Article

Deep Malaria Parasite Detection in Thin Blood Smear Microscopic Images

Asma Maqsood [1,†], Muhammad Shahid Farid [1,*,†], Muhammad Hassan Khan [1,†] and Marcin Grzegorzek [2,†]

1. Punjab University College of Information Technology, University of the Punjab, Lahore 54000, Pakistan; MSCSF18M003@pucit.edu.pk (A.M.); hassankhan@pucit.edu.pk (M.H.K.)
2. Institute of Medical Informatics, University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck, Germany; grzegorzek@imi.uni-luebeck.de
* Correspondence: shahid@pucit.edu.pk
† These authors contributed equally to this work.

Abstract: Malaria is a disease activated by a type of microscopic parasite transmitted from infected female mosquito bites to humans. Malaria is a fatal disease that is endemic in many regions of the world. Quick diagnosis of this disease will be very valuable for patients, as traditional methods require tedious work for its detection. Recently, some automated methods have been proposed that exploit hand-crafted feature extraction techniques however, their accuracies are not reliable. Deep learning approaches modernize the world with their superior performance. Convolutional Neural Networks (CNN) are vastly scalable for image classification tasks that extract features through hidden layers of the model without any handcrafting. The detection of malaria-infected red blood cells from segmented microscopic blood images using convolutional neural networks can assist in quick diagnosis, and this will be useful for regions with fewer healthcare experts. The contributions of this paper are two-fold. First, we evaluate the performance of different existing deep learning models for efficient malaria detection. Second, we propose a customized CNN model that outperforms all observed deep learning models. It exploits the bilateral filtering and image augmentation techniques for highlighting features of red blood cells before training the model. Due to image augmentation techniques, the customized CNN model is generalized and avoids over-fitting. All experimental evaluations are performed on the benchmark NIH Malaria Dataset, and the results reveal that the proposed algorithm is 96.82% accurate in detecting malaria from the microscopic blood smears.

Keywords: malaria detection; *Plasmodium* parasite; transfer learning; convolutional neural networks; computer aided design (CAD)

1. Introduction

Malaria is a disease activated by a type of microscopic parasite transmitted from infected female mosquito bites to humans. The malaria parasite ruptures the red blood cells in human blood and replicates these parasites to other cells. Malaria patients usually feel very sick, with high fever, headache, muscle pain, and fatigue. According to the World Health Organization (WHO), malaria parasites caused the death of 438,000 people in 2015 and 620,000 people in 2017, and infection cases are around 300–500 million annually [1]. Light microscopy is a standard method for identifying malaria disease and all its species of parasite by screening films of red blood cells. Other methods, like rapid diagnostic tests [2], are also used for a prompt parasite-based diagnosis. It is a widely used test with a false positive rate of less than 10%. It is helpful in initial diagnosis; however, its performance is affected by the quality of the product and parasite-related factors [3]. For examining malaria infection in light microscopy, the glass slide is prepared by applying a drop of blood, which is merged with the Giemsa staining solution to enhance the visibility of parasites in red blood cells under a microscope. Furthermore, thick and thin smears of blood are used for malaria diagnosis. A thick smear typically identifies the existence of

the parasite in blood while a thin smear detects species of malaria and parasite stages. An expert microscopist usually takes 20 to 30 min for a careful examination of a single blood film to count the number of infectious cells by inspecting the variations in the shape, color, and size characteristics of red blood cells.

Millions of blood smear films are manually examined by expert pathologists every year and it takes a massive human and economic effort to diagnose malaria. Moreover, the parasite counts from blood films should be accurate for a correct diagnosis and classification of disease severity. For example, if malaria cells were not present in a patient but the doctor erroneously prescribed antibiotics, it would unnecessarily cause abdominal pain or nausea to the suspected patient [4]. Diagnosis of malaria should be robust with high sensitivity (less false negatives) towards capturing parasites at all stages of the malaria life-cycle. Correct diagnosis at earlier stages can be helpful for the treatment of malaria in endemic regions where expert pathologists are fewer, and the workload of screening blood films is massive. Automatic malaria detection methods can serve many patients with fast, cost-effective, and accurate diagnoses.

Traditional methods of automating the malaria detection process involve complex image-processing techniques with hand-engineered features e.g., shape, color, intensity, size, and texture [5–7]. In these methods, the red blood cells are detected from microscopic images by using different segmentation techniques. After the selection of appropriate features for red blood cells, a computed set of features is used in the classification of segmented images into infected and uninfected classes. For example, morphological-based approaches are used to segment cell images with structuring elements to enhance the characteristics of red blood cells, such as the roundness of cells, which improves the classification accuracy. In the literature, various methods are adopted for the segmentation, feature extraction, and classification of malaria diagnosis [8]. After analyzing conventional and recent malaria detection methods, it is observed that there is a tradeoff between accuracy and the computational complexity of models, that is, when the accuracy of a model increases, its computational complexity also increase [9]. For example, the computation time of classification by support vector machine (SVM) is faster than a deep neural network, but the accuracy of a deep neural network is found to be higher than SVM.

In recent years, deep learning (DL) techniques are exploited for automated malaria diagnosis with appreciable detection rates. Deep learning models eliminate the computation of hand-crafted features, as the hidden layers of deep models extract features automatically by analyzing the data. Deep learning models require large datasets for training neural networks and to improve the accuracy of the model. However, in medical applications like malaria diagnosis, relatively small datasets are available. This is because building an annotated dataset requires input from pathologists that is not readily available. To overcome the paucity of the dataset, recently introduced image augmentation techniques in deep learning models provide better generalization and reduce the over-fitting. Image augmentation increases the dataset by taking the original image and transforming it into multiple images by using transformation techniques such as rotation, shear and translation, thus enabling the model to achieve higher accuracy. A convolutional neural network (CNN) is widely used for classification tasks and it is computationally efficient too [10].

In this paper, we evaluate the effectiveness of various existing deep learning models for malaria detection from microscopic blood images, and also propose an efficient DL method for the classification of infected and uninfected malaria cells. The proposed customized CNN-based algorithm outperforms all observed deep learning models. The proposed method uses bilateral filtering for improving image quality and image augmentation techniques for better generalization of the model. Our model has a simple CNN architecture containing 5 convolutional and pooling layers. The performance of the proposed method is evaluated on a benchmark malaria dataset, and the results are compared with other existing, similar techniques. The results show that our method achieves excellent performance and outperforms the compared techniques.

The rest of the paper is organized as follows. The literature on automated malaria detection is reviewed in Section 2. The various deep learning models used for performance evaluation are presented in Section 3. The proposed deep learning method is presented in Section 4. The details of the test dataset and preprocessing of the data, performance evaluation of the proposed method, and comparison of deep learning models are described in Section 5. The research is concluded in Section 6.

2. Related Work

The traditional pipeline of automating the process of malaria diagnosis follows four steps: image pre-processing, cell segmentation, feature selection, and classification of infected and uninfected malaria cells, as shown in Figure 1. For each step, different methods are proposed in the literature. Image pre-processing methods enhance the quality of blood smear images to improve the accuracy of later processing steps such as cell segmentation, feature extraction, and classification. Any kind of impurities in the images can affect the performance of the later processing steps and are vulnerable to the misclassification of malaria cells. Different smoothing filters, e.g., Gaussian, median, and geometric mean filters, are extensively used to suppress noise in microscopic images [11,12]. Morphological operators have also been used to remove impurities by improving cell contours and suppressing noise by filling holes [13]. Adaptive threshold and histogram equalization is also used to enhance the resolution and contrast of the images [14]. Malaria detection methods e.g., [15] used HSV color space and grayscale color normalization to reduce illumination variation in cells. Low-pass filters have also been exploited to eliminate noise-related frequency components from the microscopic images [16]. Some techniques e.g., [17] used the Laplacian filter to sharpen edges and to enhance the red blood cell (RBC) boundaries in images. The method in [18] used the Wiener filter to remove the blurriness induced in the microscopic images due to unfocussed optics.
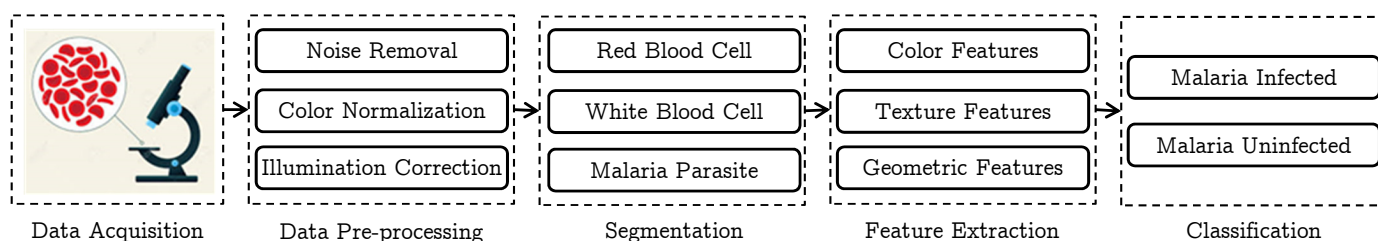

Figure 1. Traditional automated malaria detection pipeline.

