World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:2, 2021

Malaria Parasite Detection Using Deep Learning Methods

Kaustubh Chakradeo, Michael Delves, Sofya Titarenko

Abstract—Malaria is a serious disease which affects hundreds of millions of people around the world, each year. If not treated in time, it can be fatal. Despite recent developments in malaria diagnostics, the microscopy method to detect malaria remains the most common. Unfortunately, the accuracy of microscopic diagnostics is dependent on the skill of the microscopist and limits the throughput of malaria diagnosis. With the development of Artificial Intelligence tools and Deep Learning techniques in particular, it is possible to lower the cost, while achieving an overall higher accuracy. In this paper, we present a VGG-based model and compare it with previously developed models for identifying infected cells. Our model surpasses most previously developed models in a range of the accuracy metrics. The model has an advantage of being constructed from a relatively small number of layers. This reduces the computer resources and computational time. Moreover, we test our model on two types of datasets and argue that the currently developed deep-learning-based methods cannot efficiently distinguish between infected and contaminated cells. A more precise study of suspicious regions is required.

Keywords—Malaria, deep learning, DL, convolution neural network, CNN, thin blood smears.

I. Introduction

MALARIA is a severe disease transmitted by mosquitoes. If not treated in time it can be fatal. According to the World Health Organisation (WHO) [1], in 2018 alone, there were 200 million diagnosed malaria cases worldwide, with the total amount of deaths being over 400,000. The WHO African Region is affected particularly badly, carrying 93% of malaria cases and 94% of malaria deaths. Children under 5 years are the most vulnerable. According to the most recent studies WHO suggests that due to the COVID-19 issue, malaria cases could be doubled in the oncoming year. This makes the investment in malaria research even more critical.

Three of the main methods for diagnosing malaria are microscopy, a rapid diagnostic test (RDT), and Polymerase chain reaction (PCR) (see [2]). Unfortunately, both RDT and PCR, have limitations (see [3], [4]). According to WHO microscopy is the most common tool for diagnosing malaria, though the accuracy can be poor. For example in [5], it is shown that while sensitivity is 99%, specificity is only 57%.

Recent advances in Artificial Intelligence (AI) allow to analyse samples more accurately and faster than a human eye would do. For example, in [6] a machine learning (ML) technique is suggested, which allows the overall accuracy to be $\geq 90\%$. In this study, we focus on microscope diagnostics, as it is the most common and cheap method.

ML methods, and Deep Learning (DL) in particular, are extensively used in medical image classifications (see [7], [8]). DL techniques have started to get integrated into medical equipment (see examples in [9] for identification of cancer cells and in [10]).

Usually, the methods include either segmentation tasks, feature extraction, or a combination of both of them. For example, in [10] a popular U-Net method was applied for cells segmentation. In [11] convolutional neural networks (CNN), such as ResNet and VGG16 were applied to mammograms and compared. In [12] a hybrid approach of segmentation and feature extraction was developed to identify breast cancer in mammograms.

Similar DL approaches are used to identify malaria parasites in blood cells from microscopy images. Two types of microscopy images are widely used for this purpose: thin blood smear images and thick blood smear images. Thick smears represent a thick layer of red blood cells and, therefore have a higher density of parasites. This makes thick smears particularly efficient for identifying the *presence* of malaria parasites in blood cells. Thin smears represent a thin layer of blood. They are normally used by clinicians to identify *stages* of malaria. The accuracy of detection in both tests depends on the quality of smears and level of human expertise [13]. It has been shown, however, that both thin and thick smears can be efficiently used to detect malaria by DL-based methods. For example, in [14]-[16] it is demonstrated that Convolutional Neural Network (CNN) based models can successfully extract features from thin smears towards classification of parasitised and uninfected cells. The development by [17] presented an ensemble model working with high accuracy on the same thin smears database. In [18] an autoencoder based model has been presented. Examples of the successful application of DL-based models to thick smear databases can be found in [19] and [20].

In this work we present a deep-learning based model to help malaria diagnostics. The model is based on a VGG-type customised neural network. Prior to the training process, extra work has been done on the image dataset. The details of these stages are discussed in Sections II and III-C. The model has been tested on two types of dataset (see Table I). Accuracy has been evaluated through a cross-validation strategy, both at patient and cell levels. Various accuracy metrics have been calculated and compared with a set of similar studies (see V). The model also has been profiled with VGG-16 and VGG-19 networks. Based on the experimental results we argue that a greater number of layers does not necessarily improve the

K. Chakradeo is with Radboud University, The Netherlands (e-mail: kaustubh.chakradeo@student.ru.nl).

M. Delves is with London School of Hygiene and Tropical Medicine, UK (e-mail: Michael.Delves@lshtm.ac.uk).

S. Titarenko is with School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK (e-mail: S.Titarenko@hud.ac.uk).

accuracy, if applied to thin smear images.

