Malaria Detection Using Deep Leaning

A thesis submitted in partial fulfillment of the requirements for

the award of degree of

Master of Science

in

Computer Science

by

Saikat Mondal

(20419CMP020)

Department of Computer Science

Institute of Science

Banaras Hindu University,Varanasi-221005

July 2022

DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **Malaria Detection Using Deep Leaning**,in partial fulfillment of the requirement for the award of the Degree of Master of Science and submitted to the Banaras Hindu University is an authentic record of my own work carried out during the period January-2022 to July-2022 under the supervision of Dr. Marisha . I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any Institution.

Saikat Mondal

Varanasi - 221005

5-07-2022

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge. The Viva-Voce examination of Saikat Mondal M.Sc. Student has been held on _____.

Dr. Marisha
(Project Guide)
Assistant Professor
Dept.of Computer Science
Banaras Hindu University
Varanasi, Uttar Pradesh

Dr. Vivek Kumar Singh
(Head of the Department)
Professor and Head
Dept.of Computer Science
Banaras Hindu University
Varanasi, Uttar Pradesh

Abstract

Malaria is a serious public health problem that has an impact on people all around the world. Malaria is a mosquito-borne disease carried by female Anopheles mosquitos, which transfers a motile infective form to the host body, such as humans, and reproduces asexually in the blood cells of the host. Malaria symptoms include fever, headache, exhaustion, and vomiting. In severe cases, coma and death are possible outcomes. Under a microscope, lab or qualified workers check human blood smears for parasite-infected red blood cells as a standard technique of diagnosing malaria. The diagnosis is based on the examiner's expertise and experience. This process is inefficient, and the diagnosis is reliant on the examiner's knowledge and competence. Incorrect diagnoses occur due to a scarcity of competent professionals and a lack of appropriate equipment and infrastructure, resulting in an increase in death rates.

Early studies showed detection using Machine Learning approaches such as Support Vector Machine (SVM), which is time-consuming and needs hand-engineered feature extraction to train data, and the results were insufficient. The suggested technique in this study displays a system with end-to-end automated models that do both feature extraction and classification utilizing blood smear cell pictures, using a deep neural network.

The Malaria Dataset from the National Institutes of Health (NIH) was used for this study. In this study, we present a new deep learning model based on convolutional neural networks (CNN) that recognizes and predicts infectious cells in thin blood smears on traditional microscope slides and also applied the transfer Learning technique for the detection of malaria cells. The highest performing model on the malaria dataset is analyzed and compared using two Transfer Learning approach, VGG-19 and ResNet-50. To improve the model's performance, preprocessing procedures such as data segmentation and normalization are used.

Keyword: Malaria Detection, Deep Learning, CNN, VGG-16, VGG-19, Resnet-50

Acknowledgement

I feel beholden to heartily acknowledge my deep sense of obligation to persons and institutions wherefrom I received help during the project. First of all, I would like to express my sincere gratitude to my learned supervisor Dr. Marisha, a lady of broad vision, creative thinking, and lover of academic pursuit, for rendering valuable guidance throughout the project and constantly motivating me to work harder during the project.

I am grateful to all the faculties who have directly or indirectly helped in the completion of the project. I take this opportunity to express my profound gratitude to all my seniors, classmates, and friends, who supported and encouraged me from the beginning of this project till completion.

Last but not the least, I express thanks to my family for cooperating and assisting me from time to time, without them this task could not have completed. Further acknowledge that no perfection can be claimed by any human being. I take responsibility for any error which inadvertently might have crept in.

Saikat Mondal

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CAD	Computer Aided Detection
ANN	Artificial Nural Network
ML	Machine Learning
GLCM	Gray-Level Co-Occurrence Matrix
RBC	Red Blood Cell
FC	Fully connected layer
TP	True Positive
TN	True Negative
FP	False positive
FN	False negative

Chapter 1

Introduction

1.1 GENERAL

Malaria is caused by the plasmodium falciparum virus, which was first discovered in Africa. Each year, an estimated 300–500 million individuals worldwide suffer from malaria, resulting in 1.5–2.7 million fatalities [1]. The report of World Health Organization (WHO) claims that, malaria was diagnosed in roughly 219 million cases worldwide in 2017, resulting in 435,000 fatalities [2]. Malaria is a dangerous disease that is more common in rural places where medical diagnosis and health care are difficult to come by.

Practitioners use the internationally approved light microscopy approach to diagnose malaria. For malaria diagnosis, thick and thin stained blood smears are used in traditional light microscopy. The fundamental disadvantage of this strategy is that the accuracy of the diagnosis is dependent on the microscopists' skill and understanding. As a result, correct diagnosis is delayed. Most of the time, diagnoses are determined only based on clinical signs and symptoms, which is prone to inaccuracy. Inaccurate diagnoses result in higher mortality, treatment resistance, and financial costs.

By preventing severe malaria cases, early and correct malaria diagnosis and treatment can cure a patient and save many lives. Malaria prevention and treatment are not available to around 5 million individuals worldwide. As a result, a dependable alternative is required to help give routine access to high-quality diagnosis, which is currently unavailable.

The focus of this research is to develop an accurate malaria diagnosis model that can be deployed without the need for professional technicians, as well as to test the model's correctness to provide high-quality findings. The most fundamental disadvantage of the widely accepted

microscopy approach in general, reliance on human specialists for diagnostic accuracy of the results, might be removed with automated image analysis software. Using the knowledge, practice, and implementation of traditional approaches to achieving rapid and efficient results is what automating the detection process entails.

1.2 Computer Aided Detection (CAD)

In medical imaging, computer-assisted detection (CAD) and diagnosis have acquired a lot of attention. Detection and diagnosis tools provide clinicians with a crucial second opinion and aid in the screening process. The performance of several machine learning and deep learning algorithms for detecting malaria on thin blood smeary images is proposed and studied in this work. Plasmodium malaria is a parasitic protozoan that causes malaria in people, and CAD of Plasmodium on thin blood smeary pictures would help microscopists and improve their productivity. In the field of medical imaging, several machine learning and deep learning algorithms for CAD have been proposed in the literature.

1.2.1 Deep Learning for Malaria diagnosis

Many studies on various machine learning models have been undertaken to automate the diagnosing process. Tuning, factor analysis, and feature engineering are all required for machine learning models. With more data, the machine learning process is not scalable. Machine learning necessitates feature engineering and feature training, both of which are time-consuming tasks. Deep learning models are more accurate and scalable than traditional models. Convolutional Neural Networks (CNN) have demonstrated their effectiveness in a wide range of computer vision tasks.Convolutional Neural Networks convert the volume of an input image into an output volume with a class label.

There is no requirement for handcrafted feature extraction or feature engineering in Convolutional modeling. Images are fed into CNN models, and the images are processed via many layers of the CNN model, which are explored in more detail later. As a result, the Convolutional Neural Network is the best method for diagnosing Malaria from patient blood cell pictures.

In this research, Basic CNN, VGG-16 Frozen CNN, VGG-19 Frozen CNN, VGG19 fine-tuned CNN, Fozen ResNet-50, and ResNet Fine-tuned CNN models are experimented on

malaria dataset to analyze and compare the best performing model.

1.3 Transfer Learning

In a machine or deep learning, the practice of reusing previous pre-trained models to address fresh challenges is known as transfer learning. Previous training knowledge is reused to aid in the completion of a new task. To adapt to new unknown data, the initial trained model generally necessitates a high level of generalization. The amount of resources and labeled data required to train new models is decreased using this approach to machine learning development. It is increasingly being used as a tool in the development process and is becoming an important part of machine learning and deep learning.

It is anticipated that several strong pre-trained models would make effective transfer learning possible in the context of feature extraction. Examples of image classification models that have been trained on the ImageNet dataset, which comprises millions of data points and 1000 classes, include Oxford VGG and Microsoft ResNet. These models have been taught to identify low-level features including edges, corners, ridges, and blobs as well as to carry out a number of common tasks. An individually built model might not be able to learn to do some jobs as well as the pre-trained models due to a much smaller training data set. Transfer learning may be very helpful when training time and data availability are constrained.

1.4 Objective

This project aims to develop a Convolutional Neural Network model to categorize images of human blood smears as Malaria Parasitic vs. Healthy and employ several transfer learning approaches. On the basis of the accuracy graph and loss graph, we also examined how well these models performed. This study also examines how model accuracy is affected by preprocessing techniques.

With the use of machine learning techniques, substantial research has been done to automatically classify malaria cells. Machine learning has been used successfully by many academics to conduct studies on the identification of malaria, but as technology develops and more data becomes accessible, a new branch of artificial intelligence known as deep learning is gaining ground. Deep learning is a type of machine learning that helps with decision-making

by simulating the processing and pattern-making functions of the human brain. In this work, we construct a reliable malaria diagnosis model that can be used without the assistance of trained professionals and verify the model's accuracy to produce reliable results.

Chapter 2

Literature Review

2.1 General

While going through different papers it is seen how machine learning algorithms efficiently performs classification task. Before the introduction to deep learning, the task of classification through machine learning involves a sequence of steps like Preprocessing, Erythrocyte segmentation, Feature extraction, and Classification. Convolution neural network simplifies the entire challenge of the hand-crafted feature Extraction process, we need not go through the sequence of steps like Machine learning, CNN automates all the steps.
Machine learning (ML) and Deep learning algorithms applied to microscopic blood smear pictures in computer-aided diagnostic tools have the potential to reduce clinical burden by assisting with triage and disease diagnosis. To enhance clinical decision-making, these systems analyze medical images for common appearances and highlight pathological characteristics. As a result of these factors, computer-aided diagnostic tools have grown in popularity in image-based on the medical diagnosis and risk assessment.