Cell segmentation is the most significant step in any automated malaria detection system. Pre-processed microscopic images are segmented into small non-overlapping regions containing the red blood cells (RBC), the white blood cells (WBC), the malaria parasites, and other artifacts. Image-based methods such as Chan–Vese segmentation, hole-filling algorithms, and histogram-method-based methods are popular for cell segmentation in an unsupervised manner [19]. A green channel of RGB images is used to segment cells in case of low-contrast images in [20]. In [21], the Otsu threshold is used to segment RBCs from enhanced images. In malaria detectors such as [22,23], thresholding techniques are applied in Hue-Saturation-Value (HSV) color space on S and V channels to segment cells from microscopic images. The fuzzy divergence technique is used for cell segmentation in [24]. The fuzzy rule-based segmentation method is applied in [25] to segment malaria cells from images of three different color spaces. In [13], morphological-based approaches use grayscale granulometry to capture regional extrema and segment cells. Hough transform is used in [26] to identify RBC using its shape, and k-means clustering is used to segment cells from unlabeled data in [27]. Marker controlled watershed algorithm is used in [28,29] to separate the overlapping cells that complicate the segmentation process. A graph-cuts based technique for cell segmentation is proposed in [30]. The methods in [31,32] utilize

the structure, geometry, and color information of cells to identify WBC and gametocytes. Machine learning approaches and neural networks are used for cell segmentation in [33].

Feature selection for cell images depends on the shape, texture, and color of red blood cells. HSV color space and green channel of RGB color space are preferred for feature extraction because the color features are prominent in stained blood images [31,34]. Histogram of oriented gradients (HOG) features [26], Haralick's texture features [35], local binary patterns [36], and various other feature-selection methods [37] have been used to extract features from cell images. Types of parasite can be identified from cell images by using the color and shape information [38]. Morphological operations like thinning and grayscale granulometry capture relevant information from the intensity of image pixels [39,40]. In [12], the support vector machine (SVM) and Bayesian learning are used for the classification of malaria cells by utilizing the discriminative feature set.

In the malaria detection algorithm presented in [41], linear Euclidean distance classifier with Poisson distribution and Gabor filtering are used to detect malaria cells in blood smear images. An adaptive neuro-fuzzy interface system (ANFIS) is used to diagnose species of malaria infection in [42]. A genetic algorithm based on chromosome-encoding schemes and mutation strategies are also utilized to diagnose malaria-infected cells [43]. K-means clustering is used for the unsupervised detection of malaria-infected cells in [19]. In [44], SVM and artificial neural network (ANN) detect the malaria parasite by using the information of normalized red, green, and blue, and the texture features that are invariant to staining variations.

Deep learning (DL) revolutionized the traditional malaria detection pipeline by skipping the feature selection step that requires expertise to capture variability in the angle, position, shape, texture, color, and size of objects in images. Deep learning became popular for medical image analysis because of its biggest advantage of learning features from underlying data without any designing of feature sets. Deep learning models use non-linear activation units for neurons in layers that discover underlying hierarchical patterns in the data. Features are extracted using end-to-end hidden layers that learn complex decision-making functions, leading to classification. A deep learning CNN model is used by Dong et al. [45] for cell segmentation and deep belief networks are used for the classification of malaria-infected and uninfected images. The CNN models [46] can recognize patterns in microscopic images with a much higher accuracy rate than other traditional approaches. Fully convolutional regression networks (FCRN) [47] regress CNN spatial feature maps to detect and count cells in microscopic images. Deep learning approaches used for cell segmentation provide more accurate results as compared to previous techniques.

The convolutional neural networks are used for images by exploiting the information of spatial local correlation between adjacent pixels by using shared weights, local receptive fields, and down-sampling of feature maps. In recent researches, customized CNN with focus stack images showed improvements in classification performance [48]. In [49], a customized CNN model with 16 layers is proposed that outperforms all transfer learning models. The CNN Models AlexNet and VGG-16 are compared in [50] to improve the accuracy of malaria detection. Hung et al. [51] uses two-stage classification with faster RCNN for detecting red blood cells in images and and used AlexNet for the classification of malaria cells. The method in [52] proposed LeNet-5 for automated diagnosis of malaria. The VGG-16 is unified with SVM in [53] to classify *falciparum* malaria cells. Pre-trained CNN models trained on imagenet weights e.g., VGG, ResNet, etc. are used as feature extractors in [54] and classify malaria-infected and uninfected cells. The malaria detectors presented in [55,56] also use deep learning. An ensemble of three deep learning models, customized CNN, VGG-16, and CNN with SVM, is proposed in [57] for malaria parasite detection.

3. Compared Deep Learning Models

In visual recognition tasks, the computer processes images by their pixel values. To process a 30×30 color image (in RGB), 2700 ($30 \times 30 \times 3$) pixel values need to be processed. Neural network is a sequence of layers with neurons that distinguish underlying

relationships in a set of data. In fully connected neural networks, if we take an input image with $30 \times 30 \times 3$ pixels, the number of weights in the first hidden layer will be 2700. In real life, we have large images with 250×250 dimensions and more. If we take an input image with $250 \times 250 \times 3$ pixels, then the number of weights in the first hidden layer will be 187,500. Therefore, we need to deal with a massive amount of parameters and need more neurons for deep networks that can cause over-fitting.

In a convolutional neural network (CNN), the neuron in a layer will only connect to a small number of neurons in the previous layer, instead of all neurons in a fully connected way, which is why we need to handle fewer weights and fewer neurons. This is the reason that CNN performs better than fully connected neural networks for image classification tasks [58]. The CNN architecture has two important layers: convolutional layers and pooling layers. The convolutional layer performs the convolution operation on images to extract feature maps and the pooling layer reduces the size of the feature maps by performing down-sampling.

Convolutional neural networks with several hidden layers and a large number of parameters are fantastic for image classification tasks [59]. CNN masters spatial patterns from images which are also invariant to translation and learn the different features of images. VGG, ResNet, DenseNet, and Inception [60–62] are popular due to their outstanding performance in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [63]. Designing a network architecture is a tedious process and requires a lot of effort. There are various architectures designed for different problems. For the detection of malaria-infected cells, the performance of various architectures of CNN, e.g., customized CNN model, VGG, ResNet, DenseNet, Inception, Xception, and SqueezeNet are analyzed in this research. Each architecture is briefly introduced in the following sections.

3.1. VGG

This is a CNN model designed by Visual Geometry Group (VGG) of Oxford University for large-scale image recognition [64]. The VGG model is one of the prominent models in the ILSVRC-2014 competition [63] and accomplishes 92.7% accuracy on the testing data of ImageNet. It has two flavors: VGG16 and VGG19. The former has 16 layers with 5 max-pooling layers and 5 blocks of convolutional layers, where each block has two or more convolutional layers. The convolutional layers of this model only use 3×3 kernel size and the max-pool layer uses 2×2 kernel size. The VGG16 beats the previous models in ILSVRC-2012 and ILSVRC-2013 competitions. The VGG19 has 19 layers with 5 max-pooling layers and 5 blocks of convolutional layers. The only difference between VGG19 and VGG16 is the last 3 convolution blocks, which have 4 convolutional layers in VGG19 and 3 convolutional layers in VGG16.