The tests prove that the model presented can be successfully used for detecting malaria from thin blood smear test images giving an accuracy of parasite identification up to 99.3%. Using standard accuracy metrics, it surpasses or is comparable to previous studies (see Table V). Moreover, by testing the model on two types of dataset we show the limitations of the currently developed DL-based model. The limitations and further plans for model improvements are discussed in Section V.

II. Data

For the current study we used images of Giemsa-stained thin blood smear slides which have been collected from 150 *P. falciparum* infected and 50 healthy patients at Chittagong Medical College Hospital, Bangladesh. The data were automatically segmented and manually annotated by an expert at the Mahidol-Oxford Tropical Medicine Research Unit in Bangkok, Thailand. They have later been published by the National Library of Medicine (NLM) and can be found in [21]. Cells containing *plasmodium* parasites are identified as positive (parasitised). Cells not containing *plasmodium* are identified as uninfected. Fig. 1 shows malaria cells from each class. In this study we label this type of dataset as **A**.

When developing and testing DL-based models we noticed that a relatively large proportion of the data are mislabeled. The data were, therefore, relabeled by one author. As a result 5 folders were produced: 1) a folder with images containing parasites (labeled "parasitised"); 2) a folder with uninfected images (labeled "uninfected"); a folder also includes uninfected images with impurities (see examples of such images in Fig. 1); 3) a folder with the images the expert was not sure about; 4) a folder with very badly segmented images; 5) a folder with "strange" images; the images with some artefacts of unknown origin. Only the folders "parasitised" and "uninfected" were used for this study. This reduced the number of "true parasitised" images to 12,058 and increased the number of "true uninfected" to 14,142 (if compared with the original dataset **A**). In this research we label this dataset **B**. The corresponding folders with the re-labeled images can be found in [22].

We also tested our model on the dataset obtained by [18] (labeled **C**). The authors of [18] have also noticed that the images in the original dataset [21] are mislabeled. The updated thin smears dataset can be found in [23]. Note, that the dataset originally published in [21] contains 27,558 cell images, from which 13,779 images are labelled as "parasitised" and 13,779 as "uninfected". After the cleaning process proposed in [18] the folders "true parasitised" and "true uninfected" were generated. The images with uninfected cells, but containing impurities or colouring have been removed from the study (see examples in Fig. 1). This reduced the number of "uninfected" images from 13,131 to 13,028. The procedure described makes the dataset "perfect" and does not allow a deep learning model to "learn" the difference between impurities and parasites. However, using this dataset for training and comparing with the results of training on dataset **A** is very important part of the study. It gives a better insight in the properties of the current deep learning model and shows its limitations.

In the this work we compare the accuracy of our model with previous studies. Since the accuracy of a deep learning model is affected by the dataset used for training/testing, we list here the datasets used in the comparative studies:

D. The dataset uploaded in Kaggle (see [24]). The dataset looks very similar to [21] and is likely to be the same. It has been used for deep learning modelling in [15].

E. The digital images from the open source MaMic Image Database from the Institute for Molecular Medicine Finland (FIMM) [25]. They were used in [20] along with a combination of blood smear samples collected locally.

F. Thick blood smear images collected by Chittagong Medical College Hospital, Bangladesh and manually annotated by an expert at the Mahidol-Oxford Tropical Medicine Research Unit in Bangkok, Thailand. The dataset can be found in [21]. It has been used in [19].

List of the datasets is presented in Table I

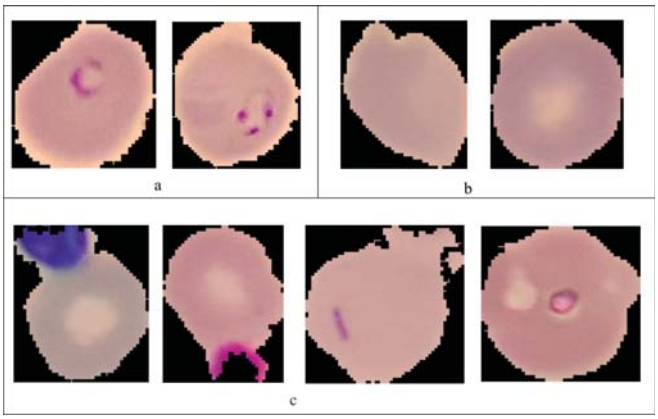

Fig. 1 Examples of (a) parasitised cell; (b) uninfected cell; (c) uninfected cells, containing impurities and artefacts

III. Materials and Methods

A. Cross-Validation Studies

Cross-validation studies are helpful in keeping the model from overfitting. This is because bias is removed from the training data. Variance is also significantly removed as most of the data is used in validation data as well [26]. Instead of splitting the data into training and testing, the data are split into groups. All groups but one are selected for training, and the remaining group is selected for testing. With each separate iteration, all groups are subsequently selected for testing one by one, which helps in reducing overfitting. For the model, the images were resized to 128×128 as input to train the CNN model. The dataset models were evaluated through the k fold cross-validation technique, with the number of folds being 5. Cross-validation has been performed at a patient level. We believe that this represents the most realistic evaluation of the method performance. This allows testing of how the model works on patients unknown for the trained system. The accuracy metrics are presented in Table V. To compare with

