of parasitemia from Giemsa-stained blood smear images [24, 18]. The study focused on developing a software to automatically compute the degree of parasitemia using hand-selected features from the stained blood smears containing *P. falciparum* parasites. However, the study did not explicitly classify normal and uninfected blood cells and the manual selection of features demanded human intervention. Another study to perform expert-level malaria diagnosis based on automated microscopy was proposed where a number of classifiers including a linear support vector machine (SVM), radial basis function SVM, multi-layer perceptron (MLP) and Gradient-Boosted Decision Trees (GBM) were used to classify the normal and parasitemic cells [5]. However, the study involved only fewer samples and the performance suffered with increase in data size. Diaz et al. proposed a SVM based classification method for detecting the infected red blood cells (RBC) in preprocessed blood smear images [7]. The algorithm performed well on a smaller dataset of 450 images, however, the performance decreased when applied to the blood images carrying infected cells.

2. BACKGROUND

Dealing with input variances is one of the main concerns in classifying images, irrespective of its kind. It is difficult to account for the changes in size, background, angle and position of the objects inside images. In the process, state-of-art image processing algorithms rely on cleverly hand-engineered features for representing the underlying data. Extensive time is spend in this preprocessing step, demanding human expertise that severely limits the accuracy achievable by a training algorithm. Sufficient large training examples are needed to learn the appropriate invariances with minimal processing, just by using the low-level data representations such as raw pixels. Deep learning (DL), also known as deep machine learning (or hierarchical learning), is a class of ML algorithms that use a cascade of layers of non-linear processing units for end-to-end feature extraction and classification and are resilient to these variances [20]. DL using convolutional neural networks (CNN) has gained research interest because it offers the promise of delivering high quality classification without the need for hand selecting features. Unlike SVMs, the performance of DL models increases with the number of training examples, making them highly scalable [25].

Small datasets are not adequate to train a DL model which has a multitude of parameters that need tuning. Transfer

Learning (TL) methods are commonly used to alleviate the problem where a pre-trained deep network is used to extract the features that are subsequently used in a conventional classifier like SVM [17]. These pre-trained models have already learned features that are useful for most computer vision (CV) problems, and visualizing such features provides a better understanding of the learning process and allows reaching a comparable accuracy to that of a customized model. Krizhevsky et al. (2012) proposed the AlexNet model, trained on ImageNet data, containing over 15 million annotated images from a total of over 22,000 categories [15]. The model used rectified linear units (ReLU) for imposing nonlinearity and data augmentation techniques that consisted of image translations and reflections. It also used dropout layers to combat the problem of overfitting to the training data and was trained using batch stochastic gradient descent (SGD) with specific values for momentum and weight decay. Simonyan and Zisserman from the University of Oxford proposed a simple and deep model in 2014 dubbed VGGNet that used only 3 x 3 sized filters all through the model [23]. Several variants of the VGG networks were proposed including VGG16 and VGG19, where "16" and "19" indicate the number of weight layers in the network. These models reinforced the notion that the combination of two 3 x 3 convolution layers has an effective receptive field of 5 x 5 that simulates a larger filter while keeping the benefits of smaller filter sizes and parameters. The model performed equally well on image classification and localization tasks.

TL models reduce the training time at the cost of performance and may be suitable when larger training datasets are not available. They produce useful features so long as the domain under study does not deviate much from the data [6]. They tend to perform poorly on data on which they are not trained on before. Further, we are also constrained in terms of the network architecture. We can't selectively modify the pre-trained network [4]. The other big difference lies in the formulation of the problem. TL models were designed for multiclass classification which means that they learn a lot of additional information that may not be needed in a binary classification problem such as ours. The issue can be resolved by fine-tuning a pre-trained model augmented with a few layers for binary classification with more number of epochs. This changes the intra-network information from multi-class to a binary-class problem. It is necessary to visualize the features extracted by the DL model and their activations in order to better understand its learning strategy. The downside of such a practice is that the amount of weights stored internally can be huge, requiring additional regularization. Also, the performance of fine-tuned models relies on the initial pre-trained model and any improvement in performance is tied to the representation learned by the original model [21].

These challenges can be overcome by using a customized model, trained on the domain of interest. CNN based DL models gives promising results for perceptual applications like image classification [14]. A survey of literature revealed few comparable articles for malaria cell classification using DL models. A method based on CNN for classifying the parasitemic and uninfected cells from thin blood smear images was attempted that used 27,578 single cell images resulting in an average accuracy of 97.3% [16]. By comparison, a pre-trained model used for classifying the same data achieved 91.99% accuracy.