3.2. ResNet

Residual Neural Network (ResNet) [60] is a convolutional neural network that can train a large number of layers with a compelling performance. It provides a state-of-the-art solution to the vanishing gradient problem. The training of neural networks relies on the back-propagation process, which needs a gradient descent to minimize the loss function and learns the model weights. For a large number of layers, the gradient becomes smaller, and even vanishes due to repeated multiplication, and performance degradation occurs with each additional layer. ResNet uses an identity skip connection that skips one or more layers and reuses the feature maps from previous layers. Skipping the layers squeezes the network, which allows faster learning. The layers expand during training and the residual portions of the network discover more features from the source image. ResNet has many flavors, such as ResNet-50, ResNet-101 and ResNet-152. ResNet-V2 differs from ResNet-V1 because it uses batch normalization before each weight layer.

3.3. DenseNet

In Dense Convolutional Neural Network (DenseNet) [61], every layer is connected with all preceding layers and the feature maps of previous layers are used as input to the new layer, and its feature map is used as input for the succeeding layer. The traditional convolutional networks have a direct connection between layers, and features of previous layers not provided to the next layer. DenseNet achieves good performance with a smaller amount of computation. DenseNet exploits the network potential by reusing the learned features and not learning the redundant features again in subsequent layers [65]. DenseNet concatenates feature maps of previous layers with a new layer, while ResNet adds feature maps of layers. Each layer of DenseNet learns a small set of new feature-maps that require fewer parameters for training. DenseNet improves the vanishing-gradient problem by allowing each layer to directly access the gradients from the loss function. DenseNet has many flavors, such as DenseNet-121, DenseNet-169, and DenseNet-201.

3.4. Inception

Inception is a deep neural network designed by Google that plays an important role in the development of convolutional network classifiers [62]. To enhance the performance of the model in terms of accuracy and speed, inception uses a lot of tricks. It has an inception module that performs a convolutional operation on the input with three filters of different sizes (1×1, 3×3, 5×5) and also performs max-pooling. The outputs of these 4 layers (3 convolutional and 1 pooling) are concatenated and fed to the next inception module in the network. To reduce the computational cost of the network, the number of input channels is limited by adding 1×1 convolution before performing 3×3 and 5×5 convolutions, because 1×1 convolution operation is cheaper and reduces the number of input channels [66]. The inception has many variants like Inception-v1, Inception-v2, Inception-v3, and Inception Resnet-v2 [67]. The inception model evolves by using smart factorization methods that reduce the convolution operation cost. It also reduces the representational bottleneck in which information is lost by reducing the dimensions of input data. The Inception-v3 model contains 11 inception modules.

3.5. Xception

Xception (extreme version of Inception) [68] is a deep neural network designed by Google (Mountain View, CA, USA). Xception model modifies the depth-wise separable convolution of inception model which implements 1×1 convolution before channel-wise spatial convolution. The Xception network has 71 layers designed for large-scale image recognition. There is non-linearity in the inception module after the first operation but there is no in-between ReLU or ELU non-linearity in the Xception model. It achieves the excellent accuracy without any intermediate activation. The Xception model also uses residual or identity skip connections to increase accuracy that is the core idea of the ResNet network. The Xception model outperforms VGG-16, ResNet-152, and Inception-v3 models trained on the imagenet dataset [69]. Xception model architecture has the same number of parameters as that of Inception-v3 but, due to effective use of these parameters, the performance of the Xception model is improved [68]. These performance gains are not due to increased capacity but due to the more efficient use of model parameters.

3.6. SqueezeNet

SqueezeNet [70] is a small neural network with 18 layers that can fit into a small amount of memory and requires less bandwidth over a computer network. It has 50-times fewer model parameters than AlexNet but it achieves AlexNet-level accuracy. In SqueezeNet, 3×3 filters are replaced by 1×1 filters, and the number of input channels to 3×3 filters is decreased by using squeeze layers to reduce model parameters. Delayed down-sampling in a network attains large activation maps that maximize accuracy with limited model parameters. SqueezeNet has a Fire module in which the squeeze layer performs convolution by using a 1×1 filter, and then the output forwards to expand layer that has a mixture of 1×1 and

3×3 convolutions. SqueezeNet has a $50\times$ reduced model size than AlexNet, which is a much higher reduction than SVD and deep compression. SqueezeNet has many flavors, such as vanilla squeezeNet and SqueezeNet with simple or complex bypass. The attributes of the DL architectures described in this section are summarized in Table 1.

Table 1. Summary of attributes of the compared deep learning models.

Model	Attributes
VGG	– Multiple 3×3 sized kernels are used instead of large-sized kernels which enables it to learn more complex features at a lower cost. – The 3×3 kernels preserve finer-level properties of the image. – Huge computation is needed both in terms of space and time. – The large width of convolutional layers makes it inefficient.
ResNet	– It uses an identity skip connection that skips one or more layers and reuses the feature maps from previous layers. – Faster learning is achieved by skipping the layers in the network. – It handles the vanishing-gradient problem.
DenseNet	– It achieves good accuracy with a smaller amount of computation. – It reuses the learned features in subsequent layers which reduces the its number of parameters. – It provide better solution to vanishing-gradient problem.
Inception	– It uses different size convolutions to capture details at varied scales. – To reduce the computational cost, the number of input channels is limited by adding 1×1 convolution before performing large convolutions. – Smart factorization methods reduce the convolution cost. – It replaces the FC layers with global average pooling layers without compromising the accuracy, resulting in few parameters.
Xception	– It achieves the high accuracy without any intermediate activation. – It also uses residual or identity skip connections to increase accuracy. – It has the same number of parameters as Inception V3, the performance improves due to efficient use of model parameters.
SqueezeNet	– It requires small memory and less bandwidth over the network. – Model parameters are reduced by using squeeze layers which perform convolution by using a 1×1 filter. – Delayed down-sampling in a network maximizes accuracy with limited model parameters.

4. Proposed Deep Learning Model for Malaria Detection

In this section, we present a novel neural architecture for malaria detection from microscopic thin blood smear images. The proposed method can be divided into three parts, data preprocessing, feature extraction, and classification. Figure 2 shows these steps in a schematic diagram.
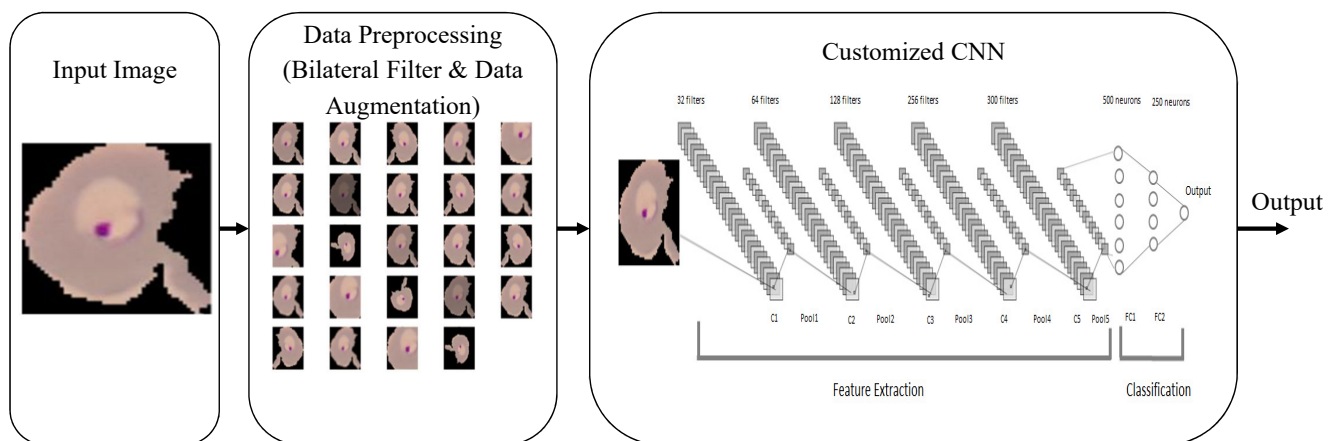
Figure 2. Block diagram of the proposed malaria detector.