TABLE I
SUMMARY OF MALARIA DATASETS USED FOR DEEP LEARNING MODELS

Label	Source	Type	Comment
A	National Library of Medicine [21]	thin	Contains a number of mislabeled images;
B	Google drive[22]	thin	Thin smears from [21] have been re-labeled and cleaned; uninfected images with impurities are used in the study; **tested on the proposed model**;
C	Google drive [23]	thin	Thin smears from [21] have been re-labeled and cleaned; uninfected images with impurities removed from the study; **tested on the proposed model**;
D	Kaggle [24]	thin	The dataset is likely to be taken from [21];
E	Institute for Molecular Medicine [25]	thick	Tagore Medical College & Hospital blood smears + Mamic dataset;
F	National Library of Medicine [21]	thick	—

[15] and [14] we have also used the train-test split method with a 80:20 split for means of comparison of results. The datasets **B** and **C** prepared for 5 folds cross-validation study (at a patient level) can be found in [27].

The model was created and trained on an Ubuntu 18.04 system with Intel Core i5-9300H CPU @2.40GHz processor, 16 GB RAM, a CUDA enabled NVIDIA GeForce GTX 1660 Ti GPU with 6 GB memory. Python 3.8.6 with Keras 2.1.1 and Tensorflow 2.0.0 backend and CUDA 10.0 with cuDNN 7.1 library, used on Jupyter Notebook version 6.0.3.

B. Preprocessing

The images were resized to 128×128 size, with a RGB scale, as input to train the customised VGG8-based model (see Table II). When working on pretrained models they were resized to 224×224 as required. While resizing, as a part of the pre-processing step, only RGB channel images were allowed. The relabeled dataset contains 14141 infected and 12057 (for the dataset **B**) uninfected samples while doing the train:test split. By using Kerass ImageDataGenerator, data augmentation was done to increase generalizations in the dataset and to reduce overfitting. Data augmentation is a way of creating new data from the existing dataset with some changes. Data augmentation is useful for increasing the size of the dataset and introducing heterogeneity, thus having more images to work with [28]. It also helps in reducing overfitting, since there is an inclusion of lots of randomness in the dataset. Some of the data augmentation techniques used were rotation, shearing, zooming, horizontal flips, featurewise normalization, width and height shifts. By introducing shifts and rotations, we increased the degree of heterogeneity in the data. After splitting, preprocessing and augmentation, the training data contained $22{,}046$ images per class and testing data contained 5512 images. Table II shows the structure of the proposed model.

C. Model Description

The VGG type of neural network was first proposed by [29]. The main difference from the suggested earlier AlexNet is that AlexNet uses larger convolutional filters (11×11). The authors used small convolution filters (3×3 and 1×1); they also experimented with the number of layers and concluded that deeper structures can benefit classification accuracy. This type of deep neural network is recognised as efficient for image classification and is widely used in the ML community. We therefore chose VGG-based NN for our experiments.

The summary of the networks used for experiments is shown in Table II. Network VGG8 is a customized networks (highlight in bold).

TABLE II
VGG-BASED CONFIGURATIONS

VGG8	VGG16	VGG19
Input $128\times128\times3$		
conv3-64	conv3-64	conv3-64
conv3-64	conv3-64	conv3-64
maxpool		
conv3-128	conv3-128	conv3-128
conv3-128	conv3-128	conv3-128
maxpool		
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
	conv3-256	conv3-256
		conv3-256
maxpool		
	conv3-512	conv3-512
	conv3-512	conv3-512
	conv3-512	conv3-512
		conv3-512
maxpool		
	conv3-512	conv3-512
	conv3-512	conv3-512
	conv3-512	conv3-512
		conv3-512
maxpool		
FC-256	FC-4096	
FC-256	FC-4096	
	FC-1024	

Customised VGG8 CNN has six convolutional layers with two fully connected dense layers. The convolutional layers use a 1-pixel stride, with 3×3 filters. The first two convolution layers have input images of $128\times128\times3$ size and BGR channel. The first two convolution layers use 64 filters, the third and fourth convolution layers use 128 filters, the fifth and sixth convolution layers use 256 filters (see Table II).