In this article, we propose to use the advantages offered through visualizing extracted features and DL network activations in a simple, customized DL model for malaria cell classification. Our goal is to not only improve the state of the art in malaria classification, but also to understand the impact of various learned parameters on DL models at different stages of the deep network. The remainder of this paper is organized as follows: Section 3 illustrates the materials and methods; Section 4 discusses the results; Section 5 gives the conclusion.

3. MATERIALS AND METHODS

3.1 Data Collection and Preprocessing

Red blood cells (RBCs) from Giemsa stained thin blood slides of images, obtained from the U.S. National Library of Medicine (IRB#12972) are used in this study. They were acquired from *P. falciparum* parasite infected and normal patients, in Chittagong Medical College hospital, Bangladesh. Cells were annotated as either parasitemic or normal, by an experienced professional slide reader. The visual region of the erythrocytes was segmented from the raw images by applying coupled edge profile active contours [8]. The dataset included 110,000 images of erythrocytes, with a 1:1 ratio of parasitemic and uninfected cells. Images were normalized to a median width and height of 32 x 32 pixels. Several instances of normal and parasitemic cells were chosen to study the performance of the customized model and other models used in this study. Images were normalized to have zero mean to assist faster convergence and whitened to reduce data redundancy so that the algorithms train with instances having independent feature variable with unitary covariance.

3.2 Configuring the customized Model

The architecture of the CNN model strongly influences its performance. In this work, a 13-layer CNN model is proposed for the binary task of classifying the normal and parasitemic cells, as shown in Figure 1. The proposed model consists of five convolutional layers including the fully-connected layers. The sandwich design with one convolutional layer and ReLU layer allow enhanced learning [22]. Additionally, 3 x 3 max-pooling layers with a stride length of 2 are used after the first and second ReLU layers. The final layer is a Softmax classifier that measures the prediction distribution between the two classes [2]. Proper initialization of weights is a key to network convergence. He et al. proposed a method for weight initializations for networks with ReLU non-linear activations [11]. It is a more sophisticated initialization than the regular method of weight initialization [9] that uses the Gaussian distribution; however, the Gaussian is rescaled according to the number of neurons connected to the input of a given layer. He's formula for weight initialization is given by:

$$He_{gain} = B * sqrt\left(\frac{2}{h * w * in}\right)$$

where 'B' is a constant scaling factor, 'h' and 'w' are the height and width of the kernel and 'in' denotes the number of neurons connected to the input of the layer. The filters and biases are rescaled to make the covariance unitary. Random samples from the normal and parasitemic datasets are presented to the model and the performance is evaluated against other pre-trained/fine-tuned models. The algorithms are coded in Matlab® R2017a and Keras wrapper with Python 2.7.5 and Tensorflow backend, in a Windows® system with NVIDIA GTX 1080 GPU and 24 GB internal random access memory (RAM). The convolutional part of the pre-trained models including AlexNet and VGG16 is instantiated. Everything up to the fully connected layers and the models are run on the training and validation data to record the last activation maps before the fully connected layers. A small fully connected model is trained on top of the stored features, and the performance of these models are compared with the customized model under study. The weights file for the pre-trained models are downloaded from Github repository [13].

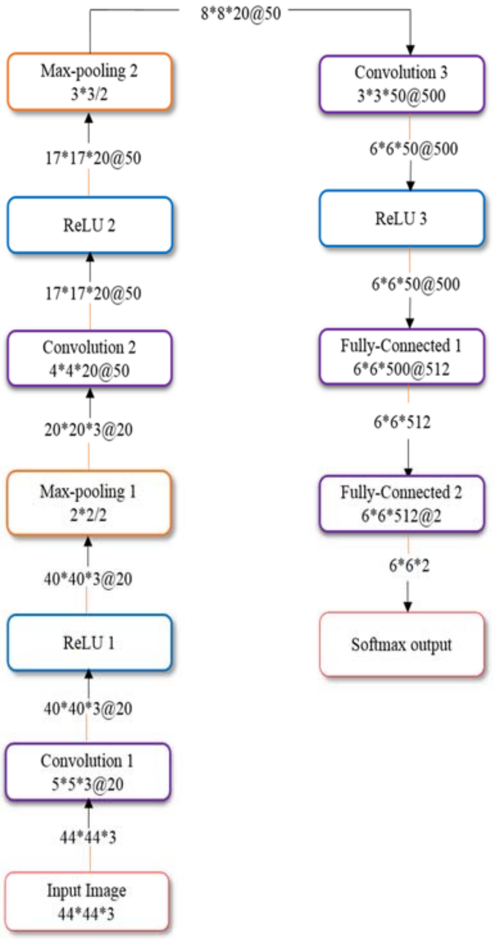

Figure 1: Customized CNN model.