Before introducing the proposed DL model for efficient malaria detection, we instigate a data pre-processing step that is effective for improving the quality of the images. During data acquisition, the images may get polluted with various types of noise, e.g., camera angle and microscope positions. Different noise removal methods have been proposed in the literature to eliminate such kinds of noise from the images. These methods include simple blur operators such as averaging filter and non-linear filters e.g., median filter. In our case, the RBC scans are of low resolution and contain important information about the parasite presentation that could be degraded or lost if simple blurring methods are used. We need an image denoising method that removes the noise from the image, preserving the structural information present in the image. We found the bilateral filter [71] to be quite effective in this scenario, as shown in [11].

In conventional image blurring methods, in the computation of the new pixel value, each pixel contributes based on its distance from the filter center. The filter weights in the bilateral filter consider both the spatial distance of the pixel from the filter and the pixel color/intensity range differences. The former factor introduces the blur in the image and the latter tries to preserve the structural information in the image.

Let I be an input image of size $M \times N$ which is subject to bilateral fileting of size size $(2d+1) \times (2d+1)$. The value of pixel (x,y) is computed as

$$\bar{I}(x,y) = \frac{1}{w} \sum_{-d \leq i,j \leq d} \left[I(x+i, y+j) g_r(\|I(x+i, y+j) - I(x,y)\|) g_s(\|i\|, \|j\|) \right] \quad (1)$$

where $\bar{I}$ is the filtered image, w is the normalizing factor, and g_r is the range kernel computed as

$$g_r(i) = \exp^{-\frac{i^2}{2\sigma_r^2}}, \quad (2)$$

The g_s contains the distance of pixel (i,j) from the center pixel (x,y). It is calculated as

$$g_s(l,m) = \exp^{-\frac{(l^2+m^2)}{2\sigma_s^2}}, \quad (3)$$

where σ_r and σ_s are variance parameters. All images are filtered using Equation (1) and are resized to equal dimensions (125×125), as the deep learning models require the same input shape for all images of the dataset. Figure 3 shows the results of applying bilateral filtering on sample infected and uninfected cell images.
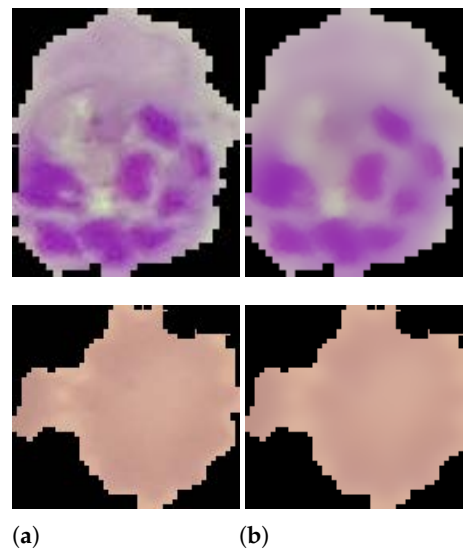
(**a**)   (**b**)

Figure 3. Bilateral filtering results: (**a**) input infected and uninfected images, (**b**) results of applying bilateral filtering on (**a**).

We propose a customized CNN model for efficient malaria detection by the classification of infected and uninfected RBC images. The input images with $125 \times 125 \times 3$ dimensions are fed toa model with 5 convolutional layers, 5 max-pooling layers, and 2 fully-connected layers. The proposed model is shown in Figure 4. The convolutional layers of model use 3×3 kernel size, ReLU activation function, and 32, 64, 128, 256, 300 filters for all five layers, respectively. By visualizing the convolutional layers of the customized CNN model in Figure 5, it can be seen that the initial layer extracts low-level features that are more recognizable to humans and the final convolutional layer extracts high-level features that are more model-recognizable.
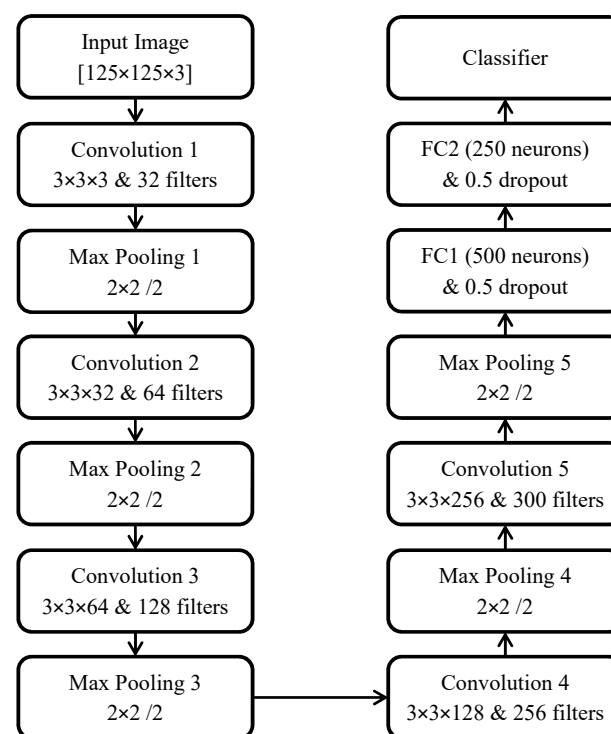

Figure 4. Proposed customized model architecture for malaria detection.
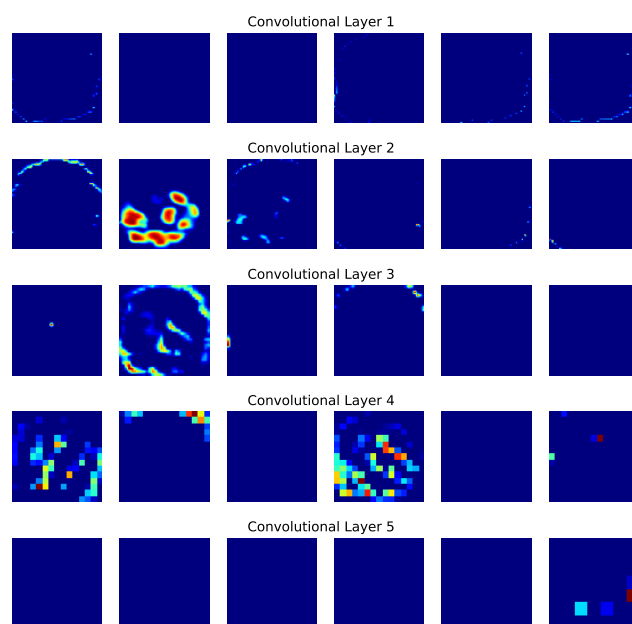
Figure 5. Visualizing Convolutional layers of CNN model.

All max-pooling layers of the model following the convolutional layers use 2×2 pool size with the stride of 2 pixels that crisps the outputs of feature maps produced by the convolutional layers. The output of the fourth pooling layer is fed to the fully connected (FC) layers with 0.5 dropouts after each FC layer, and the output of the final FC layer is fed to the sigmoid classifier. The model is trained with hyper-parameters, 64 batch size, 25 epochs, ADAM optimizer, binary cross-entropy loss, and 0.5 dropout ratio for regularization, as many pieces of research recommended this ratio of dropout [72]. The dropout layer is used to reduce the over-fitting of the model and to generalize the results of the model.

5. Experimental Evaluations and Results

In this section, we evaluate the performance of different deep learning architectures for malaria detection using the NIH Malaria dataset [54]. A time complexity analysis is also presented for these models. Moreover, the performance of the proposed method is also compared with the existing malaria-detection algorithms using different statistical measures.

