

**CLASSIFICATION OF MALARIA
INFECTED CELLS USING INCEPTION V1
NETWORK**

**A THESIS SUBMITTED TO THE
GRADUATE SCHOOL OF APPLIED
SCIENCES
OF
NEAR EAST UNIVERSITY**

**By
MAHA TELLAWI**

**In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Electrical and Electronics Engineering**

NICOSIA, 2019

MAHA TELLAWI

**CLASSIFICATION OF MALARIA INFECTED CELLS USING
INCEPTION V1 NETWORK**

**NEU
2019**

**CLASSIFICATION OF MALARIA INFECTED
CELLS USING INCEPTION V1 NETWORK**

**A THESIS SUBMITTED TO THE
GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY**

**By
MAHA TELLAWI**

**In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Electrical and Electronics Engineering**

NICOSIA, 2019

**Maha TELLAWI: CLASSIFICATION OF MALARIA INFECTED CELLS USING
INCEPTION V1 NETWORK**

**Approval of Director of Graduate School of
Applied Sciences**

Prof. Dr.Nadire CAVUS

**We certify this thesis is satisfactory for the award of the degree of Masters of Sciences
in Electrical and Electronics Engineering**

Examining Committee in Charge:

Assoc.Prof.Dr Kamil Dimililer

**Chairperson, Department of
Automotive Engineering , NEU**

Assist.Prof.Dr Sertan Serte

**Supervisor, Department of
Electrical and Electronics
Engineering, NEU**

Assist.Prof.Dr Ali Serener

**Department of Electrical and
Electronics Engineering, NEU**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name:

Signature:

Date:

To my parents...

To my dear husband...

ACKNOWLEDGMENTS

All praises and thanks to my family. It is by their grace that I have been able to access this point in my life.

I would like to express my sincere gratitude to my supervisor, Assist. Prof. Dr. Sertan Serte who has supported and directed me with her vast knowledge and also for his patience that ensured the completion of this thesis.

I got enough support and encouragement from my parents, without you, I would not be the person I am today, especially my mother-in law who always give first priority to my higher studies. My brothers, Hussam, Gazi, and Sultan, thank you for your love, support, and unwavering belief in me.

I express my gratitude to my dear husband, Osman. I am so appreciative for his constant love, understanding and encouraging, for always showing how proud he is of me, for his taking up the whole responsibilities to our family and bearing the pressure both from working and living during my studying. I cannot express my gratefulness in words without you this effort would have been worth nothing. But most of all, thank you for being my best friend. I owe you everything. I dedicate this book to him.

Finally, the last word goes for Gazi, my brother, who has stood by me through all my travails, my absences, my fits of pique and impatience. He gave me support and help.

ABSTRACT

Malaria detection and classification is still time and money costly. Nowadays, identification of malaria cells is achieved using some techniques that are relatively good, but they require time and high cost. Hence, there is a need of discovering alternative techniques to identify blood cells, that save both time and cost. In addition to time and cost, those new techniques should also be accurate and effective. Thus, in this work, we propose a transfer learning based GoogleNet approach for the classification of Malaria cells. The depth and inception of GoogleNet made it a very robust deep network that can classify accurately if trained and fine-tuned on enough number of data. Thus, in this study, 27558 of the 2 types of Malaria cells are used for fine-tuning and testing the pre-trained network GoogleNet. Experimentally, the employed GoogleNet fine-tuned to classify Malaria, showed a great capability in generalizing accurate and correct diagnosis of images that were not seen during training, in which it achieved a testing accuracy of 95% with a relatively short time and small number of epochs (1.5 hours).

Keywords: Malaria; transfer learning; GoogleNet; deep network; epochs

ÖZET

Sıtmayı tespit etme ve sınıflandırma halen çok pahalı durumdadır. Sıtma hücrelerinin tanımlanması pahalı teknikler sahesinde yapılabilir. Bu teknikler çok iyi olmalarına karşın uzun süre ve çok para isteyen yöntemlerdir. Bu nedenle, kan hücrelerini tanımlamak için hem az zaman hem az para gerektiren alternatif teknikler gerekmektedir. Zamandan ve fiyattan hesaplı olmaları yanında, bu yeni tekniklerin ayrıca doğru ve etkili olmaları lazım. Bundan ötürü, bu çalışmada biz sıtma hücrelerini sınıflandırmak için transfer öğrenme tabanlı GoogleNet yaklaşımını öneriyoruz. GoogleNet'in derinliği ve başlangıcı eğer doğru yönte eğitilirse ve ince ayarlanırsa, bu sınıflandırmayı yapabilecek sağlam ve büyük bir ağ yapmaktadır. Bu yüzden, bu çalışmada 2 tip hücreden 27558 hücre ince ayar ve test yapmak amacıyla önceden eğitilmiş GoogleNet'de kullanılmıştır. Deneysel olarak, kullanılan sıtmayı tanımlamak için ince ayarlanmış GoogleNet, eğitim sırasında gösterilmeyen resimleri gennelleyerek doğru ve kesin tanımlamakta büyük başarı göstermiştir, bunu yaparken de doğruluk oranı %95 olmakla birlikte hem kısa zamanda hem de az miktarda devir ile süreci tamamlamıştır (1.5 saat).