2.2 Machine Learning Based Approaches

Prior to the Deep Learning introduction, Image acquisition, preprocessing, erythrocyte segmentation, feature extraction, and classification are all part of the machine learning based identification and classification of malaria parasite cells [3].

The process of acquiring malaria blood smear images from a digital camera coupled with a light microscope is known as image acquisition. Preprocessing eliminates undesired

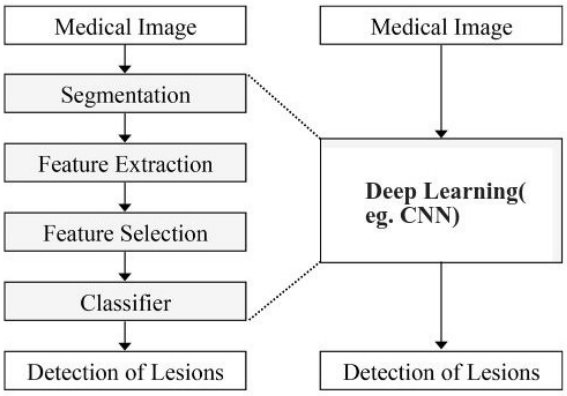


Figure 2.1: Deep Learning Approach Like CNN incorporates Different Machine learning tasks.

noise from the acquired malaria parasite picture, allowing for better viewing and analysis [6]. Segmentation is the process of extracting red blood cells (erythrocytes) from a preprocessed image and removing additional information such as white blood cells, platelets, and artifacts. Color, shape, geometry, and texture are examples of features extracted from erythrocytes. Using machine learning methods, extracted attributes are categorized as infected or non-infected erythrocytes. Generally, thick smear images are used to detect malaria infection, while thin smear images are utilized to determine the kind of malaria species and its phases [5].

2.2.1 Preprocessing

Several methods for improving the spatial quality of blood smear pictures have been suggested. To inverse image blurring, May et al. used the Median filter to reduce impulse noise and the Wiener filter to reduce additive noise [5]. To eliminate Gaussian noise from the pictures, Arco et al. employed a Gaussian filter [7]. To reduce Gaussian noise and preserve edges in microscopic images, Das et al. [3] utilized a Geometric mean filter. For picture smoothening and edge improvement, Savkare et al. utilized the Laplacian filter [8]. The Susan filter was used by Soni et al. to retain picture structure and improve image quality [9]. By averaging the intensity of the pixels, Diaz et al. used a low pass filter to eliminate the high-frequency intensity in the picture. [4]

2.2.2 Segmentation

Erythrocyte segmentation is a subjective topic, with the majority of studies focusing on a range of segmentation methods. The watershed approach was used by Savkare et al. [8] to

find overlapping cells on each linked component. To discriminate between foreground and background erythrocyte pictures, Suradkar employed an edge detection algorithm [10]. The granulometry technique was utilized by Soni et al. to efficiently detect regular-sized objects of interest [9]. The circle hough transformation was used by Ma et al. [13] to calculate the radius of the erythrocyte. Das et al. effectively located overlapping cells using a marker-controlled watershed segmentation approach [3]. By selecting the initial centroid and computing the intensity distance, Khan et al. used the K-means clustering technique to segment the Red Blood Cell (RBC) [14].

2.2.3 Feature extraction

Feature extraction is critical in distinguishing between infected and non-infected malaria. Das et al. identified haralick texture properties such as mean, standard deviation, entropy, correlation, energy, contrast, roughness, orientation, homogeneity, and Angular Second Movement using the Gray-Level Co-Occurrence Matrix (GLCM) (ASM). In order to extract shape characteristics, the author also assessed area, perimeter, and eccentricity [3]. Chavan and Sutkar used textural characteristics based on histograms to detect the malaria parasite [18]. Color histogram aspects were highlighted by Malihi et al. for recognizing infected erythrocytes [19]. Gray Level Run-Length Matrix (GLRLM) was utilized by Das et al. and Springl to detect texture characteristics based on Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray Level Non-uniformity (GLN), Run Length Non-uniformity (RLN), and Run Percentage (RP) [3] [20].

2.2.4 Supervised Learning

Many studies are using supervised learning algorithms to discriminate between infected and non-infected malaria depending on the data obtained. Using a multilayer perceptron network, Abu Seman et al. classified P.falciparum, P.vivax, and P.malariae with an accuracy of 89.80% [22]. Khot and Prasad utilized an artificial neural network to identify parasite infection based on morphological criteria, with a 73.57 percent accuracy [23]. Using the Naive Bayes technique, Das et al. classified P.vivax and P.falciparum ring and gametocyte phases with 96.73 percent accuracy [3]. Anggraini et al. used a Bayesian classifier to achieve 93.3 percent accuracy in distinguishing malaria parasites at various stages, including ring, gametocyte, and other artifacts [24].

The majority of computer-aided malaria detection methods include three primary phases of image processing, followed by a classifier. However, it has several drawbacks, which are described below;

- It is very important to know the domain knowledge for applying several preprocessing, segmentation, and feature extraction techniques. In terms of knowledge and time, this endeavor is prohibitively expensive. [6]
- We know that image preprocessing, segmentation, and feature extraction are worked sequentially, so errors in one stage will be transferred to the next, causing classification to degrade.
- Traditional image processing performance is heavily influenced by picture quality factors such as resolution, brightness, sharpness, and contrast [31].
- Most contemporary image-based algorithms use a shallow learning approach, and their performance will converge or remain stable at a given level as the amount of the training data grows. [32]
- Because image processing algorithms include handcrafted feature extracting techniques, important hidden features may be overlooked while interpreting the infected and non-infected malaria parasite.

2.3 Deep Learning Approaches

Data-driven deep learning (DL) techniques have outperformed the performance of manual feature extraction processes [5] by self-discovering attributes from raw pixel data and executing end-to-end feature extraction and classification. CNN, a type of deep learning model, has shown promise in image classification, identification, and localization. Figure 2.2 shows the basic Architecture of CNN model.

To address the aforementioned issues, we attempted to use convolution neural network techniques to detect malaria parasites for the following reasons:

- The entire difficulty of developing effective feature extraction and categorization of malaria is simplified by CNN. This incorporates image analysis filters in multi-stage

processing layers. [32]

- It's ideal for learning feature representation on a big domain-specific dataset and identifying hidden properties that human visual interpretation would otherwise overlook.
- By taking raw microscopic pictures as input and executing classification tasks with tremendous processing power, the CNN architecture allows end-to-end learning [31].

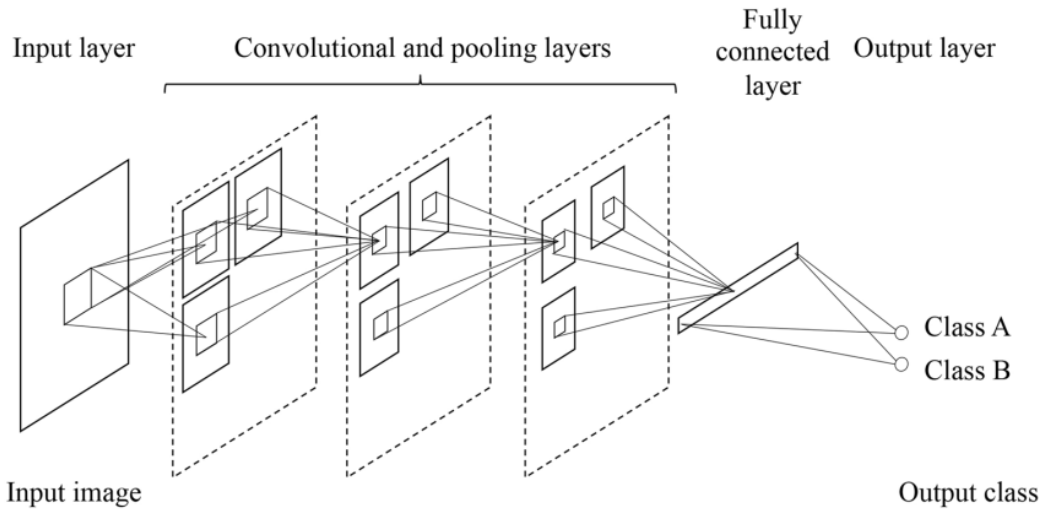


Figure 2.2: Basic_CNN

A literature study has indicated the use of conventional ML and data-driven DL techniques to the problem of malaria parasite detection in thin-blood smear pictures. Support vector machine (SVM) and convolutional neural networks (CNNs) are kernel-based algorithms that have been tested by Dong et al. [40] for their ability to distinguish between infected and healthy cells. Red blood cells that had been segmented were randomly split into a train, validation, and test groups. The CNN outperformed the SVM classifier, which had a classification accuracy of 92 percent, with a classification accuracy of over 95 percent. In order to automate diagnosis, the attributes were self-discovered by the CNNs from the raw pixel data. In the testing, the custom CNN outperformed the pre-trained AlexNet [41] model with an accuracy of 97.37 percent. The usefulness of a shallow deep belief network in recognizing parasites was examined by the authors of another study [42] using randomized splits with peripheral smear images. The results of the experiment showed that the deep belief network outperformed the SVM-based classification, which had an F-score of 78.44 percent, with an F-score of 89.66 percent. To find parasites in a targeted stack of slide images, a CNN model was modified [43]. They observed

that the customized CNN model outperformed the SVM classifier with a Matthews Correlation Coefficient (MCC) score of 98.77 percent.