3.3 Fine-tuning pre-trained models

The last convolutional block of the pre-trained models is fine-tuned alongside the top-level classifier. Fine-tuning process consists of starting from a trained network and re-training it using very small weight updates on the dataset under study [19]. Steps for fine-tuning include: (i) instantiate the convolutional base of the pre-trained models; (ii) load their weights; (iii) add the previously defined fully connected model on top; (iv) load their weights; and, (v) freeze the layers of the pre-trained models up to the last convolutional block. As the top-level classifier is trained, the convolutional weights are simultaneously fine-tuned with a very slow learning rate using the SGD optimizer to ensure the magnitude of the updates remains small. Finally, the validation accuracy is recorded.

4. RESULTS AND DISCUSSION

4.1 Visualizing the features

Analyzing the extracted features helps in understanding the impact of the learned parameters at various stages of the network. Study of literature reveals insufficient discussion on the impact of problem-specific features learned or the activations at different layers of the network. This study aims to analyze the features and activations learned by the proposed model that are specific to identifying the normal and parasitemic cells to aid in malaria screening. The features learned at different stages (layers) of the model are investigated to visualize the parameters learned from the training examples. This is done by generating images that strongly activate a particular channel of the network layers. The first convolution layer learns 20 features which are visualized in Fig. 2a. The images mostly contain colors and edges, indicating that the channels are color filters and edge detectors. This allows the proposed model to construct useful complex features in the deeper layers. The features on the second convolutional layer are crafted using the features from the first convolutional layer. The first 30 features learned by this layer are visualized as shown in Fig. 2b, where we observe that the model begins to learn high-level features including the shape and location of the parasites along with the color and texture information. The third convolutional layer learns the features by combining the low-level features from the first and second convolutional layers. It is observed that this layer, deeper into the network, yields detailed information on the shape, color and texture of the features as shown in Fig. 2c. An instance of the uninfected and parasitemic cells is shown in Fig.2d. The fully connected layer towards the end of the network learns high-level abstractions of the features learned by the earlier layers and outputs two channels corresponding to the normal and parasitemic cell classes respectively and are visualized as shown in Fig. 2e. A closer look into the features learned by this layer shows that they resemble the uninfected and parasitemic classes, respectively.

4.2 Visualizing the activations

The activations of the different layers of the network are visualized to understand the model learning strategy. The features learned by the network are evaluated by examining the activations and comparing them with the original image. A parasitemic cell image is read into the model for visualizing the activations at different layers of the network. The first convolutional layer performs convolutions with the input and the features are investigated by observing the areas where the layer activates on the input image and comparing them with the corresponding areas in the original image.

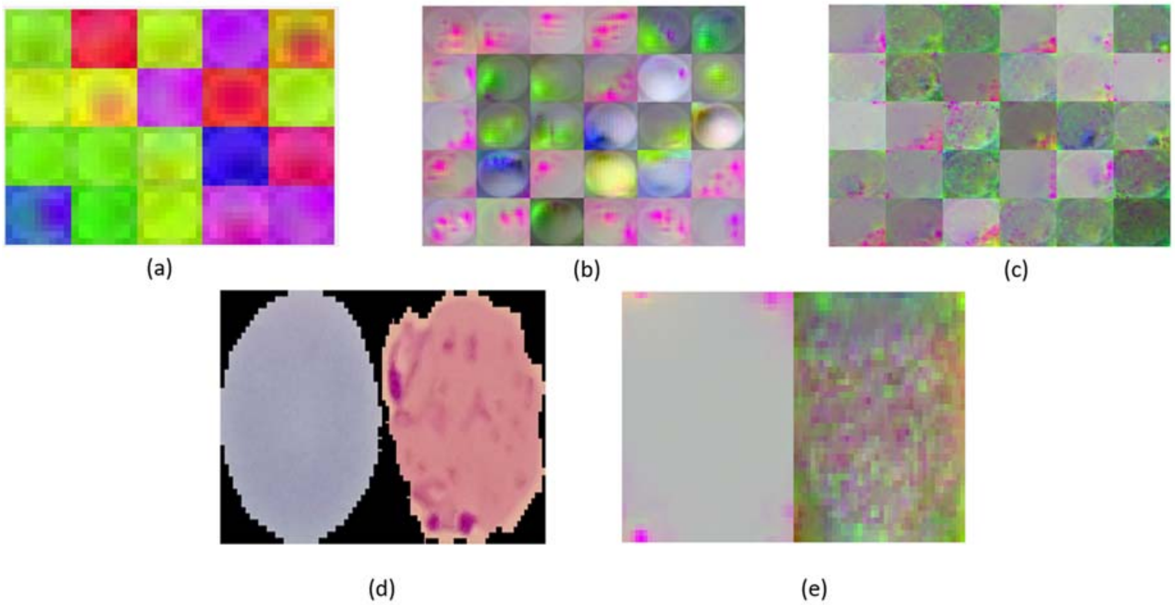

Figure 2: Visualizing filters at (a) 1st convolution layer (b) 2nd convolution layer (c) 3rd convolution layer (d) an instance of uninfected and parasitemic cells (e) fully connected layer 2.