Each convolution layer uses batch normalization. Using batch normalization standardizes the activations of each input variable per mini-batch, such as the activations of a node from the previous layer. Batch normalization is used to suppress

communication between mean and variance, allowing each layer to train separately, independently of the previous layers [30]. Using moving averages instead of taking snapshots at particular moments allows for keeping track of accuracy while the model trains. Batch normalisation which is applied is given by

$$\widehat{x}_i^k = \frac{x_i^k - \mathrm{E}[x]^k}{\sqrt{\mathrm{Var}[x]^k + \varepsilon}}, \quad k \in [1, d], \quad i \in [1, m] \tag{1}$$

where d is a dimension of an input layer $x = (x^1, x^2, \ldots x^n)$, m is a number of Batches, $\mathrm{E}[x]^k$ is a Batch mean.

$$\mathrm{E}[x]^k = \frac{1}{m}\sum_{i=1}^{m} x_i^k \tag{2}$$

and $\mathrm{Var}[x]^k$ is a Batch variance as seen in [31] is

$$\mathrm{Var}[x]^k = \frac{1}{m}\sum_{i=1}^{m}(x_i^k - \mathrm{E}[x]^k)^2. \tag{3}$$

In this research, we developed a deep learning model with Leaky ReLU as an activation function. Leaky ReLU is a customized version of parametric ReLU [32]. Leaky ReLU introduces a small slope for negative values, instead of making the slope zero, as normal ReLU does. Normal ReLU sometimes dies on finding a local minimum, LeakyReLU fixes this dying ReLU problem. In Leaky ReLU the alpha parameter denotes the negative slope coefficient. With lower values in the Fischer Information Matrix diagonal, as seen in [33], LeakyReLU also speeds up training. LeakyReLU also stabilizes training, with slope oscillations generally lying between the optimal and near optimal states as seen in Fig. 3. Leaky ReLU is defined by:

$$f(x) = \begin{cases} \alpha \cdot x, & \text{for} \quad x < 0, \\ x, & \text{for} \quad x \geq 0. \end{cases} \tag{4}$$

The value of α used is 0.1.

The output of the second convolutional layer was fed into the first pooling with dropout layer. The pooling layers have a 2×2 pooling window and summarize the convolutional output of neighbouring neuron groups in the feature maps by taking the maximum value from the 2×2 matrix. 20% of the neurons after the Max-Pooling layer output were randomly selected and dropped from the next weight update cycle in the Dropout layer to prevent overfitting.

The max-pooling layer was followed by two instances of two more convolution layers and max-pooling with dropout. The last max-pooling with dropout layer was further connected to the first fully connected dense layer. After converting the network matrix to a vector, a dense layer with all connected classes consisting of 256 neurons was created. Again, Batch normalization was used at this layer. The activation function used in this layer was also LeakyReLU with an alpha of 0.1. 30% of the nodes were dropped from the next weight update cycle as a part of optimization towards reducing overfitting. Immediately connected to this was the second fully connected dense layer. It used the same functions as the first fully connected dense layer, sans the flattening operation. This was connected to the final dense layer. At this layer, the activation function used was sigmoid. The sigmoid function is used for binary classifications. The sigmoid function is defined as

$$f(x) = \frac{1}{(1 + \mathrm{e}^{-x})}. \tag{5}$$

The architecture used for building the CNN led to a total of $17{,}994{,}561$ parameters, out of which $17{,}991{,}745$ were trainable and $2{,}816$ were non-trainable. The architecture is presented in Fig. II.

D. Feature Extraction and Optimization

Since it is a binary classification problem, we used the binary cross-entropy function [34]. It is given by the formula

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)), \tag{6}$$

where y is the label (parasitized or uninfected). $p(y)$ is the probability for the point being parasitised for all N points [35].

The adaptive learning rate optimization algorithm used was Adam with a learning rate of 0.001. Adam is a combination of two optimization algorithms– RMS prop and Stochastic Gradient descent with momentum. It scales the learning rate by using squared gradients. Instead of using the gradient snapshots, Adam uses a moving average of the momentum of the gradient. Adam is an adaptive learning rate optimizer and uses the estimations of the first and second moments of the gradient or loss to adapt the learning rate for each weight of the network for subsequent updates [36].

To speed up training, callbacks like ModelChekpoint, EarlyStopping, ReduceLROnPlateau were used [37]. Model Checkpoint monitors the validation loss and saves only the best model to decrease the memory load on disk. With a minimum change in the validation loss of 0.01 to qualify as an improvement, and patience (epochs without improvements in accuracy) of 15 epochs, the best weights were restored the subsequent weight updates. This helped speed up training a lot since the model only trained with the best weights possible. After the validation loss stopped reducing for 20 epochs, the training was stopped. To avoid the problem of local minima, the learning rate was reduced by a factor of 0.01 when the loss did not drop for 20 consecutive epochs (or reached a plateau). This reduced learning gave the optimizer time to find alternate paths towards the minima.

The images were divided into batches of size 32 [38] for each epoch. The model was then allowed to train for a theoretical 100 epochs. We evaluated the model against the already existing VGG16 and VGG19, along with the existing literature. The metrics of evaluation used were training and validation accuracy, precision, recall, F1 score and specificity. Fig. 3 shows Accuracy and Loss plotted against epochs.