2.3.1 TERMINOLOGIES USED

2.3.1.1 Convolution

Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm. Convolution is a specialized type of linear operation used for feature extraction, where a small array of numbers, called a kernel, is applied across the input, which is an array of numbers, called a tensor. An element-wise product between each element of the kernel and the input tensor is calculated at each location of the tensor and summed to obtain the output value in the corresponding position of the output tensor, called a feature map. This procedure is repeated applying multiple kernels to form an arbitrary number of feature maps, which represent different characteristics of the input tensors; different kernels can, thus, be considered as different feature extractors. Two key hyperparameters that define the convolution operation are size and number of kernels. The former is typically 3×3, but sometimes 5×5 or 7×7. The latter is arbitrary, and determines the depth of output feature maps. Figure 2.3 shown below gives the idea of convolution.

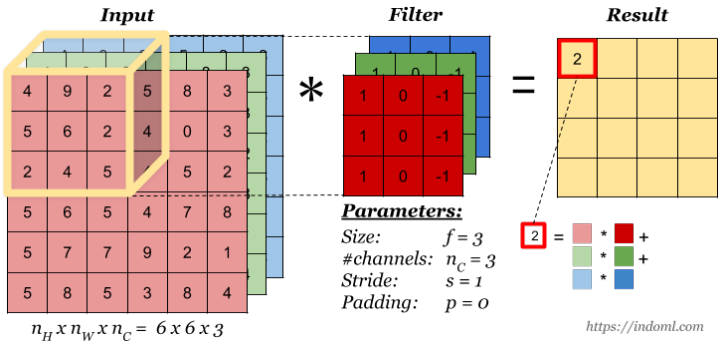


Figure 2.3: Convolution_Operation

For example, as shown in figure 2.3, The convolution layer having a 3x3x3 filter with no padding and one stride generates an output feature map of size 4x4x1 for an input of size 6x6x3. The region a filter covers at an instance in input is called its Receptive Field.

2.3.1.2 Padding and Strides

A convolution operation generates an output with the dimensions (n - f + 1) x (n - f + 1) from an input grayscale picture of size n x n and a kernel of size f x f. This suggests that with each convolution operation performed, the picture's dimensions decrease; hence, after a certain number of convolution operation executions, the image will vanish into nothing and the process of creating deeper networks would come to an end [36]. Additionally, the corner and edge pixels are frequently used less than the central pixels, increasing the possibility of missing certain prominent characteristics. Padding is used to get around this problem by adding zeros to the input's edges. The padding method moves the border pixels from the original picture's border to the center of the padded image, increasing their effect while maintaining all of the original image's information.

The stride [37] specifies how many pixels on the input matrix a kernel should slide. When the stride is set to 1, the filter will move one pixel at a time. The feature map gets smaller and the overlap between the receptive fields gets lower as the value of strides increases. As a result, the feature map finally becomes less than the initial input size.

2.3.1.3 Downsampling

Pooling layers are used to reduce the dimensions of the feature maps. It reduces the number of parameters that must be learned and the amount of network processing required. One of the most common pooling algorithms, known as max pooling, chooses the most elements possible from the feature map region that the filter covers. As a result, the output of the max-pooling layer would be a feature map that contained the most noticeable features from the prior feature map.

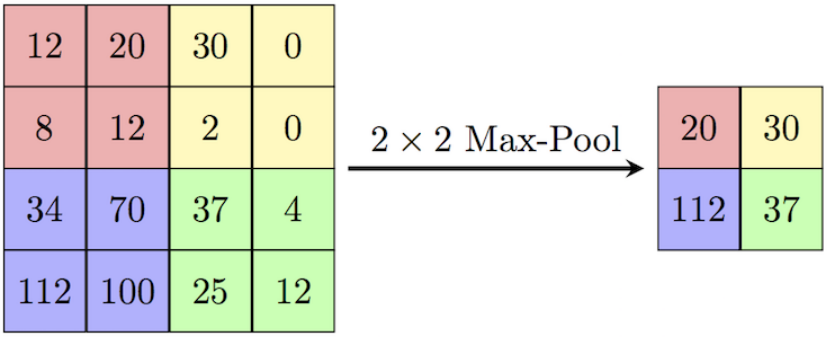


Figure 2.4: Max_pooling Operation

As we can see in Figure 2.4 above how Max-pooling works. first a filter size has decided which is in this case is 2x2 and stride mean how many pixels to jump for next operation which is in this case is 2x2 and output we have received are downsample feature map in right having having reduced size of feature map and containing only prominent pixels values or we can say in this case only largest value in corresponding subpart.

2.3.1.4 Activation Functions

In neural networks whether a neuron will activate or not depends on whether a neuron's input is important for the network or not. An activation function is a mathematical function which operates over input of neurons and it can be either linear or non-linear but most of the time for complex problems we use non-linear activation functions because this may not always be the case where a line can classify data in different classes using a straight line. In this work, the proposed models have used the ReLU activation function as shown in Figure 2.5. This monotonic, half rectified (from bottom) function ranges from 0 to infinity.

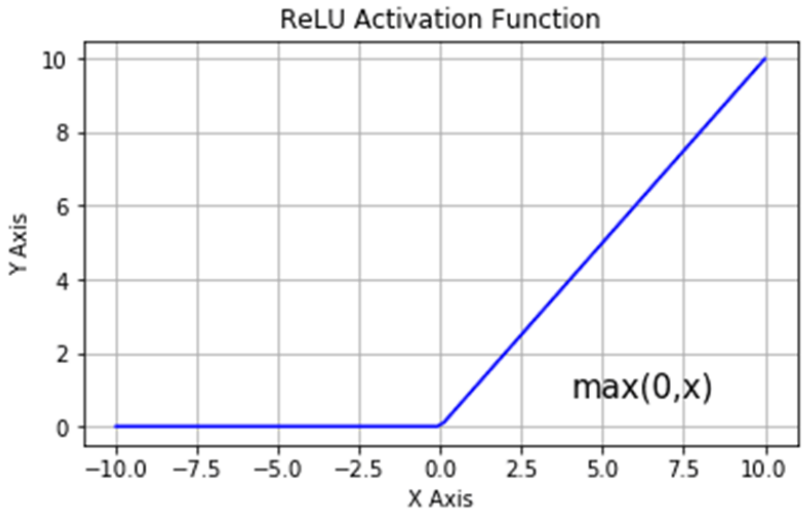


Figure 2.5: Max_pooling Operation

2.3.1.5 Loss Function

An objective function is used to assess a potential solution. We aim to minimize the error in the context of neural networks, and this function is known as a cost or loss function. This function calculates a value that it refers to as "loss." The method most frequently employed is the Cross-Entropy Loss, which evaluates each predicted probability against the actual class output value

and computes a score penalizing the probability depending on the deviation from the expected value.

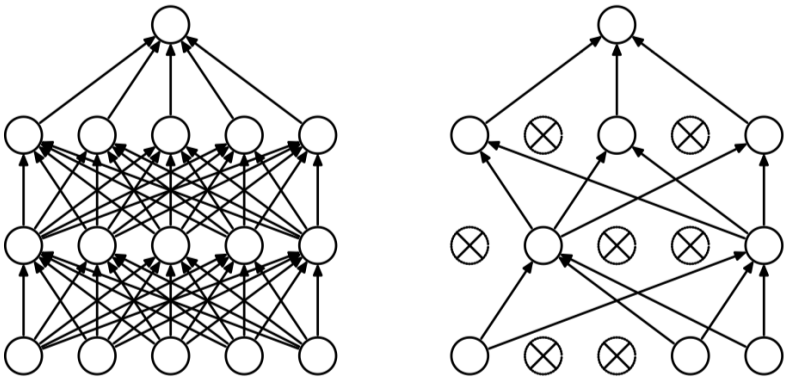
2.3.1.6 Optimization Algorithm

Using various techniques referred to as Optimizers, the parameters or features of a neural network, such as learning rate and weights, are improved in order to minimise losses. The Adam Optimizer, which derives individual adaptive learning rates for various parameters from estimations of the first and second moments of the gradients, is employed in this study [39]. Based on the uncentered variance, which is the average of the second gradient moments, the algorithm adjusts the learning rates of the parameter. An exponential moving average of the gradient and the squared gradient is calculated, and the beta1 and beta2 parameters determine how quickly these moving averages degrade. A bias of moments estimates is produced due to the initialization of the moving averages and beta1 and beta2 values close to 1.0, which can be overcome by calculation of the biased estimates first and further calculation of bias-corrected estimates.

2.3.1.7 Dropout

Dropout is a technique where randomly selected neurons are ignored during training. They ”drop-out” at random. This means that any weight changes are not applied to the neuron on the backward trip and that their effect on the activation of downstream neurons is temporally erased on the forward pass. As neural network learns, neuron weights settle into their context within the network. Neuron weights are adjusted for particular characteristics, resulting in some specialization. Neighboring neurons start to depend on this specialization, which, if it goes too far, might produce a fragile model that is overly dependent on the training data, can be dangerous. Drop out helps to overcome this problem.

As we can see in figure 2.6, on the left, a fully connected neural network and on the right, a neural network after applying dropout. We can see some neurons are deactivated after applying dropout.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.6: Dropout

2.3.1.8 Batch Normalization

The distribution of the inputs to each layer varies during training as the parameters of the preceding layers change, which makes it challenging to train Deep Neural Networks. This makes it extremely difficult to train models with saturating nonlinearities and slows down the training process by necessitating lower learning rates and careful parameter setup. We call this an internal covariate shift. By normalizing layer inputs this problem can be solved. We can employ significantly greater learning rates and less stringent initialization requirements thanks to batch normalization. Additionally, it serves as a regularizer, often doing away with the necessity for Dropout.