The activations are returned as a three-dimensional array where the third dimension represents the number of channels in a given layer. A montage for activations in each layer is shown in Fig. 3a, one for each channel in the layer. The output channel in the first convolutional layer is displayed as squares in the montage of activations. Strong positive activations are represented by white pixels and negative activations, by black pixels. A gray channel does not activate as strongly on the original input. The position of a pixel in a given activation corresponds to the same position in the original image, e.g., a white pixel at a given location in a channel activation indicates that the channel is strongly activated at that position. This is correlated with the original image to verify location of the parasites. Fig. 3b shows the original images and the activations for the 11th channel for an example image. The channel activations are resized to have the same size as the original image. The highest channel activation, identified as white pixels, corresponds to the location of the parasites in the original image. This cumbersome manual process of identifying interesting channels can be alleviated by programmatically investigating the channels for activations with large values. This automation results in a greater throughput in the analysis resulting in a better understanding of the learned parameters. The proposed model learns to detect features like colors and edges in its first convolution layer and more complex (and abstract) concepts in deeper layers. These deep layers extract this information by developing their features by combining the features from the early-stage layers. We demonstrate this by investigating the third convolutional layer in a manner similar to the first convolutional layer. The activations are displayed as a montage in Fig. 4a. In the montage of all channels, the 40th channel activates strongly on the location of the parasites. The channel contains both positive and negative activations but only positive activations are used because of the presence of ReLU non-linearity, following the convolutional layers.

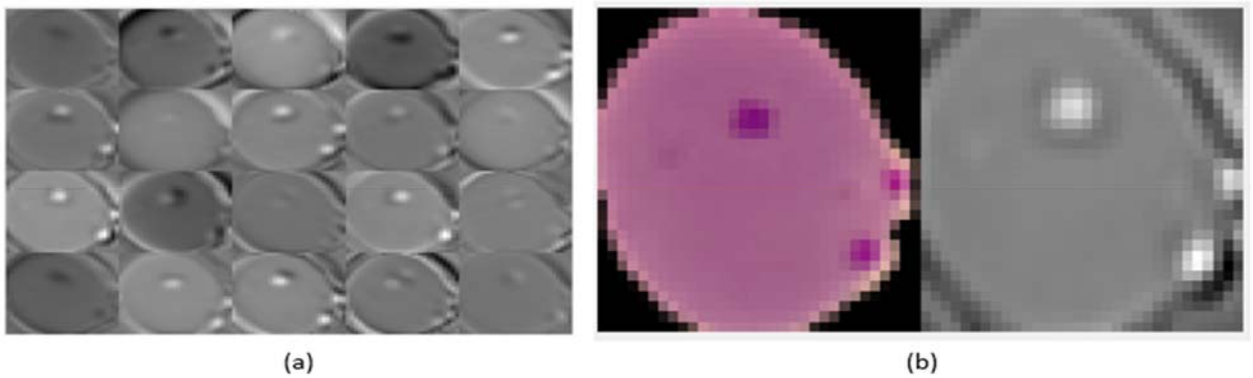

Figure 3: Visualizing activations: (a) 1st convolutional layer (b) comparing original image with 11th channel activation.