5.1. Data Acquisition

The NIH Malaria dataset [54] used to evaluate the performance of the compared models which is publicly available on the National Institute of Health (NIH) (https://lhncbc.nlm.nih.gov/LHC-publications/pubs/MalariaDatasets.html, accessed date: 18 February 2021). Researchers developed a smartphone application attached with a traditional light microscope [9] for the screening of infected and uninfected red blood cell images at the Lister Hill National Center for Biomedical Communications (LHNCBC), part of the National Library of Medicine (NLM). Thin blood smear films of 50 healthy and 150 *P. falciparum*-infected patients were prepared by using the Giemsa staining solution to enhance the visibility of parasites and were photographed at Chittagong Medical College Hospital, Bangladesh. Images of blood films are captured by the built-in smartphone camera for each microscopic field of view. These images are then manually annotated by expert slide readers at the Mahidol-Oxford Tropical Medicine Research Unit, Bangkok, Thailand. The dataset has 27,558 cell images that are equally balanced between 13,779 parasitized and 13,779 uninfected. The cell images have variations in color distributions due to different bloodstains during the process of data acquisition. Figure 6 shows samples of parasitized and uninfected segmented red blood cell images from the malaria dataset.
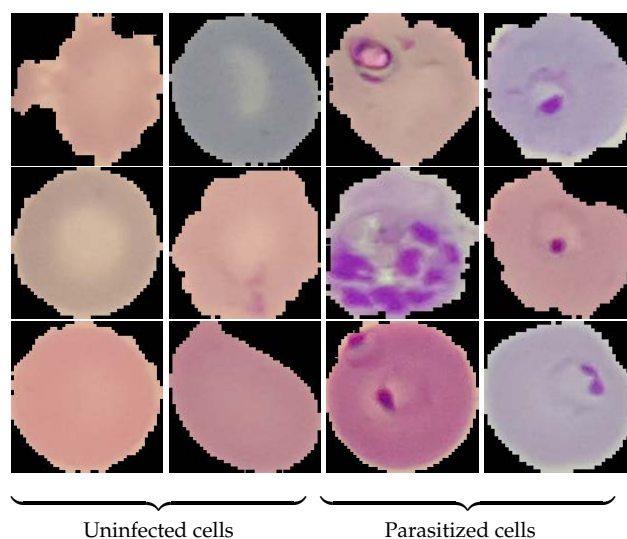
Figure 6. Sample images of parasitized (infected) and uninfected red blood cells from NIH Malaria dataset [54].

5.2. Data Preprocessing

The NIH malaria dataset is balanced with 13,779 parasitized and 13,779 uninfected cell images. To evaluate the performance of the deep learning models, the dataset is split into training, testing, and validation subsets. To this end, 60% data are used for training and 10% data for validation and the remaining 30% unseen dataset is used for testing the performance of the trained model. Partitioning details of the dataset are presented in Table 2.

Table 2. Partitioning of the test dataset into training, testing, and validation datasets before performing data augmentation.

Dataset	Parasitized	Uninfected
Training	8604	8766
Testing	4196	4076
Validation	979	952

Data augmentation techniques are usually used to improve the accuracy of deep learning models by providing variety in the dataset [73]. Neural networks require large datasets for better generalization and to avoid over-fitting. The image data generator builds powerful deep learning models by using a small dataset [74]. After resizing images of the dataset, ImageDataGenerator module of Keras library https://keras.io/ (accessed date: 18 February 2021) is used for augmenting malaria cell images of training data by applying image transformation operations: 0.1 zoom value, 25 degree rotation, 0.05 sheer range with horizontal flip, (0.1, 0.1) translation for shifting both width and height. Image augmentation is not applied to testing and validation data because model performance is being evaluated for them. After data augmentation, the size of the training dataset is 173,700.

5.3. Implementation Details

All experiments are performed on Google Colaboratory with GPU runtime using 28 GB RAM and 68 GB hard disk. All the models are implemented using the Python 3.6, Keras deep learning library with TensorFlow https://www.tensorflow.org/ (accessed date: 18 February 2021). Architectures of deep learning CNN models were downloaded from the internet and trained with hyper-parameters, such as $32 - 64$ batch size, 1×10^{-4}, 1×10^{-5} learning rate, 20–30 epochs, RMSProp or ADAM optimizer, binary cross-entropy loss, and

0.3–0.5 dropout ratio for regularization. The models were initialized with random weights and trained on the training data using the tensorflow library.

The cross-validation technique uses an independent dataset to evaluate the performance of the machine learning models on unseen data. To this end, the sample dataset is partitioned into *k* subsets of data and then training is performed on one subset of the data and validation on the other subset of the data. We evaluated the predictive models through five-fold cross-validation over 5 different test sets, with each fold having 2756 test samples and 24,802 training samples. The training data are randomly partitioned into 5 equal-sized subsets, one subset is used for validation testing and the remaining 4 subsets are used for training. The cross-validation process is then repeated 5 times for the proposed model, with each of the 5 subsets used exactly once as the validation data. The results of the validations were averaged to produce a single score.

5.4. Performance Evaluation and Comparison

For binary deep learning models, the confusion matrix is used to describe the prediction results on labeled test data. There are four possible outcomes of such a test: true-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN). The true-positive represents the infected cells correctly diagnosed as infected and the true-negative denotes the number of uninfected cells correctly diagnosed as uninfected. The uninfected cells incorrectly diagnosed as infected are considered false-positive, and the infected cells incorrectly diagnosed as uninfected are false-negatives. Different parametric and non-parametric statistical measures are used in performance evaluation, including Specificity, Sensitivity, Precision, Accuracy, F_1 score, Matthews correlation coefficient (MCC), and Cohen's kappa (κ) [75–77].

Specificity calculates the proportion of actual negatives out of total negative observations that are predicted by an algorithm. It is also known as the true negative rate and is computed as

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

Sensitivity is the ability of an algorithm to correctly predict the true positives out of total positive observations. It is also known as true positive rate or Recall.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5}$$

Precision shows the extent of correctness of the an algorithm in terms of positive results. It is computed as actual positives out of total positive observations.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The ability of an algorithm to differentiate between healthy and infected subjects flawlessly is called accuracy. In the case of the malaria diagnosis model, if an infected cell image is predicted correctly by the model as infected and vice-versa, then the model has high accuracy.

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

The F_1 score is the weighted harmonic mean of precision and recall measures. It provides an overall accuracy of the model using both positive and negative predictions. It is considered to be a more reliable performance measure than accuracy and precision metrics that can be delusive when the data are highly unbalanced [78,79]. For balancing the classification model with precision and recall, the F_1 score of the model on test data should be high.

$$F_1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

The Matthews correlation coefficient (MCC) evaluates the performance of a binary classification model with a range of −1 to +1. The minimum value −1 indicates a poor classifier while the maximum value +1 indicates a correct classifier. The MCC is regarded as a balanced measure because it considers both positive and negative observations. A recent research [80] showed that MCC is a more informative and realistic measure than other parametric statistical measures. It is computed as:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

Cohen's kappa (κ) measures the agreement between total accuracy and random accuracy of a model. Kappa value, computed as Equation (10), ranges from 0 to 1. The lowest value 0 means no agreement and the highest value 1 means complete agreement

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{10}$$

where p_o is the accuracy of the model (Equation (7)) and p_e is the hypothetical probability of chance agreement, computed as

$$p_e = \frac{(TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{(TP + TN + FP + FN)^2}$$

All these performance metrics are computed for each deep learning architecture and the results are summarized in Table 3. The results reveal that the proposed malaria detection algorithm performs better than the compared deep learning models. The proposed method achieves more than 0.96 accuracy and F_1 score, and more than 0.93 score in both MCC and κ. The malaria detector based on the VGG19 architecture also achieves appreciative results with an F_1 and κ score of 0.9592 and 0.9185, respectively.

Table 3. Performance comparison of the proposed method and other deep learning models on NIH Malaria dataset. The best results in each metric are marked in bold.