2.3.2 Transfer Learning

In most recent times researchers are interested in deploying different Transfer learning Techniques to analyze the characteristics of microorganisms from their biomedical images. Transfer learning procedures are used when there is a scarcity of data, such as in the case of medical imaging. Var et al. [33] and [34] presented computer-aided diagnostic approaches for identifying malaria parasites using pre-trained convolutional neural networks as feature extractors. The CNN models are pre-trained on large-scale datasets like ImageNet in order to transmit the learned information in the form of generic image characteristics that can be used for the target job. Pretrained weights provide a decent initialization and outperform training the model from start with randomly initialized weights.

The difficulty of manually acquiring relevant, well-augmented data in large quantities is a

drawback of a semi-supervised learning strategy compared to Transfer Learning. As a result, Transfer Learning proves to be the most practical solution. Until far, the use of transfer learning has primarily been limited to classification and regression problems. In this project, I use this strategy for the Detection of malaria cells.

There are many transfer learning architectures available for the detection of images. I have gone through some of them which are described below:

2.3.2.1 VGG-16 and VGG-19

A computer vision competition is held every year called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Teams compete on two assignments per year. First, Object localization which is the process of finding objects in a picture that belong to one of 200 classifications. The second is picture classification, which is the process of labeling each of an image's 1000 different categories. In their study "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION" from 2014, Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab at Oxford University presented VGG 16. In the aforementioned categories, this model took first and second place in the 2014 ILSVRC competition. The VGGNet has two variants: VGG16 and VGG 19. The architecture of the VGG16 has 16 layers, whereas the VGG19 has 19 layers.

2.3.2.2 VGG-16 Architecture

It is a 16 layers depth convolution neural network. sequential convolution layers are stacked together to create the model. Figure 2.7 shows the Architecture of the VGG-16 model.

- VGG-16 model takes an image input dimension of (224x224x3).
- First, this network has two convolution layers. 64 filters of size (3x3) are used in each of these two convolution layers with the same padding that makes the new dimension (224, 224, 64).
- After that a maxpooling layer of stride size (2x2) is used that reduce the height and width of a volume. It goes from (224, 224, 64) down to (112, 112, 64).
- Thereafter, this network has two convolution layers. 128 filters of size (3x3) are used in each of these two convolution layers with the same padding that makes the new dimension (112, 112, 128).

- After that max-pooling layer of stride size (2x2) is used which makes the dimension (56,56,128).
- Afterward, this network has three convolution layers. 256 filters of size(3x3) are used in each of these three convolution layers with the same padding that make the new dimension (56, 56, 256).
- After that, max-pooling layer of stride size (2x2) is used which makes the dimension (28, 28, 256).
- Thereafter this network has three convolution layers. 512 filters of size(3x3) are used in each of these three convolution layers with the same padding that make the new dimension (28, 28, 512).
- After that max-pooling layer of stride size (2x2) is used which makes the dimension(14, 14, 512).
- Thereafter this network has three convolution layers. 512 filters of size(3x3) are used in each of these three convolution layers that make the new dimension (14, 14, 512).
- After that max-pooling layer of stride size (2x2) is used which makes the dimension(7, 7, 512).
- In the end, we have the final (7,7,512) into a Fully connected layer (FC) with 4096 units and in a softmax output one of 1000 classes.

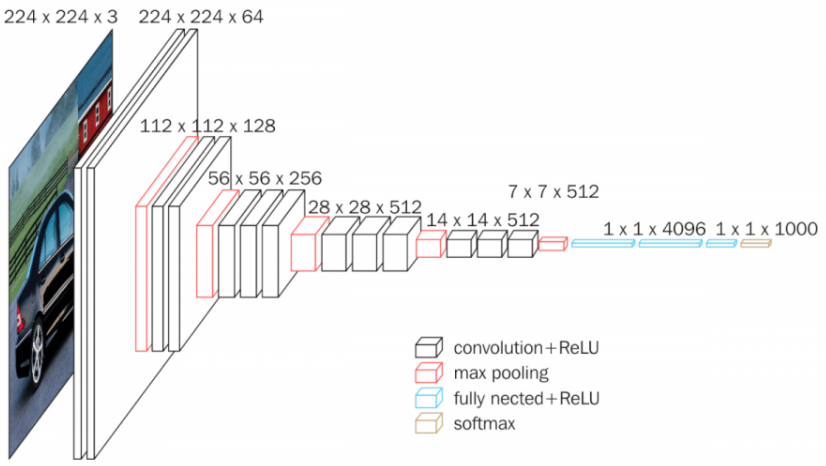


Figure 2.7: VGG-16 Model

2.3.2.3 VGG-19 Architecture

It is a 19 layers depth convolution neural network. sequential convolution layers are stacked together to create the model. Figure 2.8 shows the Architecture of the VGG-19 model.

- VGG-19 model takes an image input dimension of (224x224x3).
- First, this network has two convolution layers. 64 filters of size (3x3) are used in each of these two convolution layers with the same padding that makes the new dimension (224, 224, 64).
- After that a maxpooling layer of stride size (2x2) is used that reduce the height and width of a volume. It goes from (224, 224, 64) down to (112, 112, 64).
- Thereafter, this network has two convolution layers. 128 filters of size (3x3) are used in each of these two convolution layers with the same padding that makes the new dimension (112, 112, 128).
- After that max-pooling layer of stride size (2x2) is used which makes the dimension (56,56,128).
- Afterward, this network has four convolution layers. 256 filters of size(3x3) are used in each of these four convolution layers with the same padding that make the new dimension (56, 56, 256).
- After that, max-pooling layer of stride size (2x2) is used which makes the dimension (28, 28, 256).
- Thereafter this network has four convolution layers. 512 filters of size(3x3) are used in each of these four convolution layers with the same padding that make the new dimension (28, 28, 512).
- After that max-pooling layer of stride size (2x2) is used which makes the dimension(14, 14, 512).
- Thereafter this network has four convolution layers. 512 filters of size(3x3) are used in each of these four convolution layers that make the new dimension (14, 14, 512).

- After that max-pooling layer of stride size (2x2) is used which makes the dimension(7, 7, 512).
- In the end, we have the final (7,7,512) into a Fully connected layer (FC) with 4096 units and in a softmax output one of 1000 classes.

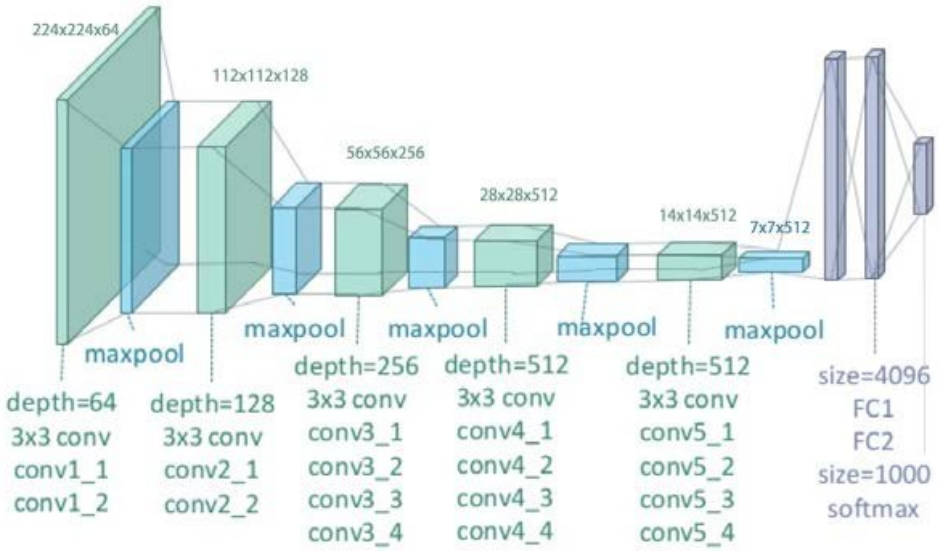


Figure 2.8: VGG-19

2.3.2.4 ResNet

ResNet is a traditional neural network that won the ImageNet competition in 2015 [28]. In traditional neural networks, connections are only allowed from one layer to the next layer forward. More layers were added to these networks to improve their performance. However, because of the Vanishing/ Exploding Gradients issue, which is produced by numerous layers of activation functions, this resulted in an increase in both training and testing errors after a certain point. The vanishing gradients problem was addressed by the development of ResNets. Skip connections are introduced in ResNet. These allow connections to be made from one layer to a few levels above it. By adding the identity term across the residual blocks, the signal loss caused by numerous levels of activation is compensated for by stronger signals. The network's strategy is to learn the underlying mapping. In other words, instead of H(x), the network fits $\mathbf{F(x) = H(x) - x}$, resulting in $\mathbf{H(x) = F(x) + x}$. This sort of skip connection is useful if any layer is degrading the architecture's performance, as it will be skipped by regularization shown in figure 2.9.

In this study, we applied transfer learning with ResNet50 to retrieve the result without changing the weights by freezing early blocks. On the model, we also used preprocessing approaches to fine-tune it.