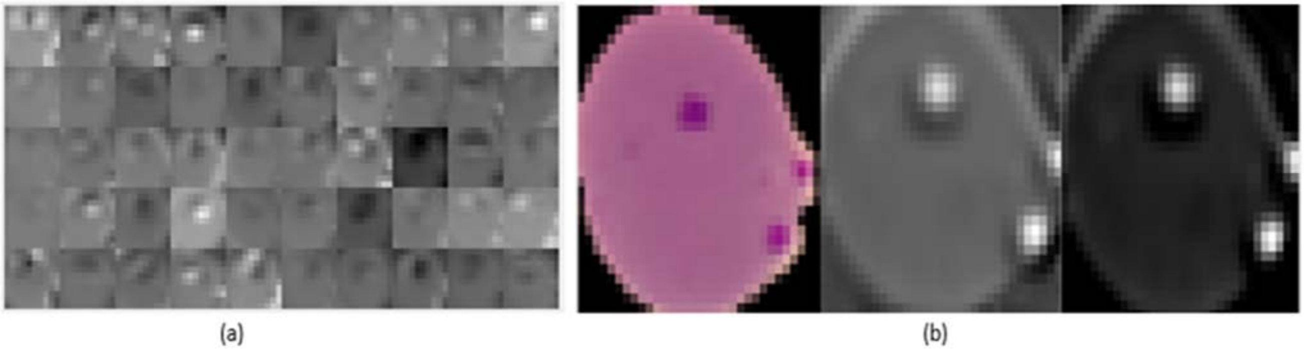

Figure 4: Visualizing activations of (a) 3rd convolutional layer (b) comparing the activations with the third ReLU layer.

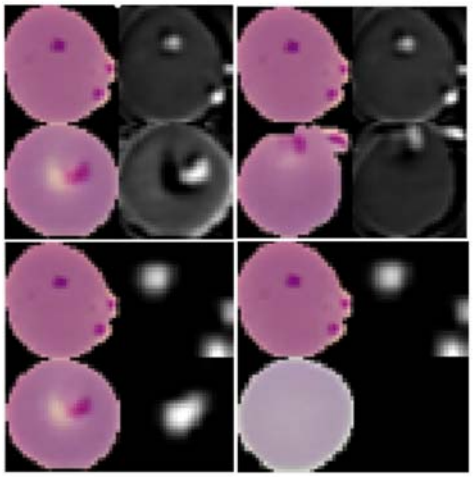

Figure 5: Channel activations at parasite locations.

The analysis is repeated to visualize the activations of the third ReLU layer to investigate the positive activations and is shown in Fig. 4b. The activations of the third ReLU layer clearly demonstrate areas of the image having strong parasitemic features in comparison to the activations of the third convolutional layer. The 40th channel of the third ReLU layer is then evaluated for getting activated on the location of the parasites. Both normal and parasitemic cells are input to the trained model and the resulting activations are compared with the activations of the original image as shown in Fig. 5. It is observed that the 40th channel activates only on the location of the parasites but not on the uninfected cell. The network has never been told to learn about the parasites, but it has discovered that the characteristics of the parasites are useful features to distinguish between the classes.

Conventional ML algorithms use handcrafted features specific to the problem, but the proposed CNN model learns useful features by itself. Learning to identify parasites helps the model to distinguish between a normal and parasitemic cell in malaria diagnosis. The customized model and the pre-trained models are optimized for the hyper-parameters, for the dataset used in this study, by a randomized grid search method [1]. The hyper-parameters used in this study for the different models are tabulated as shown in Table 1.

Model	Optimizer	Momentum	Learning Rate	Decay
Customized model	SGD	0.9	$1e^{-2}$	0.0
AlexNet	SGD	0.9	$1e^{-2}$	0.0
VGG16	SGD	0.9	$1e^{-2}$	0.0027

Table 1: Hyper-parameter optimization

The convolutional part of the pre-trained models including AlexNet and VGG16 is instantiated, everything up to the fully connected layers. The models are run on the training and validation data to record the last activation maps before the fully connected layers. The dataset is randomly split into 70% for training and 30% for validation. A small fully connected model is trained on top of the stored features and the performance is compared with that of the customized model for various sample sizes. The validation accuracy, sensitivity and specificity of the customized model and that of pre-trained models are graphically shown in Figure 6. As observed, the customized model outperforms the pre-trained models in terms of computational accuracy.