Anahtar Kelimeler: Sıtma; transfer öğrenme; GoogleNet; derin ağ; devir

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	iii
ÖZET	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix

CHAPTER 1: INTRODUCTION

1.1 Introduction.....	1
1.2 Problem Formulation	3
1.3 Aims of Thesis	6
1.4 Thesis Overview	6

CHAPTER 2: NEURAL NETWORKS, DEEP AND TRANSFER LEARNING

2.1 Introduction.....	8
2.2 Malaria Detection Using Machine Learning	8
2.3 Malaria Detection Using Deep Learning	9
2.4 Neural Networks	11
2.4.1 Neural networks types.....	11
2.4.2 Single layer perceptron.....	12
2.5 Backpropagation Algorithm	15
2.6 Convolutional Neural Networks	16
2.7 Transfer Learning	19
2.7.1 GoogleNet (Inception).....	20
2.7.2 Inception Module	21

2.7.3 AlexNet	22
2.7.4 VGG-16.....	23

CHAPTER 3: MATERIALS AND METHODS

3.1 Methodology	24
3.2 Data	27
3.2 Training of Pre-trained Model GoogleNet.....	28
3.3 Testing of GoogleNet.....	30

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Results and Discussion	32
4.2 Performance Evaluation Metrics	34
4.3 Results Comparison	35

CHAPTER 5: CONCLUSION

REFERENCES	38
-------------------------	-----------

APPENDICES

Appendix 1: GoogleNet Source Code	42
Appendix 2: Activations Source Code	44

LIST OF FIGURES

Figure 1.1: Samples of parasitized and uninfected Malaria images	1
Figure 1.2: Original GoogleNet (Szegedy et al., 2015).....	4
Figure 1.3: Proposed system design	5
Figure 2.1: Architecture of artificial-neuron model (McCulloch and Pitts model) (Du andSwamy, 2013).....	13
Figure 2.2: Architecture of of Rosenblatt’s Perceptron (Du and Swamy, 2013.....	14
Figure 2.3: Effects of learning rate and momentum parameters on weight updating (Du and Swamy, 2013).....	16
Figure 2.4: : The LeNet-5 Architecture, a convolutional neural network.....	17
Figure 2.5: A Full Convolutional Neural Network Layers (LeNet).....	20
Figure 2.6: GoogleNet architecture and its inception module(Szegedy et al., 2015).....	21
Figure 2.7: Inception module in GoogleNet (Szegedy et al., 2015).....	22
Figure 2.8: Blockdiagram of AlexNet	23
Figure 2.9: Blockdiagram of VGG-16.....	23
Figure 3.1: Fine-tuning of GoogleNet to classify Malaria cells	25
Figure 3.2: Block diagram of the whole training and testing process of GoogleNet	26
Figure 3.3: Sample of dataset images of the two different Malaria cells	27
Figure 3.4: Accuracy variations with the change of Epochs	31
Figure 4.1: Learned filters of GoogleNet	33
Figure 4.2: ROC Curve	35

LIST OF TABLES

Table 3.1: Dataset description	27
Table 3.2: GoogleNet learning parameters	29
Table 3.3: Testing performance of GoogleNet	31
Table 4.1: Performances of the model during training and testing	33
Table 4.2: Performance metrics of the networks	34
Table 4.3: Results comparison with other works	36

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
MLP	Multilayer Perceptron
BPNN:	Back Propagation Neural network
SAE:	Stacked Auto-encoder
BPLA	Back Propagation Learning Algorithm
AE:	Auto-encoder
MSE:	Mean Square Error
SVM:	Support Vector Machine
CNN:	Convolutional Neural Network

CHAPTER 1

INTRODUCTION

1.1 Introduction

Malaria is a possibly deadly parasitic disease of both human and creatures. Half of the total population is in danger of this dangerous irresistible disease. Intestinal sickness is of extraordinary threat to pregnant lady and kids, particularly those under five (Razzak, 2015). As a rule, Malaria infection could be analyzed by tiny examination of blood films. So as to give a solid conclusion, important preparing and concentrated human asset are required. Patients experiencing Malaria disease ought to be analyzed at beginning time and ought to be given a compelling and reasonable treatment inside 24 hours (WHO, 2015). Shockingly, most infections happen in rustic regions, where assets are a long way from being sufficient. Additionally inability to analyze on time may prompt off base medicines. This disturbing circumstance has incited scientists to create telemedicine answers for quick and precise recognizable proof of intestinal sickness infection. To give a thought of the red platelets engaged with this examination, we demonstrate a few examples in Figure 1.