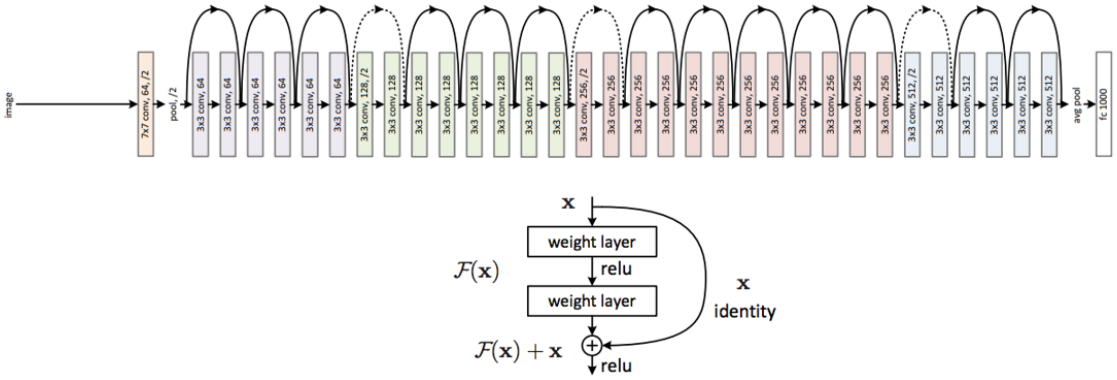


Figure 2.9: Resnet50

Chapter 3

Methodology

3.1 Dataset and Preprocessing

The data is collected from the official website of the National Institute of Health (NIH). There are 27,557 images in the collection, including 13,778 infected images and 13,779 non-infected images. Researchers from the National Library of Medicine's Lister Hill National Center for Biomedical Communications (LHNCBC) compiled the data. 150 P.falciparum-infected and 50 healthy individuals' Giemsa-stained thin blood smear slides were collected and photographed . Professional slide readers at the Mahidol-Oxford Tropical Medicine Research Unit in Bangkok, Thailand, painstakingly tagged the images.

There are 27,557 images in the malaria dataset, including 13,778 parasitized images (Figure 3.1) and 13,779 non-infected images (Figure 3.2). For training, validation, and testing, the dataset is separated into three sets. The train-to-validation-to-test ratio is 63:7:30. To avoid excessive computational intensity owing to the big dataset, we chose to train on only 63 percent of it.

The training dataset makes up 63 percent of the total dataset, with 8681 uninfected and 8681 parasited images, the validation dataset makes up 7% of the total dataset, with 964 uninfected and 964 parasited images, and the testing dataset makes up 30% of the total dataset, with 4135 and 4135 uninfected and Parasited images, respectively.

After studying the data set it is seen that images are in the dataset have different dimensions ranging from [46, 58] to [385, 394]. for this research, we resize all the images to [224,224] because the neural network takes an input of the same size.

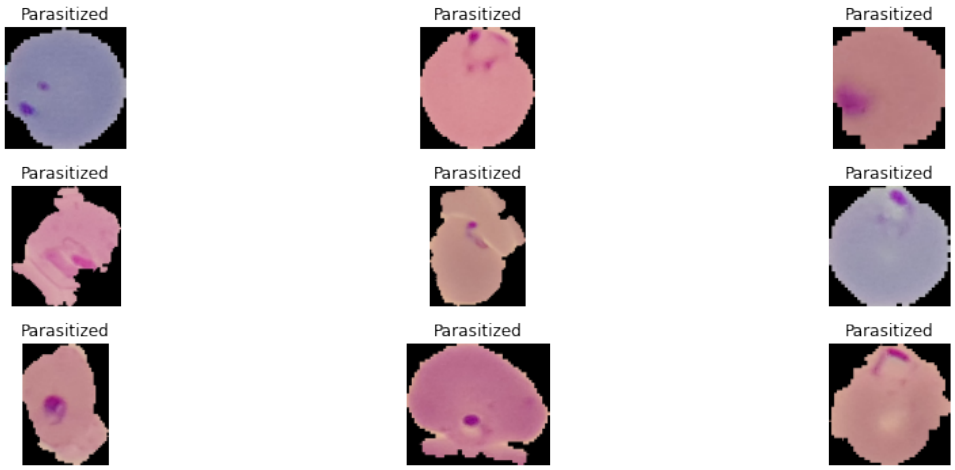


Figure 3.1: Parasitized cells

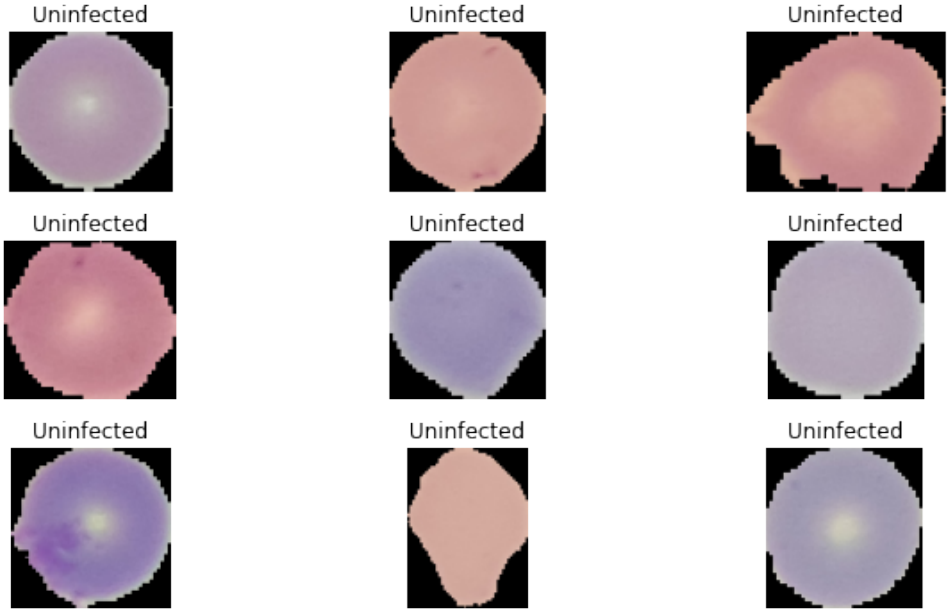


Figure 3.2: Uninfected cells

3.2 Data Augmentation

Deep learning algorithms are data-hungry. To learn the feature, the models need a huge amount of data. Data augmentation is used to increase the data points in the dataset. It expands the sample in the dataset by adding a variety of data samples. When we don't have enough training samples to cover diverse cases in image classification, often CNN might overfit. To address this we use a technique called data augmentation in deep learning. Data augmentation is used to generate new training samples from the current training set using various transformations such as scaling, rotation, contrast change, etc.

3.3 Proposed Models

3.3.1 Convolutional Neural Network

A convolutional neural network (CNN) is a type of deep learning architecture designed to recognize images. CNN is the deep learning model that is used as a feature extractor.For classifying the malaria cell we first create the CNN model from scratch. Figure 3.3 shows the model structure of our CNN model.

- In the first convolution layer, there is 32 kernels/filters size of(2x2), relu as an activation function, and max-pooling of pool_size (2x2) has been used.
- In the second convolution layer is the same as the first layer, here also we have used 32 filters of size(2x2), relu as activation function, and max-pooling of pool_size (2x2).
- In the third convolution layer we used 64 filters of size (2x2) ,relu activation, and max-pooling operation of pool_size(2x2).
- In the fourth convolution layer we used 128 filters of size (2x2), relu activation function, and max-pooling operation of pool size(2x2).
- Then we flatten the feature maps that are created by the previous convolution layers, that are used as input of the fully connected neural network that classifies the data.
- There has 512 nodes in 1st hidden layer. The next hidden layer has 256 nodes and the last output layer has 2 nodes.

BatchNormalization and dropout have been used after every max-pooling operation of the convolution layer for preventing the model to be overfitted. Model is compiled with 'adam ' optimization with 'categorical_crossentropy' as its loss function. This CNN network has:
Total params: 13,027,202
Trainable params: 13,025,154
Non-trainable params: 2,048