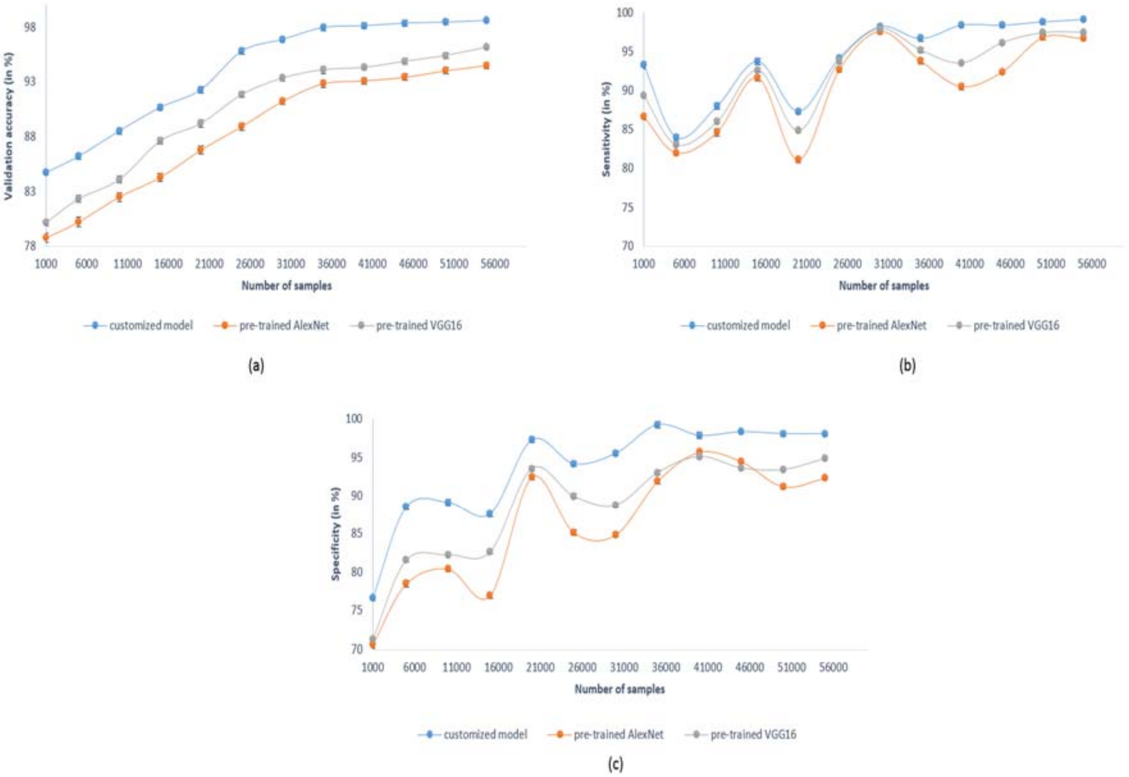

Figure 6: Performance comparison of the customized model and pre-trained models.

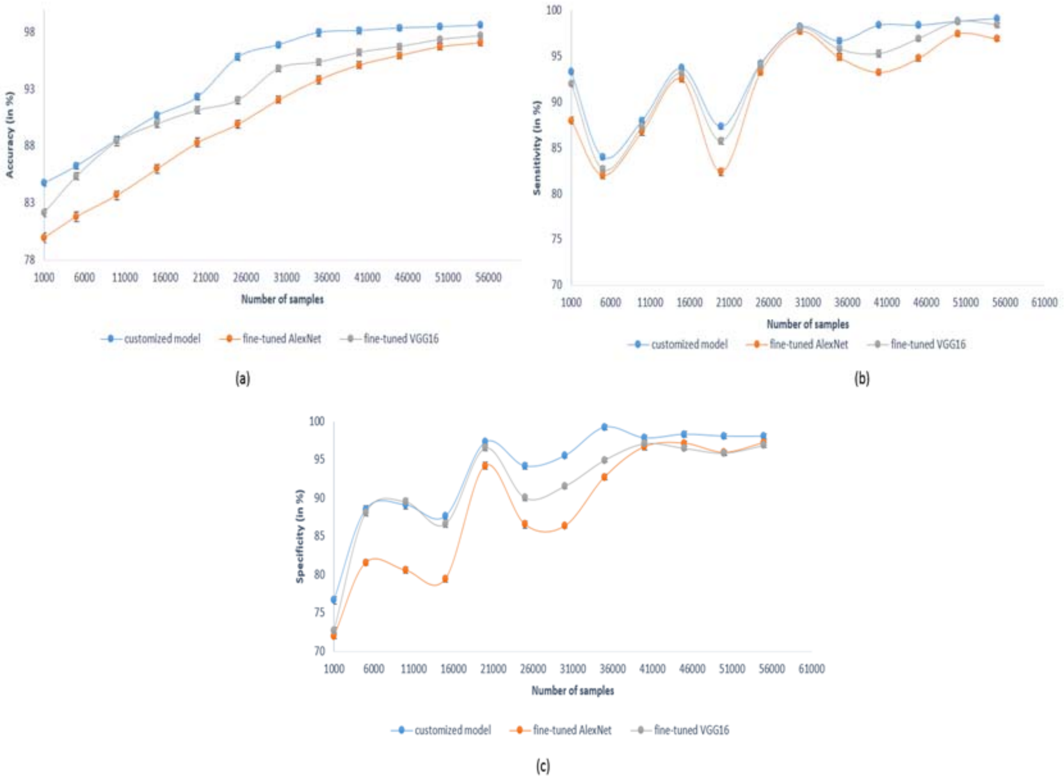

Figure 7: Performance comparison of the customized model and fine-tuned models.