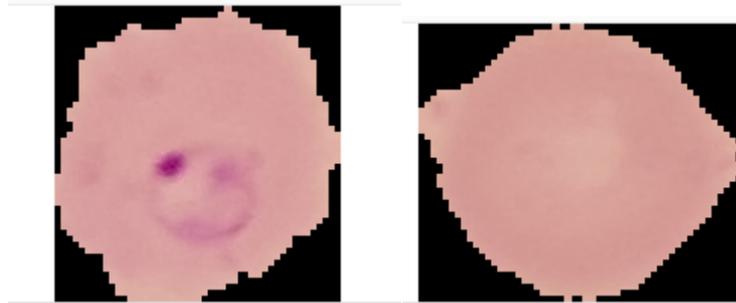


Figure 1.1: Samples of parasitized and uninfected Malaria images

Studies on Malaria cell classification provided many diagnostic methods, most of which were based on machine learning, including unsupervised (Tek et al., 2010) and supervised learning (Ghosha, et al., 2013). However, the performance of these methods are highly sensitive to features extracted from original images. Although many works has been done on feature extraction for malaria cells, new feature extraction methods need to be designed for different datasets.

Deep learning (DL) has been lately applied to medical field and it showed a great efficiency and accuracy. Deep networks showed a great capability in image classification in particularly, when transfer learning is used to transfer the knowledge extracted from one well-trained convolutional neural network (CNN) that is trained on millions of images to train on another task such medical image classification. Those convolutional neural networks (AlexNet, GoogleNet, VGG16, ResNet etc..) have been applied to many medical applications (Szegedy et al., 2015; Simonyan & Zisserman, 2014; Krizhevsky et al., 2012) and it was found that those networks can perform accurately without the need of large databases as they are already trained on millions of data which provided them with the good features extraction power due to their trained convolutional filters and feature maps.

With the progression of computational abilities and in addition the advancement of capacity limit with regards to expansive scale information, profound learning has grown quickly as of late. Along these lines, the achievability of mechanizing lab investigation work utilizing neural system has been examined in the ongoing literary works (Mckenna et al., 1993; Elsalamony, 2016; Manik et al., 2017). As of late, Andre Esteva et al. (2017) utilized profound neural systems in skin disease classification undertaking and the execution was practically identical to 21 talented American dermatologists, which showed the great capability of utilizing profound learning calculations in clinical examination work.

In this thesis, a transfer learning based automatic classification system that is applied for the Classification of Malaria Infected Cells is presented. In order to achieve fully automated diagnosis without any manual feature extraction, we chose deep convolutional neural network (CNN) as the classifier. CNN can extract hierarchical representations of the input data. In this work, Inception network (Szegedy et al., 2015); was used to learn the inherent features of malaria infected and non-infected cells. Inception network is one of the best known CNN architectures. In this work, we aim to apply transfer learning using Inception network that will be trained on large number of malaria images and evaluate their performances in classifying them into parasitic and uninfected. The network will be trained on a dataset of 27558 images collected from the Malaria Cell Images Dataset (Malaria Cell Images Dataset, 2017).

Infected and non-infected cells of malaria are being classified in this work. The pre-trained model employed in this work is the GoogleNet (Szegedy et al., 2015) which is due to its efficacy in images classification. The Network is fine-tuned on 2 malaria classes in order to learn the different levels features that can classify each type and ends up with a good accuracy and minimum error. The images used in fine-tuning the pre-trained model GoogleNet are obtained from a public database in which microscopic images of the different classes cells are available. Those microscopic images are then used in order to train and test the GoogleNet and evaluate its performance in this medical classification task. Moreover, the network performance is compared to similar and related researches that used other networks and methods in order to classify blood cells.

1.2 Problem Formulation

Malaria detection and classification is still time and money costly. Identification of malaria cells can be done through some costly techniques. Those techniques are good but they require time and high cost. Hence, there is a need of discovering alternative techniques to identify blood cells, that saves both time and reduce cost. In addition to time and cost, those new techniques should also be accurate and effective. Thus, in this work, we propose a transfer learning based GoogleNet approach for the classification of Malaria cells. The depth and inception of GoogleNet made it a very robust deep network that can classify accurately if trained and fine-tuned on enough number of data. Thus, in this study, 27558 of the 2 types of cells are used for fine-tuning and testing the pre-trained network GoogleNet.



Figure 1.2: Original GoogleNet (Szegedy et al., 2015)

Figure 1.1 shows the original structure and architecture of GoogleNet. As can be seen the network is a convolutional neural network that consists of many layers which makes it so deep as it is comprised of 22 layers. More details about GoogleNet and its working principles are found in chapter three. In this part, only the problem formulation and the proposed system for classifying of malaria cells are shown.

Figure 1.2 shows a glance of the problem that is solved in this thesis, and the way it is solved. As seen, GoogleNet is employed here in order to transfer its knowledge and features extraction abilities into a new target task which is 2 types of cells classification. GoogleNet is fine-tuned using enough number of images until it become capable of generalizing the type of input malaria cell image as seen in Figure 1.2. Note that a deep convolutional neural network uses Softmax at its output layer, which means it gives outputs as probabilities of each class, and the class that achieved the higher probabilities is the actual output predicted by the network.