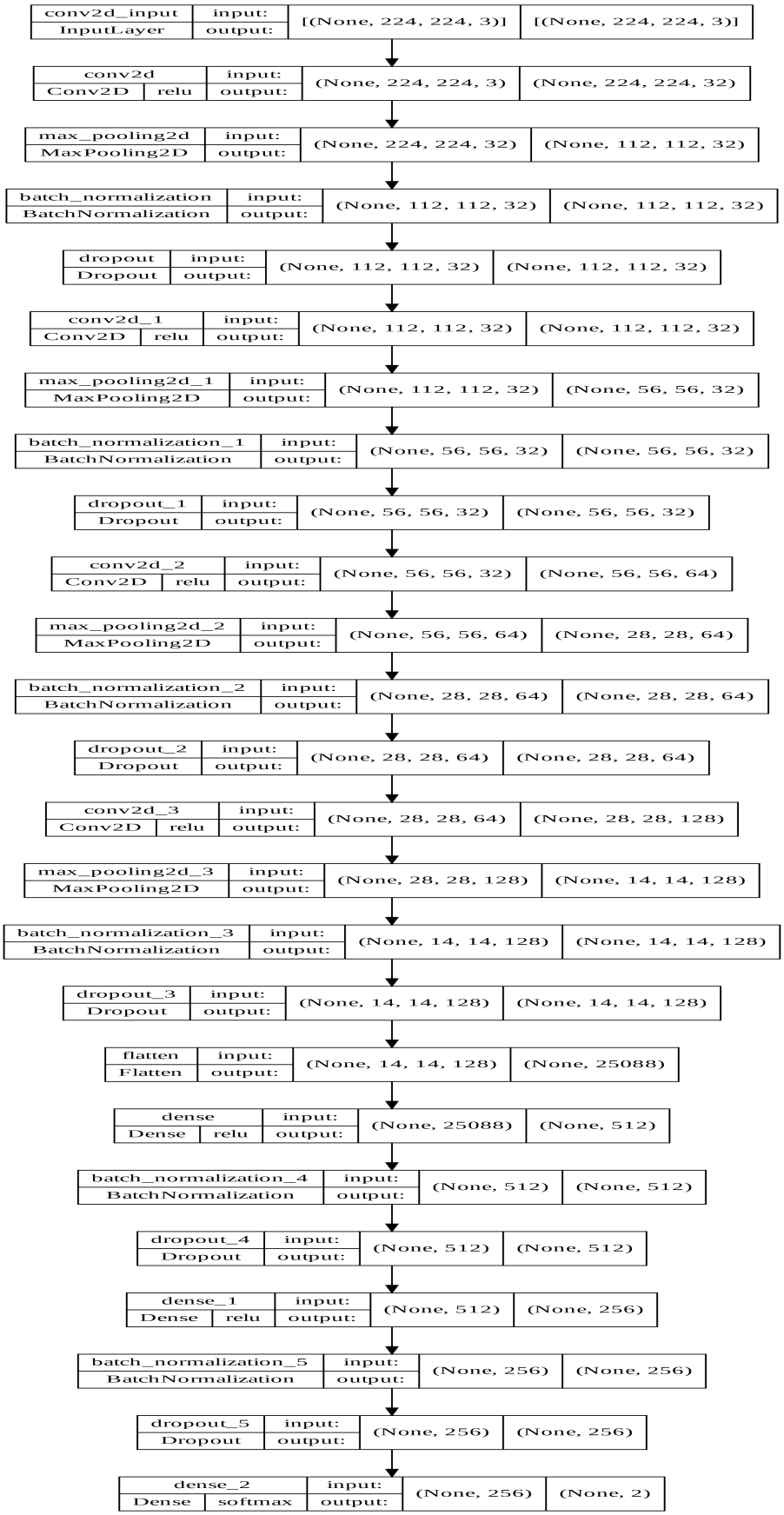


Figure 3.3: CNN Model

3.3.2 Transfer learning

Transfer learning is the technique where we reuse previous models to solve a new challenge or problem. Previous training knowledge is reused to aid in the completion of a new task. here in this research, we used models that trained on an ImageNet dataset with 100,000 data points and these transfer learning models are used in many image classification tasks and give better performance.
we have used VGG19 and ResNet-50 Transfer learning models to classify the malaria cells and evaluate their performance.

3.3.2.1 VGG-16

One of the variants of the VGG model is the VGG-16. It is a pre-trained model trained on Imagenet Dataset and saved the models weights. The 16 in VGG-16 stands for 16 weighted layers. Thirty convolutional layers, five Max Pooling layers, three Dense layers, and a total of 21 layers make up VGG-16, although only sixteen of those layers are weight layers, also known as learnable parameters layers. we implemented the Frozen VGG-16 by freezing all inherent convolution layers without adding the top layers of the original model. ather we created a dense layer of a network of 128 nodes and an output layer of 2 nodes after flattering the output of this VGG-16 model. Then this model is trained on our malaria dataset. Figure 3.4 (UP) shows the architechture of the VGG-16 pre-trained model.

3.3.2.2 VGG-19

VGG19 is one of the varient of VGG model. it is 19 layer depth convolution neural network. VGG-19 Architecture has 16 convolutional layers and 3 fully connected layers with five max-pool layers. It takes the input of size (224x224x3). Figure 3.4 (down) shows the general Architecture of the VGG-19 model. we implemented the Frozen VGG19 by freezing all inherent convolution layers without adding the top layers of the original model. Here we used VGG-19 as a feature extractor and we created a fully connected neural network on the top of the output of this model which is used to classify these features. At first, we initialized the VGG-19 base model with pre-trained weights. We did not add the top layers(fully connected layers) of the original VGG-19 model. Rather we established a dense layer of a network of 128 nodes and an output layer of 2 nodes after flattering the output of this VGG-19 model. This

model is trained on our malaria dataset.

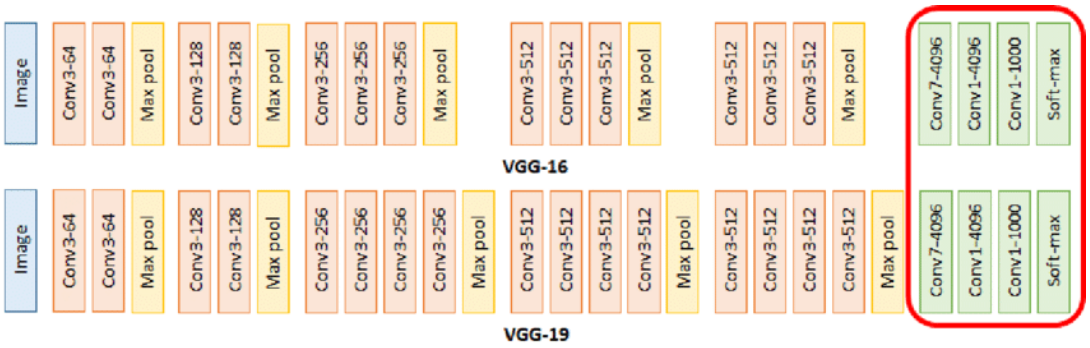


Figure 3.4: VGG16 & VGG19

3.3.2.3 Fine-Tuned VGG-19

The second model was a Fine-tuned model, in which we used transfer learning from the ImageNet dataset to freeze the first four blocks and made block five trainable on the malaria dataset. The VGG model was fine-tuned by unfreezing the last block so that their weights are changed in each iteration (per batch of data) while we train our model. In this model, we used all preprocessing procedures, including standardization, data augmentation, and normalization. To solve our classification problem, we employed the sigmoid activation function using two methods to get the result as 1 for malaria and 0 for healthy.

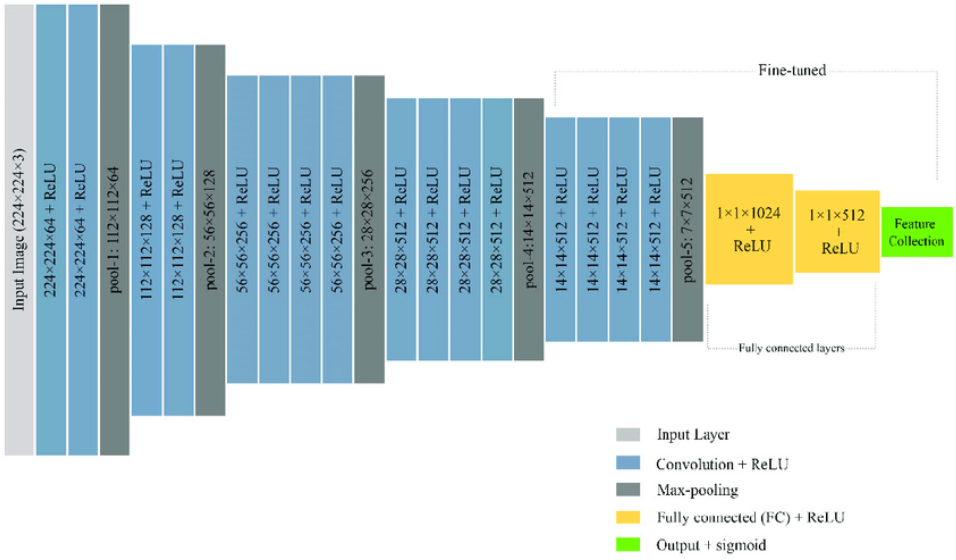


Figure 3.5: Fine-Tuned VGG-19

3.3.2.4 ResNet-50 and Fine-tuned ResNet-50

In the classification job, ResNet50 is the other most usually utilized architecture. The design is well-known for its depth and uses residual blocks to teach deep architecture by introducing identity skip links.In stages of residual blocks, the convolutional layer depth is increased in the ResNets design. Five stages make up the ResNet-50 model, each comprising a convolution block and identity block. There are three convolution layers in each convolution block, and there are three convolution layers in each identity block. More than 25 million parameters can be trained with the ResNet-50. The task at hand requires the binary classification of images, and sigmoid performs well with binary classification, which is why it should be used instead of softmax. Multi-class classification works well with Softmax. The metrics used to measure the model were Accuracy and Loss.

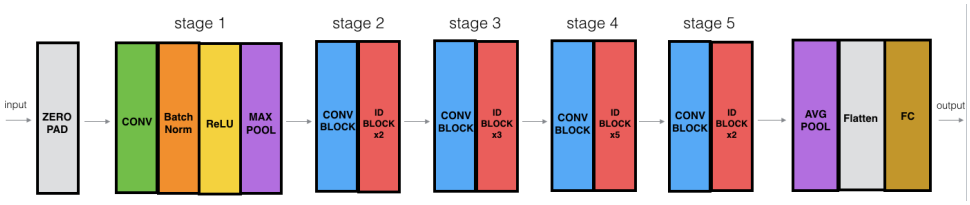


Figure 3.6: Resnet-50 Model

As we see in the fine_tuned in VGG-19, in the process of fine-tuning we don't train all the layers of the model. we train some of the layers so that the weights of these layers can change at the time of training and the rest of the layers' weights remain the same (Pre-trained weights). The base Resnet-50 model has 190 layers including weighted and non-weighted layers. we train the layers from 100 to 190 and before the 100th layer, all the layer's weights remain the same, they are not updated during training.

3.4 Performance Metric

The evaluation of a malaria detection task is done by calculating the percentage of the correctly classified image. The performance metric employed in this study is test accuracy and validation accuracy, which is utilized to assess the suggested models' correctness. The measure is often calculated for each class individually as well as for all courses collectively. Each predicted class value can be true positive (TP), true negative (TN), false positive (FP), or false negative (FN) for the per-class accuracy, i.e. when the prediction of the cell image to which class belongs is

assessed.

When the model correctly classifies positive class the healthy images it is called True Positive. When the model correctly classifies negative class the infected images it is called True Negative. False positive is the case when model classify the infected images as healthy. This is mostly when malaria is in the early stages and it is not easy to spot enough marks to classify it as infected. False negative is the case when the model classifies a healthy cell as infected. This is usually the case when the model fails to differentiate between red blood cells and mark them as infected.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

3.5 Experimental Setup

Deep learning models depend hugely on the GPUs and Compute Unified Device Architecture (CUDA) core enabled. Table 3.2 provides the system specifications required.