Model	Specificity	Sensitivity	Precision	Accuracy	F_1 Score	MCC	κ
VGG-16	0.9632	0.9538	0.9585	0.9585	0.9585	0.9170	0.9170
VGG-19	0.9655	0.9529	0.9593	0.9592	0.9592	0.9185	0.9185
Xception	0.9753	0.9255	0.9508	0.9494	0.9494	0.9002	0.8989
Densenet-121	0.9526	0.9378	0.9453	0.9452	0.9452	0.8905	0.8904
Densenet-169	0.9436	0.9327	0.9382	0.9382	0.9382	0.8764	0.8764
Densenet-201	0.8769	0.9399	0.9079	0.9054	0.9052	0.8132	0.8106
Inception_v3	0.9313	0.9297	0.9306	0.9306	0.9306	0.8611	0.8611
Inc. Resnet_v2	0.9662	0.9556	0.9539	0.9539	0.9539	0.9179	0.9179
Resnet-50	0.9727	0.9237	0.9536	0.9517	0.9617	0.9054	0.9035
Resnet-101	0.9710	0.9419	0.9566	0.9562	0.9562	0.9129	0.9124
Resnet-152	0.9726	0.9215	0.9525	0.9505	0.9505	0.9031	0.9011
Squeeze Net	0.9562	0.9311	0.9438	0.9435	0.9435	0.8874	0.8871
Proposed	**0.9778**	**0.9633**	**0.9682**	**0.9682**	**0.9682**	**0.9364**	**0.9364**

5.5. Performance Comparison with Existing Methods

We also compare the performance of the proposed CNN model with other malaria detectors to evaluate its effectiveness. The compared methods include state-of-the-art and widely accepted automated malaria detection algorithms, including [11,12,51–56]. The malaria detection algorithm proposed by Fatima et al. [11] uses adaptive thresholding technique with a morphological image-processing tools. Specifically, object contours and the eight-connected rule are exploited to confirm the existence of malaria parasites in the cell. The malaria detector presented by Das et al. in [12] uses a marker-controlled

watershed approach to segment the erythrocytes. Various features describing the shape, size, and texture characteristics of segmented erythrocytes are computed and classified with a support vector machine. In Hung [51] algorithm, a faster, region-based convolutional neural network (Faster R-CNN) is proposed for malaria parasite detection. The model is pre-trained on ImageNet and is fine-tuned on the malaria dataset. Sanchez [55] also uses a deep convolutional neural network for malaria parasite detection. Pan et al. [52] proposed a deep convolutional neural-network-based algorithm for malaria detection. It uses Otsu's method and morphological operations for RBC segmentation, and LeNet-5 [81]-based CNN architecture is used for cell classification. A transfer learning based approach is presented by Vijayalakshmi et al. [53] for identifying malaria-infected cells in microscopic blood images. The presented model is a unification of VGG and support vector machine. The malaria detection algorithm introduced by Rajaraman et al. [54] consists of a three-layer convolutional neural network. It is a fully connected sequential architecture. A deep belief network-based malaria detection approach is proposed by Bibin et al. [56].

The performance achieved by the proposed and the compared methods is presented Table 4. The datasets used in some of the compared methods were different from the test dataset, and therefore such a comparison might not be just; nonetheless, it can help to assess the effectiveness of the proposed method. The results show that our method performs better than the compared methods in all performance metrics except sensitivity, where the Das [12] algorithm performs better; however, its specificity and accuracy are considerably lower than the proposed method. The performance results of Rajaraman, Fatima, and proposed algorithms are computed over the same dataset. These statistics show that the proposed method performs the best, followed by the Rajaraman method, which achieves a good detection rate.

The results presented in Table 4 indicate that the proposed deep malaria detector is an efficient and reliable tool to test microscopic blood smears for *Plasmodium* parasite infection. The proposed method exploits the bilateral filtering to suppress the noise in the images and provide better quality images for model training. Moreover, the data augmentation techniques allow variations in the dataset that helps in improving the performance and also generalizes the model to avoid over-fitting. It is also worth noting that the computational cost of the proposed method is reduced by using a 3×3 convolution filter for all convolutional layers, which eventually reduces the number of input channels. The deep learning models discussed in Table 4 have at least 16 layers and a large number of training parameters. Contrastingly, the proposed CNN architecture has only 8 layers and requires fewer training parameters compared to other deep learning models.

Table 4. Performance comparison of the proposed and the compared methods. Size represents the number of images in the dataset. The best results are marked in bold.

Method	Dataset	Size	Specificity	Sensitivity	Precision	Accuracy	F_1 Score
Das [12]	Self collected	–	0.6890	**0.9810**	–	0.8400	–
Sanchez [55]	–	–	–	–	0.8927	–	0.9530
Hung [51]	Self collected	1300	0.8519	0.7766	0.7804	0.8215	0.7784
Bibin [56]	Self collected	630	0.9590	0.9760	–	0.9630	0.8960
Pan [52]	PEIR-VM	24,648	0.8273	0.7402	0.7439	0.7921	0.7420
Rajaraman [54]	NIH dataset	27,558	0.9720	0.9470	–	0.9590	0.9590
Vijayalakshmi [53]	Self collected	2550	0.9292	0.9344	0.8995	0.9313	0.9166
Fatima [11]	NIH dataset	27,558	0.9500	0.8860	0.9466	0.9180	0.9153
Proposed	NIH dataset	27,558	**0.9778**	0.9633	**0.9682**	**0.9682**	**0.9682**

5.6. Time Complexity Analysis

The training time of deep learning models increases with the number of parameters and weights. We observe the training time of the deep learning models used in our study

for 25 epochs to evaluate their time complexity; the results are shown in Figure 7a. We recall that 173,700 images of the test dataset have been used in training. The results show that our model takes around 25 min for training, which is less than all the compared models. The reason for this computational efficiency is its small number of convolutional layers with a small number of parameters that eventually require less time for training. The VGG-19 and Inception models exhibit the best training times amongst the rest of the models. We also evaluated the testing time of compared models for complete testing data containing 8272 images, the results are presented in Figure 7b. The proposed model takes 5 s for a complete testing dataset, less than the other compared models. Among the rest of the models, the SqueezeNet model takes 6 s for testing, and it took around 30 min for training.
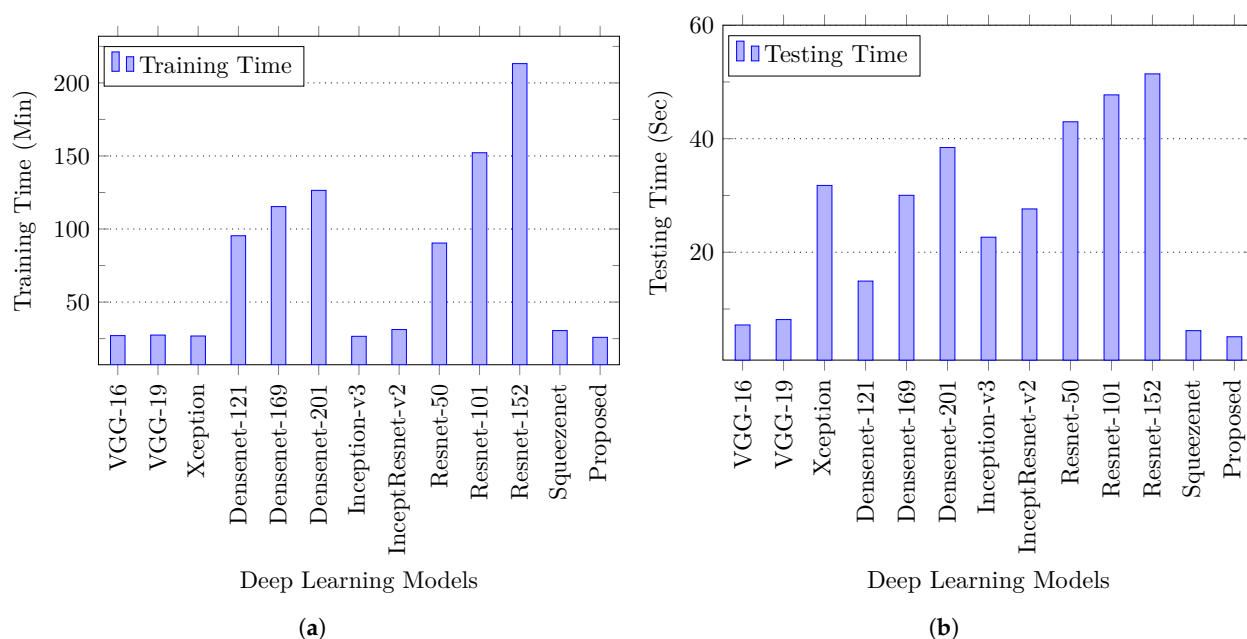

Figure 7. Execution time analysis: (**a**) Training time comparison of deep learning models for malaria detection., (**b**) Testing time comparison of deep learning models for malaria detection. The reported time is the time the model takes for the whole test dataset (8272 images).