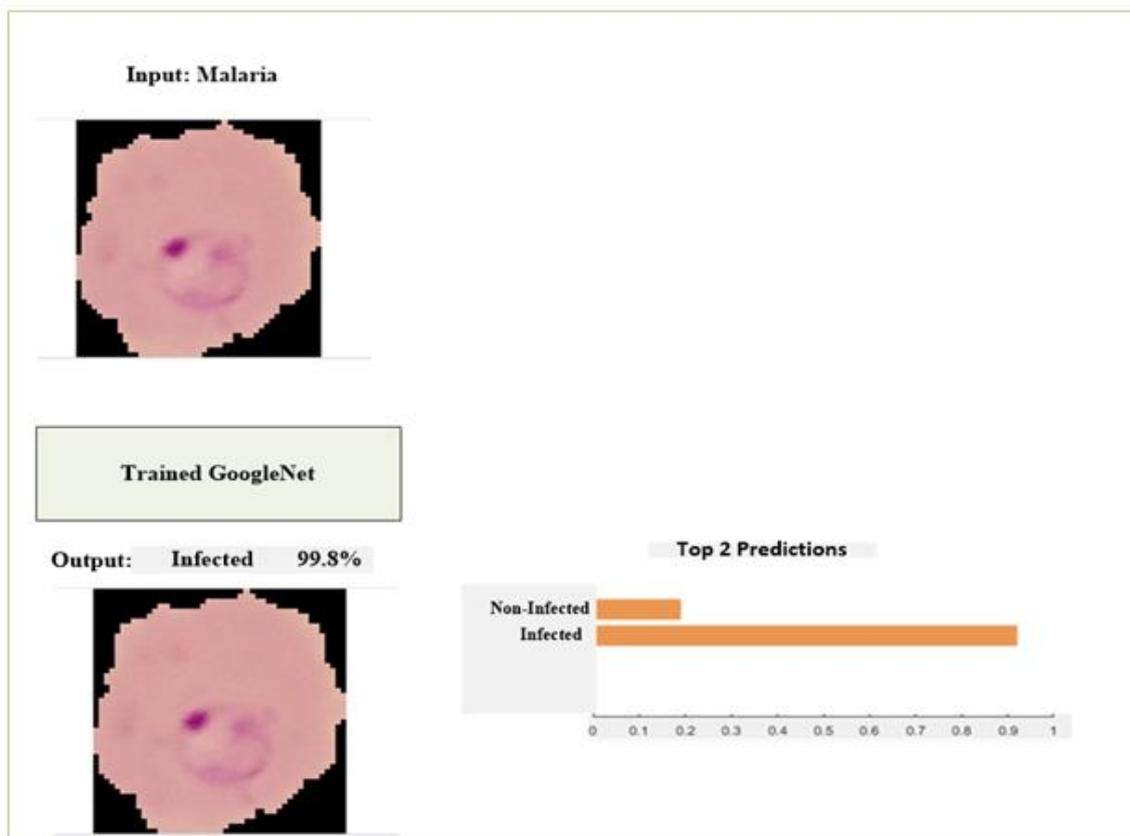


Figure 1.3: Proposed system design

1.3 Aims of Thesis

An effective and robust intelligent system for the Malaria cells classification is required. Thus, transfer learning was found as the best technique that can achieve that goal. The reason behind using transfer learning is that pre-trained models do not need large number of images to be trained and consequently, this results in shorter training time compared to training a network from scratch. Thus, GoogleNet is fine-tuned in order to perform one more classification which is Malaria cells classification into 2 types: Infected and non-infected.

Deep learning models are mostly effective when a large training dataset is applied. In the medical field, large datasets are not usually available. Hence, transfer learning may be the only solution. Transfer learning refers to the use of pre-trained convolutional neural networks that are already trained on large datasets such as ImageNet (Russakovsky et al., 2015), and benefit from their learned parameters, in particular weights, to the target network model.

In this thesis, we explore the strength of pre-trained model: GoogleNet, which is trained on images from ImageNet, which is considered a large scale nonmedical image dataset, for the task of Malaria cells images classification. We aim to investigate the generalization power of these deep learning approach when trained and tested on a relatively small database consisting of 27558 Malaria cells images.

1.4 Thesis Overview

This thesis is structured as follows:

- **Chapter one** is an introduction of the thesis which also shows the problem that is detected and studies in this thesis, as well as the scope, objective and the significance of the study.
- **Chapter two** includes theoretical background about neural networks. It also explains deep learning, transfer learning concepts in addition to discussing the convolutional neural network and how can transfer learning is accomplished.

- **Chapter three** presents the training phase of the work in which network training performance is discussed. Moreover, it also shows the network testing performance in addition to discussing the results and discussion of the thesis.
- **Chapter four** presents the results discussion and comparison of the network, in which results are discussed and compared with other works.
- **Chapter five** is a conclusion of the work.