IV. Results

A. Performance Evaluation

The final model was evaluated against existing literature with the metrics- cross-validation accuracy, precision, F1

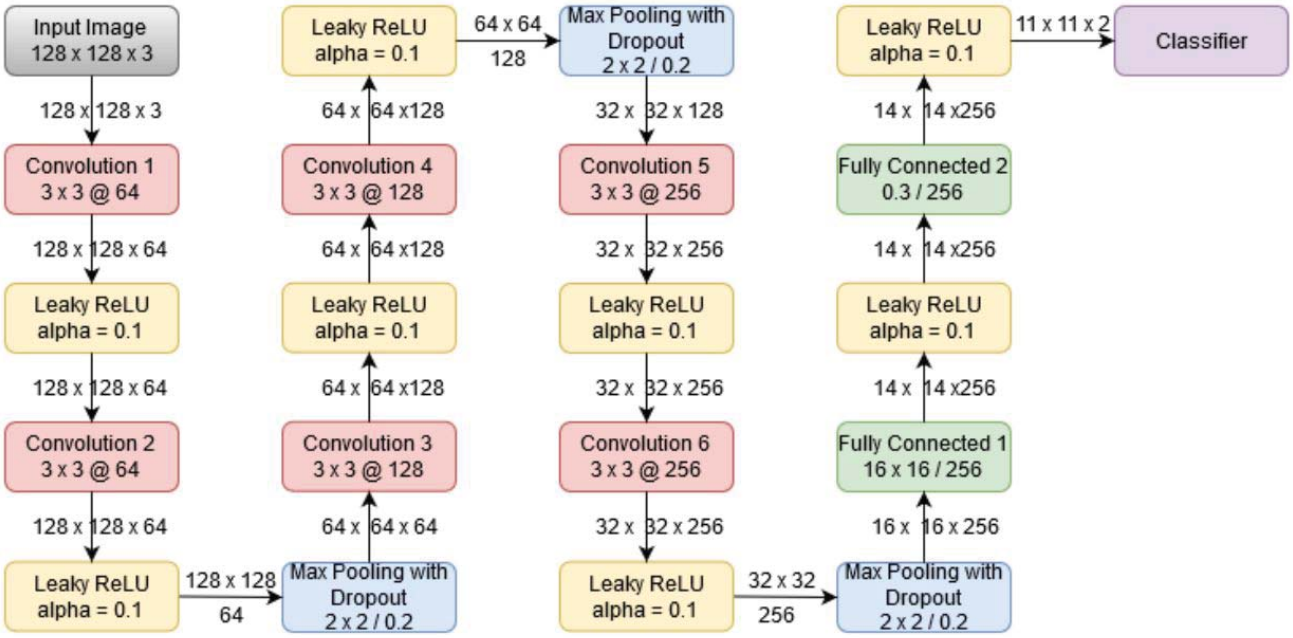

Fig. 2 Architecture of the customised VGG-based Neural Network

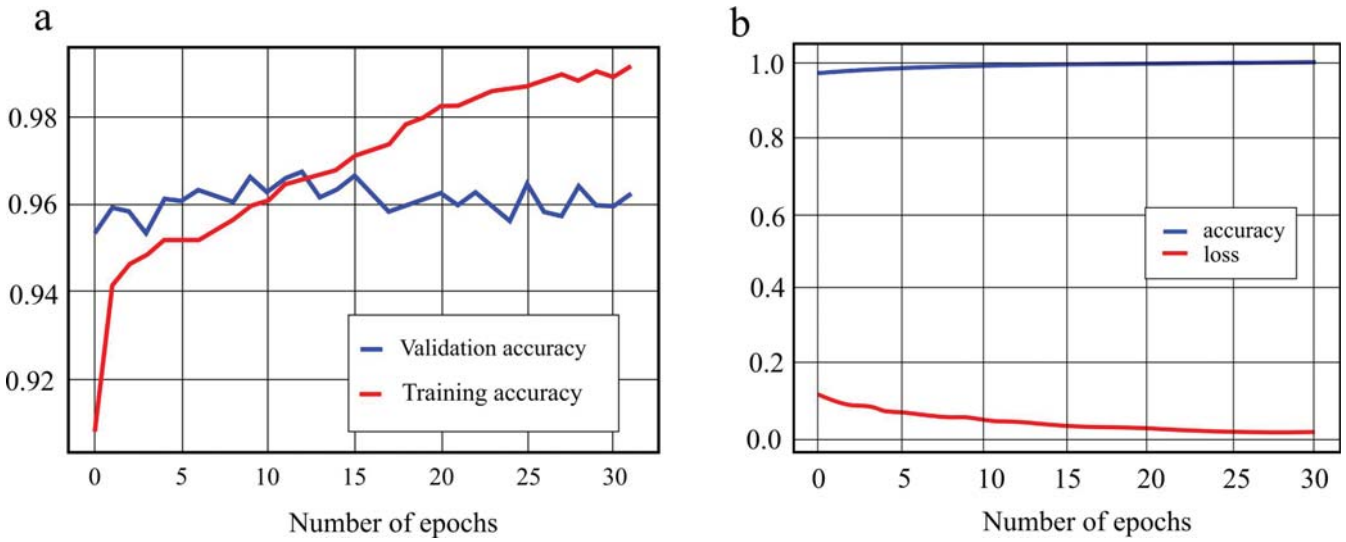

Fig. 3 (a) Accuracy vs epochs; (b) loss and accuracy vs Epochs

score, sensitivity and specificity, after tuning the hyper parameters and optimisation tools. All the metrics are explained in brief, along with the results obtained. Let us define the following metrics:

- *True positive (TP)*: the number of cells with parasites identified correctly.
- *True negative (TN)*: the number of uninfected cells identified correctly.
- *False positive (FP)*: the number of uninfected cells wrongly classified as infected.
- *False negative (FN)*: the number of cells with parasites wrongly classified as uninfected.

Accuracy is a metric which can be calculated according to

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

Precision is a measure used to determine how many correct malarial classes are predicted [39]. Precision is defined by

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{8}$$

Recall (or sensitivity) is a measure of total malarial classes correctly classified by the model [39]. It is also referred to as true positive rate [40], [41]. Recall is defined by

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{9}$$

F1-score gives a statistical measure of the accuracy of the predictions by the model. F1 score is given by

$$\mathrm{F1-score} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \quad (10)$$

Specificity is the measure of the proportion of uninfected predicted classes which are actually uninfected. Specificity is also referred to as true negative rate [40], [41]. Specificity is given by

$$\mathrm{Specificity} = \frac{TN}{TN + FP}. \quad (11)$$

The confusion matrix is also presented in Table III for [42].

TABLE III
CONFUSION MATRIX

	Predicted **No**	Predicted **Yes**
Actual **No**	2746	57
Actual **Yes**	36	2401

Table V gives a comparison of results for the existing and most recent cross-validated studies, including the model presented in this work and dataset C (bold text indicates best performing models).

V. DISCUSSION

Our model is a VGG-based model. VGG19 is too complex for the task, a simplified version works better in this application, and also avoids overfitting. VGG16 also performs worse in changing brightness-levels of different images (as seen in [43]). Only 6 convolution layers were used instead of the 13 convolution layers used by [29]. It is worth mentioning that the smaller input for VGG16 and VGG19 improves the accuracy metrics. Since the smaller images are derived from the larger images by interpolation, this proves that the type of dataset used in this research does not require very deep neural networks which are designed to grasp small features of images. Most of the images can be classified by a singularity point, representing an impurity or a parasite. Therefore, more shallow networks are more suitable for this type of classification.

In [44] it is demonstrated that a deeper neural network demands more computational resources: more memory and more computational time. They, therefore, demand more powerful and more expensive graphics cards. In this work we show that a very high accuracy can be achieved even if trained on a more shallow customised neural network than an ensemble model, which includes VGG19 and VGG16 suggested in [17]. Moreover, this work shows that our model outperforms the latest work [18] in accuracy and specificity on "perfect" dataset **C**. When tested on a closer to real-life dataset **B** our model still outperforms [18] in precision and has comparable values in accuracy and specificity.

We tested the algorithm on datasets **B** and **C**. As discussed earlier, both datasets derived from [21] were images double-checked by malaria experts; wrongly labelled images were relabeled. The main difference between datasets **B** and **C** is that dataset **B** keeps uninfected images with impurities for study, while dataset **C** has removed many them. The purpose of keeping the images with impurities is to test how the deep learning model works in the conditions close to real-life. Despite keeping the images with impuriries, Table IV shows that the accuracy on patient level data remains consistent, however, Table V shows that the accuracy dropped compared to the one achieved when training and testing on the "perfect" dataset **C**. Based on this we argue that the learning model may struggle distinguishing between parasites and impurities. More sophisticated models, possibly including feature extraction from regions of interest (suspicious regions) are required.

TABLE IV
K FOLD CROSS VALIDATION RESULTS

Fold	Parasitized Samples	Uninfected Samples	Accuracy
1	2633	2634	97.90
2	2647	2646	97.85
3	2559	2559	97.91
4	2638	2638	98.59
5	2626	2626	98.38

Our model achieves an accuracy comparable with [17] on dataset **C**.

VI. CONCLUSION

Our work shows that the deep learning can be efficiently used to detect malaria parasites on thin blood smears. We have presented a high accuracy DL-based model and tested it on two datasets. The proposed model outperforms or is comparable with the earlier studies. We demonstrated that to achieve a high accuracy one does not have to use very deep neural networks and more shallow versions maybe preferable. We have relabeled wrongly classified images in the dataset [21]. By testing on two different datasets we have confirmed the limitations of the model. More work need to be done to build a deep learning model which would be able to distinguish between true parasites, impurities and artefacts in the way a human expert would do. However, the model presented allows identification most of cases correctly. It can be treated as the first AI-based step which is able to detect if a blood cell *is highly likely* to contain a parasite, faster and more accurately than manual testing. More detailed analysis of a suspicious region, either by an expert or by a new deep learning method, is required.