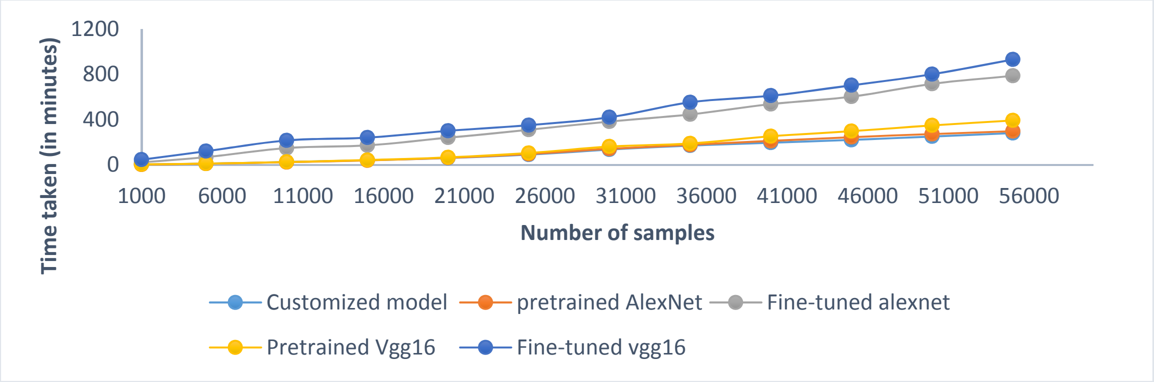

Figure 8: Comparison of computation time

The model does reasonably well, irrespective of the number of samples which implies that the performance is due to the allocation of capacity to the important features and the specificity of the outputs to the task. The pre-trained models are further fine-tuned to evaluate if an additional increase in the computational accuracy is observed. Fine-tuning is done with a very slow learning rate of 1e^{-4} for 1000 epochs with the SGD optimizer to ensure the magnitude of the updates remains small. The validation accuracy, sensitivity and specificity are recorded and graphically shown in Figure 7. As observed, the fine-tuned models performed below par in comparison to the customized model. The customized model is compared to the pre-trained and fine-tuned models in terms of computation time and the results are shown in Figure 8. The customized model is noticeably faster than the pre-trained and fine-tuned models. Fine-tuned models are far more expensive in terms of computation time for the reason that the customized model devotes its entire capacity to the relevant features of the problem under study whereas the pre-trained and fine-tuned models devote their capacity to a huge number of irrelevant features and hence take a long time to retrain everything and get to a useful result.

5. CONCLUSION

The study reveals that, unlike pre-trained/fine-tuned models, a simple and customized CNN based DL model is considerably the best solution for task-specific classification like classifying the normal and parasitemic cells as an effective aid in malaria screening. DL models offer the promise of serving as an effective diagnostic aid where manual examination can be burdensome for large-scale diagnoses. The model understands a probabilistic mapping between the combinations of filters and the given set of labels. The filters in each layer form the basis vectors that are used to encode the layer's input in a compact way. In comparison to the pre-trained/fine-tuned models, the customized model learns task-specific features for the problem under study and exhibits a superior performance with lesser model complexity and computation time. The proposed model can be adopted to significantly improve the accuracy of screening for other health-related applications. Next steps in our work aim to expand our analysis of pre-trained networks for various image types.

6. ACKNOWLEDGEMENTS

This work is supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and Lister Hill National Center for Biomedical Communications (LHNCBC).

7. REFERENCES

[1] Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13 (February 2012), 281–305

[2] Boser, B.E., Guyon, I.M. and Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of COLT’92 (Pittsburg PA, July 1992), ACM Press, 144–152.

[3] Centers for disease control and prevention (CDC). https://www.cdc.gov/malaria/diagnosis_treatment/clinicians1.html

[4] Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/transfer-learning/

[5] Delahunt, C.B., Mehanian, C., Hu, L., McGuire, S.K., Champlin, C.R., Horning, M.P., Wilson, B.K. and Thompson, C.M. Automated Microscopy and Machine Learning for Expert-Level Malaria Field Diagnosis. In Proceedings of GHTC’15 (Seattle WA, October 2015), IEEE Press, 393–399.

[6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of CVPR’09 (Miami FL, June 2009), IEEE Press, 248–255.

[7] Díaz, G., González, F.A. and Romero, E. A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. Journal of Biomedical Informatics, 42 (April 2009), 296–307.