CHAPTER 2

NEURAL NETWORKS, DEEP AND TRANSFER LEARNING

2.1 Introduction

The nature of human brain structure is complex and precise and because of these properties of brain structure, makes brain to have capability to perform various difficult assignments. Scientifically, human brain uses biological neurons to perform these tasks. The exact number of neurons is unknown but these neurons can be approximated around billions of neurons (linked with each other). The principle of artificial neural networks (ANNs) is inspired by the mechanism of human biological brain; thus artificial neural networks can be defined as an imitation of the structure and the function of the biological brain. ANN can be used for various applications, such as pattern recognition and classification of the data by training operations (Haykin, 2009) and (Du and Swamy, 2013).

2.2 Malaria Detection Using Machine Learning

A few methodologies have been proposed and executed in which Malaria must be recognized by taking a blood test of patients in the research facility. These methods cause a postponement in the beginning of treatment. Because of which, Death proportion is impressively higher for Malaria disease on the planet. Parveen et al., (2017) proposed a research to accelerate the procedure of Malaria finding and processing. An Artificial Neural Network with MPL (Multi Layer Perceptron) is utilized alongside back propagation, back propagation with energy and versatile propagation rule for the expectation of Malaria (Parveen et al.,2017). Among every one of the three learning rules, Back propagation gives the more effective outcomes around 85%. In their proposed methodology, history and indications of patients are considered as information, framework examinations that information and foresee the outcome for the unfortunate casualty as positive or negative for Malaria. This application is valuable for those regions where there is no any research center office or where there is no Doctor; in such condition the individual who ready to work the application by giving just verbal history and physical appearance of the patient.

Moreover, Anand et al., (2012) have utilized connection strategy on the pictures acquired from DHIM to separate the malarial cells from the sound ones. A proficient computational systems, in light of the 3-D pictures delivered by advanced holographic interferometric microscopy (DHIM), to naturally segregate among malarial and solid RBCs can be exceptionally helpful. It will be invaluable if such DHIM catching instruments are versatile and simple to utilize. We propose the strategy to recognize the malaria contaminated RBCs from the solid ones utilizing Artificial Neural Network (ANN), in view of its physical/factual highlights that are separated from the pictures got from Digital Holographic Interferometric Microscope (DHM).

Artificial Neural Network (ANN) for malaria finding is one time prepared clever program which is anything but difficult to utilize, compact, minimal effort and make malaria conclusion progressively fast and exact. Computerized holographic interferometric microscopy 3D image is the photographically or generally recorded obstruction design between a wave field dispersed from the item and a reasonable background, called the reference wave (Pandit & Anand, 2016).

2.3 Malaria Detection Using Deep Learning

Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected mosquitoes. Automation of the diagnosis process will enable accurate diagnosis of the disease and hence holds the promise of delivering reliable health-care to resource-scarce areas. Machine learning technologies have been used for automated diagnosis of malaria. Pan et al., (2017) present some of their recent progresses on highly accurate classification of malaria-infected cells using deep convolutional neural networks. First, authors described image processing methods used for segmentation of red blood cells from wholeslide images. Then they discussed the procedures of compiling a pathologists-curated image dataset for training deep neural network, as well as data augmentation methods used to significantly increase the size of the dataset, in light of the overfitting problem associated with training deep convolutional neural networks. Moreover, authors compared the classification accuracies obtained by deep convolutional neural networks through training, validating, and testing with various combinations of the datasets. Authors claimed that their deep convolutional neural networks behaved differently on different

datasets; means that accuracy was different from one dataset to another where it achieved a maximum value for the CNN based on LeNet-5, which achieved an accuracy of almost 98%.

Also, Vijayalakshmi and Kanna, (2019) presented a deep learning system based on VGG and Support vector machine (SVM) for the detection of malaria in microscopic images. Authors of this work used SVM as a classifier instead of neural network and the accuracy achieved was high (93.1%) when it was tested.

Furthermore, a valuable research work of the evaluation of deep networks in identifying Malaria was presented by Dong et al., (2017). This research work encountered the usage of many and different deep networks that were all trained and tested to identify Malaria. A dataset of 1282 images of normal and infected images were used for training the system. However, 1283 were used for testing and validation (Dong et al., 2017). Deep networks that were employed for this task are: LeNet-5, AlexNet, and GoogleNet. Experimentally, authors showed that these networks behave differently and hence achieved different accuracies after testing on the same number of images: LeNet-5: 96.18%, AlexNet: 95.78%, GoogleNet: 98.13%.

The Rajaraman, et al., paper use six pre-prepared models on a dataset to acquire an amazing accuracy of 95.9% in distinguishing malaria vs. non-infected samples. Their center was to attempt some straightforward CNN models starting with no outside help and a few pre-prepared models utilizing transfer learning to see the results.

Liang et al., (2016) presented a study that proposes a new and robust machine learning model based on a convolutional neural network (CNN) to automatically classify single cells in thin blood smears on standard microscope slides as either infected or uninfected. In a ten-fold cross-validation based on 27,578 single cell images, the average accuracy of our new 16-layer CNN model is 97.37%. A transfer learning model only achieves 91.99% on the same images. The CNN model shows superiority over the transfer learning model in all performance indicators such as sensitivity (96.99% vs 89.00%), specificity (97.75% vs 94.98%), precision (97.73% vs 95.12%), F1 score (97.36% vs 90.24%), and Matthews correlation coefficient (94.75% vs 85.25%).