6. Conclusions

In this paper, we proposed a deep learning solution for automated malaria detection from microscopic blood smears. The proposed CNN model exploits the bilateral filtering to remove noise from images and uses image augmentation techniques to achieve generalization. We also evaluated different deep learning models for malaria detection problems and compared their performance with the proposed method. The performance of the proposed method is also compared with the existing automated malaria detectors. The experimental evaluations performed on a benchmark dataset show that the proposed method performs better than other deep learning models. The proposed method also outperforms the compared malaria detection algorithms by achieving more than 0.96 accuracy and F_1 score. A time complexity analysis shows that our method is computationally efficient.

References

1. World Health Organization *Malaria Microscopy Quality Assurance Manual-Version 2*; World Health Organization: Geneva, Switzerland, 2016.
2. Moody, A. Rapid diagnostic tests for malaria parasites. *Clin. Microbiol. Rev.* **2002**, *15*, 66–78. [CrossRef]
3. Bosco, A.B.; Nankabirwa, J.I.; Yeka, A.; Nsobya, S.; Gresty, K.; Anderson, K.; Mbaka, P.; Prosser, C.; Smith, D.; Opigo, J. et al. Limitations of rapid diagnostic tests in malaria surveys in areas with varied transmission intensity in Uganda 2017–2019: Implications for selection and use of HRP2 RDTs. *PLoS ONE* **2021**, *15*, e24445. [CrossRef]
4. Grabias, B.; Kumar, S. Adverse neuropsychiatric effects of antimalarial drugs. *Expert Opin. Drug Saf.* **2016**, *15*, 903–910. [CrossRef] [PubMed]
5. Frean, J. Microscopic determination of malaria parasite load: Role of image analysis. *Microsc. Sci. Technol. Appl. Educ.* **2010**, 862–866.
6. Moon, S.; Lee, S.; Kim, H.; Freitas-Junior, L.H.; Kang, M.; Ayong, L.; Hansen, M.A.E. An Image Analysis Algorithm for Malaria Parasite Stage Classification and Viability Quantification. *PLoS ONE* **2013**, *8*, e61812. [CrossRef]
7. Maity, M.; Maity, A.K.; Dutta, P.K.; Chakraborty, C. A web-accessible framework for automated storage with compression and textural classification of malaria parasite images. *Int. J. Comput. Appl.* **2012**, *52*, 31–39. [CrossRef]
8. Jan, Z.; Khan, A.; Sajjad, M.; Muhammad, K.; Rho, S.; Mehmood, I. A review on automated diagnosis of malaria parasite in microscopic blood smears images. *Multimed. Tools Appl.* **2018**, *77*, 9801–9826. [CrossRef]
9. Poostchi, M.; Silamut, K.; Maude, R.J.; Jaeger, S.; Thoma, G. Image analysis and machine learning for detecting malaria. *Transl. Res.* **2018**, *194*, 36–55. [CrossRef] [PubMed]
10. Manzo, M.; Pellino, S. Bucket of Deep Transfer Learning Features and Classification Models for Melanoma Detection. *J. Imaging* **2020**, *6*, 129. [CrossRef]
11. Fatima, T.; Farid, M.S. Automatic detection of Plasmodium parasites from microscopic blood images. *J. Parasit. Dis.* **2020**, *44*, 69–78. [CrossRef]
12. Das, D.K.; Ghosh, M.; Pal, M.; Maiti, A.K.; Chakraborty, C. Machine learning approach for automated screening of malaria parasite using light microscopic images. *Micron* **2013**, *45*, 97–106. [CrossRef]
13. Di Ruberto, C.; Dempster, A.; Khan, S.; Jarra, B. Analysis of infected blood cell images using morphological operators. *Image Vis. Comput.* **2002**, *20*, 133–146. [CrossRef]
14. Arco, J.E.; Górriz, J.M.; Ramírez, J.; Álvarez, I.; Puntonet, C.G. Digital image analysis for automatic enumeration of malaria parasites using morphological operations. *Expert Syst. Appl.* **2015**, *42*, 3041–3047. [CrossRef]
15. Tek, F.B.; Dempster, A.G.; Kale, I. Malaria parasite detection in peripheral blood images. In Proceedings of the British Machine Vision Conference 2006, Edinburgh, UK, 4–7 September 2006; BMVA: Edinburgh, UK, 2006; pp. 347–356.
16. Díaz, G.; González, F.A.; Romero, E. A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *J. Biomed. Inform.* **2009**, *42*, 296–307. [CrossRef]
17. Savkare, S.S.; Narote, S.P. Automated system for malaria parasite identification. In Proceedings of the 2015 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, India, 15–17 January 2015; pp. 1–4.
18. Rakshit, P.; Bhowmik, K. Detection of presence of parasites in human RBC in case of diagnosing malaria using image processing. In Proceedings of the IEEE Second International Conference on Image Information Processing (ICIIP-2013), Shimla, India, 9–11 December 2013; pp. 329–334.
19. Purwar, Y.; Shah, S.L.; Clarke, G.; Almugairi, A.; Muehlenbachs, A. Automated and unsupervised detection of malarial parasites in microscopic images. *Malar. J.* **2011**, *10*, 364. [CrossRef]
20. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 898–916. [CrossRef]
21. Savkare, S.; Narote, S. Automatic system for classification of erythrocytes infected with malaria and identification of parasite's life stage. *Proc. Technol.* **2012**, *6*, 405–410. [CrossRef]

22. Damahe, L.B.; Krishna, R.; Janwe, N.; Thakur, N. Segmentation based approach to detect parasites and RBCs in blood cell images. *Int. J. Comput. Sci. Appl.* **2011**, *4*, 71–81.

23. Chakrabortya, K.; Chattopadhyayb, A.; Chakrabarti, A.; Acharyad, T.; Dasguptae, A.K. A combined algorithm for malaria detection from thick smear blood slides. *J. Health Med. Inf.* **2015**, *6*, 645–652. [CrossRef]

24. Ghosh, M.; Das, D.; Chakraborty, C.; Ray, A.K. Plasmodium vivax segmentation using modified fuzzy divergence. In Proceedings of the 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011; pp. 1–5.

25. Chayadevi, M.; Raju, G. Automated colour segmentation of malaria parasite with fuzzy and fractal methods. In *Computational Intelligence in Data Mining-Volume 3*; Springer: New Delhi, India, 2015; Volume 33, pp. 53–63.

26. Zhang, Z.; Ong, L.L.S.; Fang, K.; Matthew, A.; Dauwels, J.; Dao, M.; Asada, H. Image classification of unlabeled malaria parasites in red blood cells. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 August 2016; pp. 3981–3984.

27. Aimi Salihah, A.N.; Yusoff, M.; Zeehaida, M. Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering. *WSEAS Trans. Biol. Biomed* **2013**, *10*.

28. Bhowmick, S.; Das, D.K.; Maiti, A.K.; Chakraborty, C. Structural and textural classification of erythrocytes in anaemic cases: A scanning electron microscopic study. *Micron* **2013**, *44*, 384–394. [CrossRef]

29. Devi, S.S.; Sheikh, S.A.; Talukdar, A.; Laskar, R.H. Malaria infected erythrocyte classification based on the histogram features using microscopic images of thin blood smear. *Ind. J. Sci. Technol.* **2016**, *9*, 1–10. [CrossRef]

30. Mandal, S.; Kumar, A.; Chatterjee, J.; Manjunatha, M.; Ray, A.K. Segmentation of blood smear images using normalized cuts for detection of malarial parasites. In Proceedings of the Annual IEEE India Conference (INDICON), Kolkata, India, 17–19 December 2010; pp. 1–4.