[8] Ersoy, I., Bunyak, F., Higgins, J.M. and Palaniappan, K. Coupled edge profile active contours for red blood cell flow analysis. In Proceedings of ISBI’12 (Barcelona SP, May 2012), 748–751.

[9] Goodfellow, I., Bengio, Y. and Courville, A. Deep Learning. MIT Press, Massachusetts MA, 2016.

[10] Hawkes, M., Katsuva, J.P., Masumbuko, C.K. Use and limitations of malaria rapid diagnostic testing by community health workers in war-torn Democratic

Republic of Congo. Malaria Journal, 8 (December 2009), 308.

[11] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[12] Hommelsheim, C.M., Frantzeskakis, L., Huang, M., Ulker. B. PCR amplification of repetitive DNA: a limitation to genome editing technologies and many other applications. Scientific Reports, 4 (May 2014), 5052.

[13] Keras. https://github.com/fchollet/keras

[14] Keras blog. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[15] Krizhevsky, A., Sutskever, I. and Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of NIPS'12 (Lake Tahoe NA, December 2012), 1097–1105.

[16] Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M., Antani, S.K., Maude, R.J., Huang, J.X., Jaeger, S. and Thoma, G,R. (eds.). CNN-Based Image Analysis for Malaria Diagnosis. In Proceedings of BIBM'16 (Shenzhen CN, December 2016), 493–496.

[17] Mathworks.http://www.mathworks.com/discovery/dee-p-learning.html.

[18] Poostchi, M., Ersoy, I., Bansal, A., Palaniappan, K., Antani, S., Jaeger, S. and Thoma, G. Image analysis of blood slides for automatic malaria diagnosis. In proceedings of HI-POCT'15 (Bethesda MD, November 2015), MoPoster04.22.

[19] Roth, H., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K.M., Turkbey, E. Summers, R. Improving computer-aided detection using convolutional neural networks and random view aggregation. IEEE Transactions on Medical Imaging, 35 (May 2016), 1170–1181.

[20] Schmidhuber, J. Deep Learning in Neural Networks: An Overview. Neural Networks, 61(February 2015), 85–117.

[21] Schwenk, H. and Bengio, Y. Boosting neural networks. Neural Computation, 2 (August 2000), 1869–1887.

[22] Shang, W., Sohn, K., Almeida, D. and Lee, H. Understanding and Improving Convolutional Neural Networks. In Proceedings of ICML'16 (New York NY, June 2016), IMLS Press, 2217–2225.

[23] Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for large scale image recognition. arXiv preprint arXiv: 1409,1556, 2015.

[24] Sio, S.W., Sun, W., Kumar, S., Bin, W.Z., Tan, S.S., Ong, S.H., Kikuchi, H., Oshima, Y. and Tan, K.S. MalariaCount: an image analysis-based program for the accurate determination of parasitemia. Journal of microbiological methods, 68 (February 2007), 11–18.

[25] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15 (June 2014), 1929–1958.

[26] World Malaria Report 2016. http://www.who.int/malaria/publications/world_malaria_report/en/

About the authors:

Dr. Sivarama Krishnan is a postdoctoral fellow at the U.S. National Library of Medicine (NLM), part of the National Institutes of Health (NIH). His expertise includes applying Deep Learning for medical image analysis and understanding, medical image processing, machine learning and biomedical signal analysis. Prior to joining NLM, he was an Associate Professor at SSN College of Engineering, India. He earned his Ph.D. in Information and Communication engineering from Anna University, Chennai, India, and his B.Eng. in Electronics and Communication Engineering discipline from Madurai Kamaraj University, Madurai, India.

Dr. Sameer Antani is a Staff Scientist at NLM/NIH. He leads several scientific and technical research projects toward advancing the role of computational sciences in biomedical research, education, and clinical care. Dr. Antani is a Senior Member of the International Society of Photonics and Optics, Institute of Electrical and Electronics Engineers, and serves as Vice Chair for IEEE Technical Committee on Computational Life Scienes. He is Associate Editor for IEEE Journal of Biomedical and Health Informatics. Dr. Antani earned his Ph.D. and M.Eng. from Pennsylvania State University, USA, and his B.Eng. (with Distinction) from University of Pune, India.

Dr. Stefan Jaeger is a Research Fellow at NLM/NIH. He conducts research into image analysis and image informatics for clinical care and eduction. He leads a project team that develops computational screening methods for malaria and other diseases. Dr. Jaeger received his diploma in computer science from University of Kaiserslautern and his PhD from University of Freiburg, Germany. He is an editorial board member of Quantitative Imaging in Medicine and Surgery and associate editor of Electronic Letters on Computer Vision and Image Analysis.