2.4 Neural Networks

Artificial Neural Networks (ANNs) can be defined as a data processing model which tries to imitate the way of human biological brain works. There are many nodes (neurons) that linked or connected with each other through lines (weight) in ANNs; these neurons work with each other to find solution for specific tasks. The processes of neural networks (NN) consist of two steps; the first step is training or learning of neural network through use of data (examples) which can be carried out by using learning algorithm. Whereas, the second step is recalling; this step means testing the trained network for new given data (examples). However, the structure, properties of neurons and training methods are factors that affects classification of neural networks or specify the type of neural network. The most common types of neural network are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

2.4.1 Neural networks types

A Feed-Forward Neural Networks (FFNNs): are the most commonly used type of neural networks. FFNNs consist of three types of layers (inputs layer, hidden layer and output layer). the structure of FFNNs is sorted by the type of layers, such as the first layer is input layer and last layer is the output layer, whereas the middle layers (located between input and output layer) can be called as hidden layers, which can be one or more layers. Moreover, in FFNNs, the neurons are connected to the following layer neurons by one-direction lines (weights). In other words, there is no feed-back connection in FFNN and the neurons of same layer are not connected with each other. The most common types of Feed-Forward neural networks are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

- a) Multilayer perceptron
- b) Radial basis function network

Recurrent neural network: is a less conventional type of neural network. The architecture of this network allows feed-back connection between neurons. Further, minimum amount of feed-back connection between neurons in this network must one feed-back connection.

Also in this network, the neurons of same layer can be connected with each other. The commonly used types of Recurrent neural network are listed below (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Tino et al., 2015; Gurney, 1997).

- a) Hopfield network
- b) Boltzmann machine.

2.4.2 Single layer perceptron

It is artificial neuron model that can be defined as a mathematical model of a biological neuron with several inputs (x_1, x_{j1}) and one single output (y). Furthermore, McCulloch and Pitts model also can be referred as a simple neuron paradigm that gathers input patterns and assign them as input parameters through the associated parameters of the weights. In other words, linear threshold system is a neuron that can operates all the number of inputs from another units and form an actual values, this process is performed in accordance to the activation function. The transfer function performs mapping from the input (real values) to the output (into interval); this mapping can be a linear or nonlinear. The sigmoidal function (hard-limiter) was used in McCulloch and Pitts model as transfer function, which referred by (\emptyset). The synapses in artificial neuron model is referred as weights (w) which is the connection lines between inputs and neuron. Moreover, in McCulloch and Pitts model the values of the weight (w) and threshold (θ) were fixed. Artificial neuron model can easily classify inputs set into two various classes (which means the output is binary). The output (y) in artificial neuron or McCulloch and Pitts model is specified by summation of the dot product between weight and input parameters ($w_i \cdot x_i$) with respect to the activation function \emptyset (Haykin, 2009; Du and Swamy, 2013; Gurney, 1997).

$$\mathbf{N} = \sum_{i=1}^{J1} \mathbf{w}_i \mathbf{x}_i - \boldsymbol{\theta} = \mathbf{w}^T \mathbf{x} - \boldsymbol{\theta} \quad (2.1)$$

$$\mathbf{y} = \emptyset (\mathbf{N}) \quad (2.2)$$

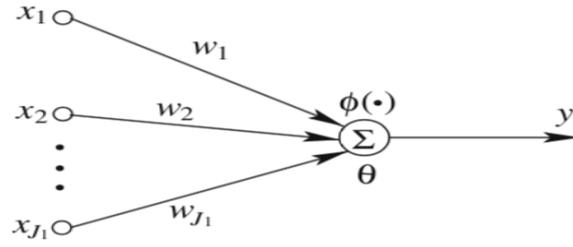


Figure 2.1: Architecture of artificial-neuron model (McCulloch and Pitts model) (Du and Swamy, 2013)

N = network of artificial neuron model, whereas, x is the input parameters.

w represents the weight or the connection lines between inputs and transfer function.

ϕ is the activation function (sigmoidal).

θ is the threshold which is an attribute uses to move the decision boundary away from the origin.