Hardware/Software	**Specifications**
RAM	12 GB
GPU	NVIDIA Tesla T4
GPU Memory	16 GB
GPU Memory Clock	1.59 GHz
Performance	8.1 TFLOPS
CUDA Cores	2560
No. CPU Cores	2
Disk Space Available	358GB
Programming Language	Python
Platform	Google Colaboratory

Table 3.1: System Configuration

On the NVIDIA Tesla T4 machine, the suggested model was trained and tested. It contains 16 GB of memory and 2560 CUDA cores. The suggested models were developed using the Keras deep learning framework. Keras manages a variety of input-output model configurations and layer definitions, among other neural network building components. Additionally, it builds our model with a loss function, an optimization function, and a fit function. All of the low-level

processing and model construction is done by Keras' "backend" component. TensorFlow is the backend engine by default, however, this may be modified later in settings.

Chapter 4

Results and Discussion

In this chapter, we discuss the result of our experiment. For this detection of malaria, we have created a CNN model and applied different transfer learning techniques.Table 4.1 represents the series of models developed, accuracy(%) and the validation accuracy(%) acquired after 25 epochs.

Model	Accuracy	Validation Accuracy
CNN	95	95
VGG-16	91	93
VGG-19	90	91
VGG-19 Fine Tuned	94	96
Resnet-50	91	93
Resnet-50 Fine Tuned	96	97

Table 4.1: Accuracy of the Proposed Models

Initially, we created a CNN model from scratch that gives the accuracy of 95% and validation accuracy of 95%. Train loss gradually decreases during 25 epochs and decreases at 0.1504. Figure 4.1 graph (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase.

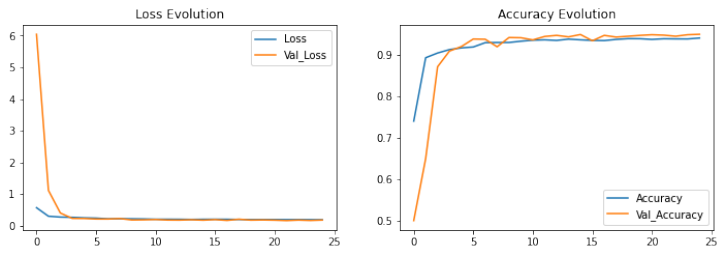


Figure 4.1: Accuracy and loss Graph

After that, we applied the Transfer learning approach Vgg-16. This is a pre-trained model, trained on the ImageNet dataset. we did not train all the layers of the VGG-16 model, we used the pre-trained weights of the original model without adding the top layers of the original model. This model gives the accuracy of 91% and validation accuracy of 93%. Train loss gradually decreases during 25 epochs and decreases at 0.2130. Our CNN model performed better than this Pretrained VGG-16 model. Figure 4.2 graph (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase.

Figure 4.2: Accuracy and loss Graph of VGG-16

We applied the VGG-19 model, a Transfer learning approach, which is trained on the ImageNet dataset. we did not train all the layers of the VGG-19 model, we used the pre-trained weights without adding the top layers of the original model. This model gives the accuracy of 90% and validation accuracy of 91%. Train loss gradually decreases during 25 epochs and decreases at 0.2457. Our CNN model performed better than this Pretrained VGG-19 model. Figure 4.3 (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase.

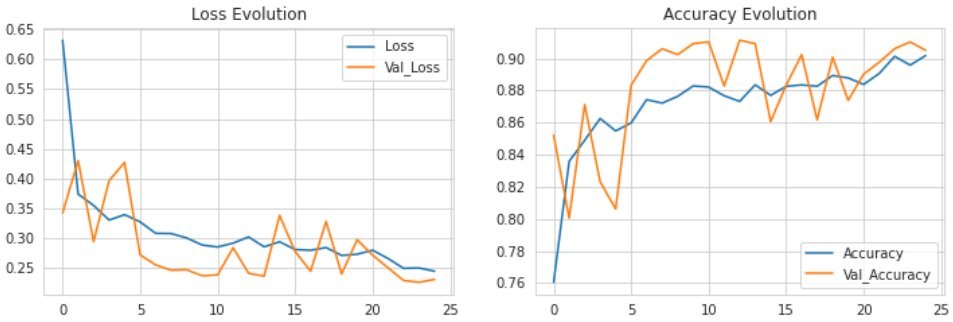


Figure 4.3: Accuracy and loss Graph of VGG-19

Next applied the Fine-Tuning approach, In the fine-tuning of VGG-19, we trained some of the layers near to tops layers because the initial convolution layers of the model capture the generalized features but as the depth of convolution layers increases the convolution layers try to capture the domain-specific feature. so we trained some of the deep layers and other layers' weight remains as it is, that is its pre-trained weight. This model gives the training accuracy of 94% and validation accuracy of 96%. Train loss gradually decreases during 25 epochs and decreases at 0.1441. Figure 4.4 (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase. This model performs slides better than our CNN model.

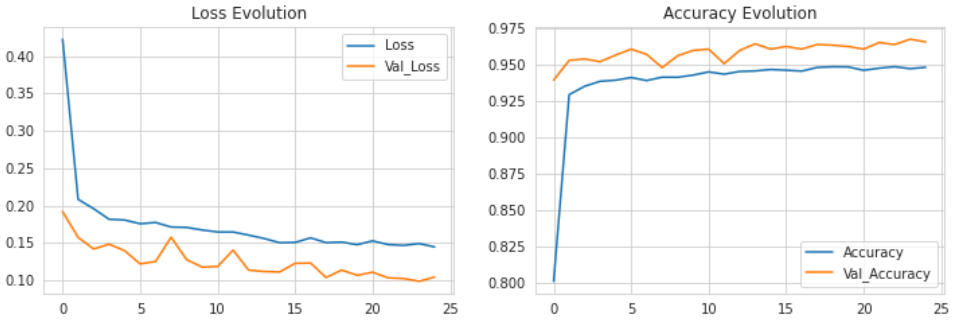


Figure 4.4: Accuracy and loss Graph of Fine_tuned_VGG-19

.

we applied another Transfer learning model, Resnet-50, which is trained on the ImageNet dataset. we did not train all the layers we used the pre-trained weights. This model gives the training accuracy of 91% and validation accuracy of 93%. Train loss gradually decreases during 25 epochs and decreases at 0.2263. Our CNN model performed better than this Pretrained model. Figure 4.5 (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase.

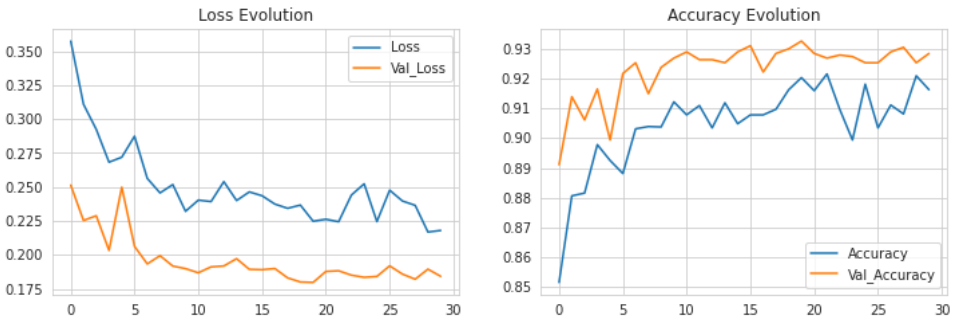


Figure 4.5: Accuracy and loss Graph of Resnet-50 model

At last, we used Fine-Tuning of Resnet-50 model. we trained some of the deep layers and other layers' weight remains as it is, that is its pre-trained weight. This model gives the training accuracy of 96% and validation accuracy of 97%. Train loss gradually decreases during 25 epochs and decreases at 0.0914. Figure 4.6 (left) shows how train loss and validation loss go down as epochs increase and (right) train and validation accuracy increase as epochs increase. This model performs better than our CNN model.

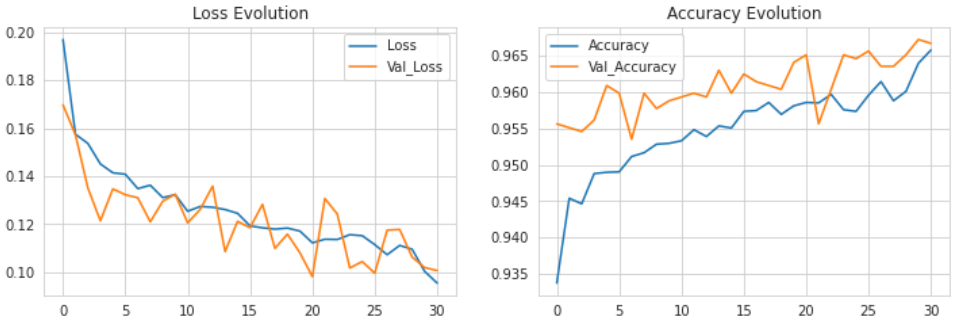


Figure 4.6: Accuracy and loss Graph of Fine_Tuned_Resnet-50 model

After analyzing the results of the various model, we can say that our CNN model performs better than VGG-16, VGG-19. Our CNN model presents the training accuracy and validation accuracy of 95%.

Then we try to increase the accuracy and decrease the loss by fine-tuning the VGG-19 model and fine-tuning the Resnet-50 model. From the above result, we can see that VGG-19 does not perform better than our CNN model. But Resnet-50 performs better than our CNN model. The training accuracy and validation accuracy of this fine-tuned model are respectively 96% and 97%.