ACKNOWLEDGMENT

The authors would like to thank the University of Huddersfield for sponsoring this work.

REFERENCES

[1] "World malaria report 2019," p. 232, 2019, Geneva: World Health Organization. (Online). Available: https://www.who.int/malaria/publications/world-malaria-report-2019/en/
[2] N. Tangpukdee, C. Duangdee, P. Wilairatana, and S. Krudsood, "Malaria diagnosis: a brief review," *The Korean journal of parasitology*, vol. 47, no. 2, pp. 93–102, 6 2009.
[3] C. M. Hommelsheim, L. Frantzeskakis, M. Huang, and B. Ülker, "Pcr amplification of repetitive dna: a limitation to genome editing technologies and many other applications," *Scientific Reports*, vol. 4, no. 1, p. 5052, 5 2014.

TABLE V
COMPARISON OF THE EXISTING DEEP LEARNING BASED MODELS

Method	Year	Reference	Dataset	Accuracy	Precision	Sensitivity	Specificity	F1-Score
			Cross-validation studies					
Vijayalakshmi A, et al	2019	[20]	E	89.95	93.13	93.44	92.92	91.66
Yang F, et al	2019	[19]	F	93.46	82.73	94.25	98.39	80.81
Rajaraman S, et al	2018	[16]	A	95.9	94.70	-	97.20	95.90
Rajaraman S, et al, ensemble D	2019	[17]	A	**99.50**	**99.80**	-	-	99.5
Rajaraman S, et al, VGG-19	2019	[17]	A	99.32	99.31	-	-	99.31
Rajaraman S, et al, Custom	2019	[17]	A	99.09	99.56	-	-	99.08
VGG16, input [128×128]	2014	this	B	95.62	97.28	92.17	65.02	94.66
VGG19, input [128×128]	2014	this	B	96.31	95.75	95.02	63.49	95.39
VGG16, [64×64]	2014	this	B	97.99	96.46	98.80	98.89	97.61
VGG19, [64×64]	2014	this	B	98.09	96.84	98.70	98.91	97.75
Our model	2020	this	B	**98.22**	**99.1**	97.81	98.71	**98.37**
Our model (patient level)	2020	this	B	98.18	98.8	97.81	97.67	98.11
Our model	2020	this	C	**99.30**	**99.65**	**99.66**	**99.42**	**99.33**
K. M. Faizullah Fuhad, et al	2020	[18]	C	99.23	98.92	99.52	99.17	99.22
			Other studies					
Qayyum A. et al	2019	[15]	D	96.05	95.80	96.33	-	96.06
Vijayalakshmi A. et al (VGG19-SVM)	2020	[20]	E	93.13	89.95	93.44	92.92	91.66
Our model	2020	this	B	**98.22**	**99.1**	**97.81**	**98.71**	**98.37**

[4] M. Hawkes, J. P. Katsuva, and C. K. Masumbuko, "Use and limitations of malaria rapid diagnostic testing by community health workers in war-torn democratic republic of congo," *Malaria Journal*, vol. 8, no. 1, p. 308, 12 2009.

[5] K. O. Mfuh, O. A. Achonduh-Atijegbe, O. N. Bekindaka, L. F. Esemu, C. D. Mbakop, K. Gandhi, R. G. F. Leke, D. W. Taylor, and V. R. Nerurkar, "A comparison of thick-film microscopy, rapid diagnostic test, and polymerase chain reaction for accurate diagnosis of plasmodium falciparum malaria," *Malaria Journal*, vol. 18, pp. 1475–2875, 3 2019.

[6] C. B. Delahunt, C. Mehanian, L. Hu, S. K. McGuire, C. R. Champlin, M. P. Horning, B. K. Wilson, and C. M. Thompon, "Automated microscopy and machine learning for expert-level malaria field diagnosis," in *2015 IEEE Global Humanitarian Technology Conference (GHTC)*, 2015, pp. 393–399.

[7] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, no. 1, pp. 221–248, 2017.

[8] A. Fourcade and R. H. Khonsari, "Deep learning in medical image analysis: A third eye for doctors," *Journal of Stomatology, Oral and Maxillofacial Surgery*, vol. 120, no. 4, pp. 279–288, 2019, 55th SFSCMFCO Congress.

[9] L. von Chamier, J. Jukkala, C. Spahn, M. Lerche, S. Hernández-Pérez, P. K. Mattila, E. Karinou, S. Holden, A. C. Solak, A. Krull, T.-O. Buchholz, F. Jug, L. A. Royer, M. Heilemann, R. F. Laine, G. Jacquemet, and R. Henriques, "Zerocostdl4mic: an open platform to simplify access and use of deep-learning in microscopy," *bioRxiv*, 2020.

[10] T. Falk, D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, "U-net: deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, 1 2019.

[11] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Scientific Reports*, vol. 9, no. 1, p. 12495, 8 2019.