In 1957, the first perceptron (single-layer perceptron paradigm) was developed by Rosenblatt which was inspired by McCulloch & Pitts model and the idea of Hebb (Hebbian learning rule). Rosenblatt's Perceptron model has the capability to classify inputs set into more than two classes unlike artificial neuron (McCulloch & Pitts) model which can only classify inputs set into two classes. In single-layer perceptron model, different activation functions (ϕ) have been used such as a bipolar. Also, the weights (w) and thresholds or biases (θ) is calculated analytically or by a learning algorithm. However, the output (y) of single-layer perceptron can be written as following (Haykin, 2009; Du and Swamy, 2013; Fausett, 1994; Tino et al., 2015).

$$N = \mathbf{w}^T \mathbf{x} - \theta \quad (2.3)$$

$$y = \phi(N) \quad (2.4)$$

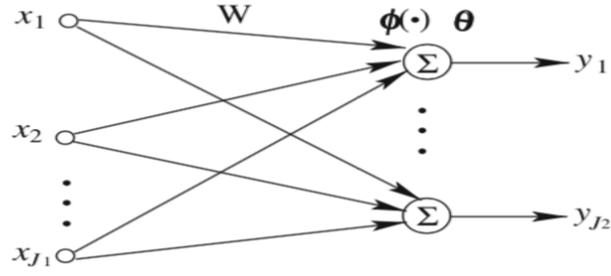


Figure 2.2: Architecture of of Rosenblatt's Perceptron (Du and Swamy, 2013)

Single-layer perceptron has capability only to find solution for linear separable problems. The weight between neurons can be adjusted by using learning algorithm (Rosenblatt's perceptron convergence theorem) and this can be driven through error equation ($E_{t,j}$). Moreover, the learning algorithm of perceptron can be written as following:

$$N_{t,j} = \sum_{i=1}^{J1} x_{t,i} w_{ij}(t) - \theta_j = \mathbf{w}_j^T \mathbf{x}_t - \theta_j \quad (2.5)$$

$$'y_{t,j} = \begin{cases} 1 & N_{t,j} > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (2.6)$$

$$\mathbf{E}_{t,j} = \mathbf{y}_{t,j} - 'y_{t,j} \quad (2.7)$$

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_{ij}(t) - n x_{t,i} \mathbf{E}_{t,j} \quad (2.8)$$

N = network of single-layer perceptron, whereas, $x_{t,i}$ is the i th input of the t th example.

w_{ij} is the i th weigh at the t th node (stand for connected lines between neurons).

θ is the bias or threshold for neuron. while, ϕ is the transfer or activation function.

$E_{t,j}$ is denote to the error.

$y_{t,i}$ is referred to the real output (desired).

$'y_{t,i}$ is the actual output (predicted from network) (Du and Swamy, 2013).

2.5 Backpropagation Algorithm

It is a well-known and widely used training rule which is type of supervised learning. It is delta rule generalization which also referred as Least Mean Squares Algorithm (LMS). This algorithm aims to reduce the cost function analogous to the mean square error among the real and predicted output values through using gradient- descent method. In Back propagation algorithm, at the beginning of first epoch, the input layer in network is fed by the input pattern and then the output is produced. The error (the difference between target and actual value) propagates to backward and thus a blocked-loop hold system is formed. The gradient-descent algorithm is used to modify the weights. The activation function plays important role in allowing to back-propagation rule to be applied. The error can be calculated by using mean square error MSE equation.

$$E = \frac{1}{M} \sum_{z=1}^M E_z = \frac{1}{2M} \sum_{z=1}^M \|Y_Z - 'Y_Z\|^2 \quad (2.9)$$

$$E_z = \frac{1}{2} \|Y_Z - 'Y_Z\|^2 = \frac{1}{2} e_z^T e_z \quad (2.10)$$

$$e_z = Y_Z - 'Y_Z \quad (2.11)$$

The Error (E) is reduced by employing gradient-descent which allows to the weights to be adjusted. This can be done using below equation.

$$\Delta_z \mathbf{W} = -\eta \frac{\partial E_z}{\partial \mathbf{W}} \quad (2.12)$$

η is referrers to rate of learning and represents our step size which ranged between (0-1) and this can be chosen manually. \mathbf{W} is representing the parameters of networks such as weights and bias. Furthermore, equation (22) referred back-propagation algorithm.

Moreover, the algorithm can be better through involve using of (μ) momentum factor which analyze and the provide status for convergence (Haykin, 2009; Du and Swamy, 2013; Kriesel, 2007; Xiao, 1996; Tino et al., 2015; Shwartz and David, 2014).

$$\Delta_z(t) \mathbf{W} = -n \frac{\partial E_z}{\partial \mathbf{W}} + \mu \Delta \mathbf{W}(t - 1) \quad (2.13)$$



Figure 2.3: Effects of learning rate and momentum parameters on weight updating

2.6 Convolutional Neural Networks

Convolutional neural network (CNN) has been employed successfully for several tasks in machine vision (Simonyan & Zisserman, 2014). Generally, the CNN relies on architectural features which include the receptive field, weight sharing and pooling operation to take into account the 2D characteristic of structured data such as images. The concept of weight sharing for convolution maps drastically reduce model parameters; this has the important implications that the model is less prone to over-fitting as compared to fully connected models of comparable size. The pooling operation essentially reduces the spatially dimension of input maps and allow the CNN to learn some invariance to moderate distortions in the training; this feature enhances the generalization of the CNN at test time as model is more tolerant to moderate distortion in the test data (Szegedy et al., 2015).