31. Kareem, S.; Kale, I.; Morling, R.C.S. Automated P.falciparum Detection System for Post-Treatment Malaria Diagnosis Using Modified Annular Ring Ratio Method. In Proceedings of the 2012 UKSim 14th International Conference on Computer Modelling and Simulation, Cambridge, UK, 28–30 March 2012; pp. 432–436.

32. Kareem, S.; Kale, I.; Morling, R.C.S. Automated malaria parasite detection in thin blood films: A hybrid illumination and color constancy insensitive, morphological approach. In Proceedings of the 2012 IEEE Asia Pacific Conference on Circuits and Systems, Kaohsiung, Taiwan, 2–5 December 2012; pp. 240–243.

33. Maitethia Memeu, D.; Kaduki, K.; Mjomba, A.; Muriuki, N.; Gitonga, L. Detection of plasmodium parasites from images of thin blood smears. *Open J. Clin. Diagn.* **2013**, *03*, 183–194. [CrossRef]

34. Abdul Nasir, A.S.; Mashor, M.Y.; Mohamed, Z. Segmentation based approach for detection of malaria parasites using moving k-means clustering. In Proceedings of the 2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences, Langkawi, Malaysia, 17–19 December 2012; pp. 653–658.

35. Das, D.; Ghosh, M.; Chakraborty, C.; Maiti, A.K.; Pal, M. Probabilistic prediction of malaria using morphological and textural information. In Proceedings of the 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011; pp. 1–6.

36. Linder, N.; Turkki, R.; Walliander, M.; Mårtensson, A.; Diwan, V.; Rahtu, E.; Pietikäinen, M.; Lundin, M.; Lundin, J. A malaria diagnostic tool based on computer vision screening and visualization of Plasmodium falciparum candidate areas in digitized blood smears. *PLoS ONE* **2014**, *9*, e104855. [CrossRef] [PubMed]

37. Muralidharan, V.; Dong, Y.; David Pan, W. A comparison of feature selection methods for machine learning based automatic malarial cell recognition in wholeslide images. In Proceedings of the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), Las Vegas, NV, USA, 24–27 February 2016; pp. 216–219.

38. Suwalka, I.; Sanadhya, A.; Mathur, A.; Chouhan, M.S. Identify malaria parasite using pattern recognition technique. In Proceedings of the 2012 International Conference on Computing, Communication and Applications, Dindigul, India, 22–24 February 2012; pp. 1–4.

39. Chavan, S.N.; Sutkar, A.M. Malaria disease identification and analysis using image processing. *Int. J. Latest Trends Eng. Technol.* **2014**, *3*, 218–223.

40. Ghate, D.A.; Jadhav, C.; Rani, N. Automatic detection of malaria parasite from blood images. *Int. J. Comput. Sci. Appl.* **2012**, *1*, 66–71.

41. Suryawanshi, M.S.; Dixit, V. Improved technique for detection of malaria parasites within the blood cell images. *Int. J. Sci. Eng. Res.* **2013**, *4*, 373–375.

42. Annaldas, M.S.; Shirgan, S.; Marathe, V. Enhanced identification of malaria parasite using different classification algorithms in thick film blood images. *Int. J. Res. Advent Technol.* **2014**, *2*, 16–20.

43. Tsai, M.H.; Yu, S.S.; Chan, Y.K.; Jen, C.C. Blood smear image based malaria parasite and infected-erythrocyte detection and segmentation. *J. Med. Syst.* **2015**, *39*, 118. [CrossRef]

44. Poostchi, M.; Ersoy, I.; McMenamin, K.; Gordon, E.; Palaniappan, N.; Pierce, S.; Maude, R.J.; Bansal, A.; Srinivasan, P.; Miller, L.; et al. Malaria parasite detection and cell counting for human and mouse using thin blood smear microscopy. *J. Med. Imaging* **2018**, *5*, 044506. [CrossRef]

45. Dong, Y.; Jiang, Z.; Shen, H.; David Pan, W.; Williams, L.A.; Reddy, V.V.B.; Benjamin, W.H.; Bryan, A.W. Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In Proceedings of the 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), Orlando, FL, USA, 16–19 February 2017; pp. 101–104.

46. Quinn, J.A.; Nakasi, R.; Mugagga, P.K.; Byanyima, P.; Lubega, W.; Andama, A. Deep convolutional neural networks for microscopy-based point of care diagnostics. In Proceedings of the Machine Learning for Healthcare Conference, Los Angeles, CA, USA, 19–20 August 2016; pp. 271–281.

47. Xie, W.; Noble, J.A.; Zisserman, A. Microscopy cell counting and detection with fully convolutional regression networks. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2018**, *6*, 283–292. [CrossRef]

48. Gopakumar, G.P.; Swetha, M.; Sai Siva, G.; Sai Subrahmanyam, G.R.K. Convolutional neural network-based malaria diagnosis from focus stack of blood smear images acquired using custom-built slide scanner. *J. Biophotonics* **2018**, *11*, e201700003. [CrossRef]

49. Liang, Z.; Powell, A.; Ersoy, I.; Poostchi, M.; Silamut, K.; Palaniappan, K.; Guo, P.; Hossain, M.A.; Sameer, A.; Maude, R.J.; et al. CNN-based image analysis for malaria diagnosis. In Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shenzhen, China, 15–18 December 2016; pp. 493–496.

50. Pattanaik, P.A.; Wang, Z.; Horain, P. Deep CNN frameworks comparison for malaria diagnosis. In Proceedings of the IMVIP 2019 Irish Machine Vision and Image Processing Conference, Dublin, Ireland, 28–30 August 2019.

51. Hung, J.; Carpenter, A. Applying faster R-CNN for object detection on malaria images. In Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 56–61.

52. Pan, W.D.; Dong, Y.; Wu, D. Classification of malaria-infected cells using deep convolutional neural networks. In *Machine Learning—Advanced Techniques and Emerging Applications*; Intech Open: London, UK, 2018; Volume 159.

53. Vijayalakshmi, A. Deep learning approach to detect malaria from microscopic images. *Multimed. Tools Appl.* **2019**, *79*, 15297–15317. [CrossRef]

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

55. Sánchez, C.S. Deep Learning for Identifying Malaria Parasites in Images. Master's Thesis, University of Edinburgh, Edinburgh, UK, 2015.

56. Bibin, D.; Nair, M.S.; Punitha, P. Malaria parasite detection from peripheral blood smear images using deep belief networks. *IEEE Access* **2017**, *5*, 9099–9108. [CrossRef]

57. Rahman, A.; Zunair, H.; Rahman, M.S.; Yuki, J.Q.; Biswas, S.; Alam, M.A.; Alam, N.B.; Mahdy, M. Improving Malaria Parasite Detection from Red Blood Cell using Deep Convolutional Neural Networks. *arXiv* **2019**, arXiv:1907.10418.

58. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

59. Sharma, N.; Jain, V.; Mishra, A. An analysis of convolutional neural networks for image classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [CrossRef]

60. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

61. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

62. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.

63. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]

64. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.

65. Zhu, Y.; Newsam, S. DenseNet for dense flow. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 790–794.

66. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.

67. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.

68. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.

69. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]

70. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.

71. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the Sixth International Conference on Computer Vision (ICCV '98), Bombay, India, 4–7 January 1998; IEEE Computer Society: Washington, DC, USA, 1998; p. 839.

72. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

73. Wang, J.; Perez, L. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.

74. Chollet, F. Building powerful image classification models using very little data. *Keras Blog*, 5 June 2016.

75. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]

76. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]

77. Matthews, B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta Protein Struct.* **1975**, *405*, 442–451. [CrossRef]

78. Sasaki, Y. The truth of the F-measure. *Teach Tutor Mater* **2007**, *1*, 1–5.

79. Farid, M.S.; Lucenteforte, M.; Grangetto, M. DOST: A distributed object segmentation tool. *Multimed. Tools Appl.* **2018**, *77*, 20839–20862. [CrossRef]

80. Chicco, D.; Jurman, G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 16. [CrossRef]

81. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]