[12] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. C. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F. J. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C. J. Kelly, D. King, J. R. Ledsam, D. Melnick, H. Mostofi, L. Peng, J. J. Reicher, B. Romera-Paredes, R. Sidebottom, M. Suleyman, D. Tse, K. C. Young, J. De Fauw, and S. Shetty, "International evaluation of an ai system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, 1 2020.

[13] WHO, "Malaria microscopy quality assurance manual," p. 140, 2016. (Online). Available: https://www.who.int/malaria/publications/atoz/9789241549394/en/

[14] K. E. Delas Peñas, P. T. Rivera, and P. C. Naval, "Malaria parasite detection and species identification on thin blood smears using a convolutional neural network," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2017, pp. 1–6.

[15] A. B. Abdul Qayyum, T. Islam, and M. A. Haque, "Malaria diagnosis with dilated convolutional neural network based image analysis," in *2019 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON)*, 2019, pp. 68–72.

[16] S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger, and G. R. Thoma, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 6, p. e4568, 2018.

[17] S. Rajaraman, S. Jaeger, and S. K. Antani, "Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images," *PeerJ*, vol. 7, p. e6977, 2019.

[18] K. Fuhad, J. F. Tuba, M. Sarker, R. Ali, S. Momen, N. Mohammed, and T. Rahman, "Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application," *Diagnostics*, vol. 10, no. 5, p. 329, 2020.

[19] F. Yang, M. Poostchi, H. Yu, Z. Zhou, K. Silamut, J. Yu, R. J. Maude, S. Jaeger, and S. Antani, "Deep learning for smartphone-based malaria parasite detection in thick blood smears," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1427–1438, 2020.

[20] A. Vijayalakshmi and B. R. Kanna, "Deep learning approach to detect malaria from microscopic images," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15 297–15 317, 6 2020.

[21] "Malaria datasets," 2019, National Library of Medicine. (Online). Available: ftp://lhcftp.nlm.nih.gov/Open-Access-Datasets/Malaria/

[22] "Corrected malaria data II," 2020, google Drive. (Online). Available: https://drive.google.com/drive/folders/1GeQap_A5rc29NnBTAewe52pb0JpmLyVJ

[23] "Corrected malaria data I," 2019, google Drive. (Online). Available: https://drive.google.com/drive/folders/10TXXa6B_D4AKuBV085tX7UudH1hINBR

[24] Kaggle, "Malaria cell images dataset," 2019. (Online). Available: https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria

[25] M. Group, "The mamic image database," 2014. (Online). Available: http://fimm.webmicroscope.net/Research/Momic/mamic

[26] J. Shao, "Linear model selection by cross-validation," *Journal of the American statistical Association*, vol. 88, no. 422, pp. 486–494, 1993.

[27] "Datasets B, C, codes," 2020, google Drive. (Online). Available: https://drive.google.com/drive/folders/1IHeihe6PlJuCQLvL796D1wMrF5tP99OS
[28] F. Chollet, "Building powerful image classification models using very little data," *Keras Blog*, 2016.
[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.
[30] R. Ilango, "Batch normalization — speed up neural network training," 2018. (Online). Available: https://medium.com/@ilango100/batch-normalization-speed-up-neural-network-training-245e39a62f85
[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
[32] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30 th International Conference on Machine Learning*, vol. 30, no. 1, 2013, p. 3.
[33] A. Fourcade and R. Khonsari, "A tutorial on fisher information," *Journal of Stomatology, Oral and Maxillofacial Surgery*, vol. 120, no. 4, pp. 279–288, 2019, 55th SFSCMFCO Congress.
[34] L. Liu and H. Qi, "Learning effective binary descriptors via cross entropy," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 1251–1258.
[35] D. Godoy, *Binary Crossentropy log loss*, 2018. (Online). Available: towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.
[37] Tensorflow, *Keras Callbacks*, 2020. [Online]. Available: www.tensorflow.org/guide/keras/custom_callback
[38] E. Goceri and A. Gooya, "On the importance of batch size for deep learning," in *An Istanbul Meeting for World Mathematicians*, 2018, p. 100. [Online]. Available: raims.org/files/Abstract_Book_2018.pdf
[39] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.
[40] D. G. Altman and J. M. Bland, "Diagnostic tests. 1: Sensitivity and specificity," *BMJ: British Medical Journal*, vol. 308, no. 6943, p. 1552, 1994.
[41] W. Zhu, N. Zeng, N. Wang *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations," *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, vol. 19, p. 67, 2010.
[42] F. Provost and R. Kohavi, "Glossary of terms," *Journal of Machine Learning*, vol. 30, no. 2-3, pp. 271–274, 1998.
[43] K. Grm, V. Štruc, A. Artiges, M. Caron, and H. K. Ekenel, "Strengths and weaknesses of deep learning models for face recognition against image degradations," *Iet Biometrics*, vol. 7, no. 1, pp. 81–89, 2017.
[44] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.