Chapter 5

Conclusion and Future Work

In this research, we developed a CNN model and explored end-to-end deep learning neural networks to enhance the performance of malaria diagnostic categorization. We demonstrated that performance is not greatly enhanced by preprocessing approaches including normalization, standardization, and staining. As an alternative, techniques like data augmentation improved the performance of the models. We also compared the performances of several models, including the VGG-19 and ResNet-50. By starting from scratch, using transfer learning, and hyper-tuning the parameters, we created the VGG-19 and ResNet-50 models.Transfer learning is a great technique and could be used to gain satisfactory performance compared to machine learning models that require a lot of feature scaling and engineering. ResNet-50 outperformed all the other models by achieving an accuracy of 96.%.

In future works, we plan to focus on network architecture of the models used in the research to find the driving performance factors. We plan on finding ways to improve the performance by manipulating the network architecture and hyper tuning the features to achieve better performing model.

Although this proposed model of mine is not absolutely fool-proof and has got certain low accuracy in some models, it can still come in handy to the general physicians all across the terrain of a region. The question is how?: Due to the catastrophic situation which arose across the nation due to the epidemic, there was acute crysis of proper healthcare facilities/infrastructure/medical attendant in various parts of the nation, especially affecting the densely populated areas of a township, metropolitan, and in remote areas as well. After the advent of this model, it will be relatively easier to estimate whether the subject has been infected with malaria or not by virtue of the pictorial analysis of body fluids and comparing the

data afterwards.

For the consultant physician at a healthcare unit, this model can be handy and can easily play second fiddle to the mainstream treatment. For the newly recruited medical staff, this model can provide them with the near sufficient amount of info. for them to initiate the proceedings of preliminary checkup and thereafter treatment.

References

[1] Andrej Trampuz, Matjaz Jereb, Igor Muzlovic, and Rajesh M. Prabhu. Clinical review: Severe malaria. Critical Care, 7(4):315–323, aug 2003.

[2] This year's World malaria report at a glance. https://www.who.int/malaria/media/world-malaria-report-2018/en/, 2018.

[3] Das DK, Maiti AK, Chakraborty C (2015) Automated system for characterization and classification of malaria-infected stages using light microscopic images of thin blood smears. J Microsc 257(3):238–252

[4] Díaz G, González FA, Romero E (2009) A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. J Biomed Inform 42(2):296–307

[5] May Z, Mohd Aziz SSA, Salamat R (2013) Automated quantification and classification of malaria parasites in thin blood smears. IEEE International Conference on Signal and Image Processing Applications, Melaka, pp 369–373

[6] Poostchi M, Silamut K, Maude RJ, Jaeger S, Thoma G (2018) Image analysis and machine learning for detecting malaria. Trans Res: J Lab Clin Med 194:36–55

[7] Arco JE, Gorriz JM, Ramírez J, Alvarez I, Puntonet CG (2015) Digital image analysis for automatic enumeration of malaria parasites using morphological operations. Expert Syst Appl 42:3041–3047

[8] Savkare SS, Narote SP (2015) Automated system for malaria parasite identification. International Conference on Communication, Information Computing Technology (ICCICT), Mumbai, pp 1–4

[9] Soni J, Mishra N, Kamargaonkar (2011) Automatic difference between RBC and malaria parasites based on morphology with first order features using image processing. Int J Adv Eng Technol 1:290–297

[10] Pallavi T (2013) Suradkar, detection of malarial parasite in blood using image processing. Int J Eng Inn Technol (IJEIT) 2(10):124–126

[11] Boray Tek F, Dempster AG, Kale İ (2010) Parasite detection and identification for automated thin blood film malaria diagnosis. Comput Vis Image Underst 114(1):21–32

[12] Abbas N, Mohamad D (2013) Microscopic RGB color images enhancement for blood cells segmentation in YCBCR color space for k-means clustering. J Theor Appl Inf Technol 55(1):117–125

[13] Ma C, Harrison P, Wang L, Coppel RL (2010) Automated estimation of parasitaemia of Plasmodium yoelii- infected mice by digital image analysis of Giemsa-stained thin blood smears. Malaria J 9(348)

[14] Khan NA, Pervaz H, Latif AK, Musharraf A, Saniya (2014) Unsupervised identification of malaria parasites using computer vision, 11th international joint conference on computer science and software engineering (JCSSE). Chon Buri:263–267

[15] Chayadevi M, Raju G (2014) Usage of art for automatic malaria parasite identification based on fractal features. Int J Video Image Proc Netw Sec 14(4):7–15

[16] Makkapati VV, Rao RM (2011) Ontology-based malaria parasite stage and species identification from peripheral blood smear images. Ann Int Conf IEEE Eng Med Biol Soc:6138–6141

[17] Damahe LB, Krishna R, Janwe N (2011) Segmentation based approach to detect parasites and RBCs in blood cell images. Int J Comput Sci Appl 4(2):71–81

[18] ' Chavan SN, Sutkar AM (2014) Malaria disease identification and analysis using image processing. Int J Latest Trends Eng Technol 3:218–223

[19] Malihi L,Ansari-Asl K,Behbahani A (2013) Malaria parasite detection in giemsa-stained blood cell images, 8th Iranian Conference on Machine Vision and Image Processing (MVIP), Zanjan. 360–365

[20] Špringl V (2009) Automatic malaria diagnosis through microscopy imaging. Higher Diploma, Faculty of electrical engineering, Czech Technical University in Prague, Prague

[21] Arunagiri, Vijayalakshmi B, Rajesh. (2020). Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications. 79. 10.1007/s11042-019-7162-y.

[22] Abu NS, Ashidi NMI, Chia LL, Mohamed Z, Kalthum UN, Zuhairi KZ (2008) Classification of malaria parasite species based on thin blood smears using multilayer perceptron network. Int J Comput Int Manag 16(1):46–52

[23] Khot ST, Prasad RK (2014) Optimal computer based analysis for detecting malarial parasites, proceedings of the 3rd international conference on Frontiers of intelligent computing: theory and applications (FICTA). Adv Intel Syst Comput, Springer, Cham 327:69–80

[24] Anggraini D, Nugroho AS, Pratama C, Rozi IE, Pragesjvara V, Gunawan M (2011) Automated status identification of microscopic images obtained from malaria thin blood smears using bayes decision: a study case in Plasmodium falciparum. International Conference on Advanced Computer Science and Information Systems, Jakarta, pp 347–352

[25] Memeu, Daniel Mathethia (2014) A rapid malaria diagnostic method based on automatic detection and classification of plasmodium parasites In stained thin blood smear images, University of Nairobi

[26] Prasad K, Winter J, Bhat UM, Acharya RV, Prabhu GK (2012) Image analysis approach for development of a decision support system for detection of malaria parasites in thin blood smear images. J Digit Imaging 25(4):542–549

[27] Kumarasamy SK, Ong SH, Tan KSW (2011) Robust contour reconstruction of red blood cells and parasites in the automated identification of the stages of malarial infection. Mach Vis Appl 22(3):461–469

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on

Computer Vision and Pattern Recognition, volume 2016-Decem, pages 770–778. IEEE Computer Society, dec 2016.

[29] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning, may 2015.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks.

[31] Rajaraman S et al. (2018) Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. Peer J

[32] Kermany DS et al (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122–1131

[33] Esra Var and F. Boray Tek. Malaria Parasite Detection with Deep Transfer Learning. In UBMK 2018 - 3rd International Conference on Computer Science and Engineering, pages 298–302. Institute of Electrical and Electronics Engineers Inc., dec 2018.

[34] Sivaramakrishnan Rajaraman, Sameer K. Antani, Mahdieh Poostchi, Kamolrat Silamut, Md A. Hossain, Richard J. Maude, Stefan Jaeger, and George R. Thoma. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. IEEE Transactions on Medical Imaging, 35(5):315–323, aug 2016.

[35] Team, K. (n.d.). Keras documentation: Conv2D layer. Keras. https://keras.io/api/layers/convolution_ layers/convolution2d/. Retrieved 21 February 2021.

[36] GeeksforGeeks. (2019, July 26). CNN: Introduction to Padding. GeeksforGeeks. https://www.geeksforgeeks.org/cnn-introduction-to-padding/. Retrieved 02 May 2021.

[37] DeepAI. (2019, May 17). Stride (Machine Learning). DeepAI. https://deepai.org/machine-learning-glossary- and-terms/stride. Retrieved 23 April 2021.

[38] Wikipedia.(2020, December 12). Cross entropy. Wikipedia. https://en.wikipedia.org/wiki/Cross_entropy. Retrieved 23 june 2022.

[39] Brownlee, J. (2021, January 12). Gentle introduction to the Adam optimization algorithm for deep learning. Machine Learning Mastery. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep- learning/. Retrieved 02 February 2021.

[40] Yuhang Dong, Zhuocheng Jiang, Hongda Shen, W. David Pan, Lance A. Williams, Vishnu V.B. Reddy, William H. Benjamin, and Allen W. Bryan. Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In IEEE International Conference on Biomedical and Health Informatics, pages 101–104. Institute of Electrical and Electronics Engineers Inc., apr 2017.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks.

[42] Dhanya Bibin, Madhu S. Nair, and P. Punitha. Malaria Parasite Detection from Peripheral Blood Smear Images Using Deep Belief Networks. IEEE Access, 5:9099– 9108, 2017.

[43] Gopalakrishna Pillai Gopakumar, Murali Swetha, Gorthi Sai Siva, and Gorthi R. K Sai Subrahmanyam. Convolutional neural network-based malaria diagnosis from focus stack of blood smear images acquired using custom-built slide scanner. Journal of Biophotonics, 11(3):e201700003, mar 2018.