The typical CNN is shown in Figure 2.4. Essentially, convolution layers, pooling layers and the fully connected layer are shown. For example, layer 1 employs n convolution filters of size $a \times a$ to generate a bank of n convolution maps (C1) of size $i \times i$; this is

followed by a pooling (sub-sampling) operation on the convolution maps with a window size of $b \times b$. Therefore, pooling layer (S1) compose n feature maps of size $j \times j$; where, $j = i/b$ (He et al., 2016). The convolution layer performs feature extraction on the incoming inputs via a convolution filter of specified size.

The pooling operation pools features across input maps using a window of specified size; common pooling operations used in applications are the average and max pooling (Rios & Kavuluru, 2014). In average pooling, the average value of the inputs captured by the pooling window is taken; while, in max pooling, the maximum value of the inputs captured by the pooling window is taken.

For learning the classifier model, features are forward-propagated through the network to the fully connected layer with an output layer of Softmax units. Then, the backpropagation learning algorithm can be employed to update the model parameters via gradient descent update rule.

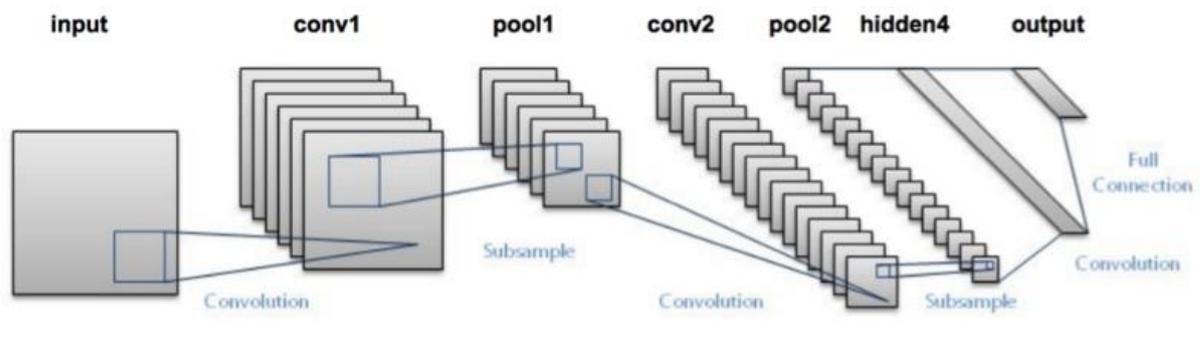


Figure 2.4: The LeNet-5 Architecture, a convolutional neural network.

The advancement in computer industries has motivated the researchers to further improve the performance of the CNN by making it deeper and more feasible. Therefore, a CNN of 19 layers was proposed and called VGG-Net (Simonyan & Zisserman, 2014). Also, Szegedy et al. (2015) proposed a 22 layers deep network named GoogLeNet which also includes an improvement in the architecture and working principles of the CNN by adding an inception module to it. Moreover, a CNN of 152 layers named ResNet (ResNet-152) was proposed by He et al. (2016).

Convolutional layers in a convolutional neural network systematically apply learned filters to input images in order to create feature maps that summarize the presence of those features in the input.

Convolutional layers prove very effective, and stacking convolutional layers in deep models allows layers close to the input to learn low-level features (e.g. lines) and layers deeper in the model to learn high-order or more abstract features, like shapes or specific objects.

A limitation of the feature map output of convolutional layers is that they record the precise position of features in the input. This means that small movements in the position of the feature in the input image will result in a different feature map. This can happen with re-cropping, rotation, shifting, and other minor changes to the input image.

A common approach to addressing this problem from signal processing is called down sampling. This is where a lower resolution version of an input signal is created that still contains the large or important structural elements, without the fine detail that may not be as useful to the task.

Down sampling can be achieved with convolutional layers by changing the stride of the convolution across the image. A more robust and common approach is to use a pooling layer.

A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer; for example the layers in a model may look as follows:

1. Input Image
2. Convolutional Layer
3. Nonlinearity
4. Pooling Layer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model.

The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps.

Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2×2 pixels applied with a stride of 2 pixels.

This means that the pooling layer will always reduce the size of each feature map by a factor of 2, e.g. each dimension is halved, reducing the number of pixels or values in each feature map to one quarter the size. For example, a pooling layer applied to a feature map of 6×6 (36 pixels) will result in an output pooled feature map of 3×3 (9 pixels).

The pooling operation is specified, rather than learned. Two common functions used in the pooling operation are:

- Average Pooling: Calculate the average value for each patch on the feature map.
- Maximum Pooling (or Max Pooling): Calculate the maximum value for each patch of the feature map.

2.7 Transfer Learning

In medical image analysis and processing, a most common issue is that the number of available data for research purposes is limited and small. Hence, training a fully deep network structure like CNN with small number of data may result in Overfitting, which is usually the reason of low performance and generalization power (Long et al., 2015).

Transfer learning is one solution of this problem, by sharing the learned parameters of effective and well-trained networks on a very large dataset. The concept of transfer learning is the use of a pre-trained model that is already trained on large datasets, and transfer its pre-trained learning parameters, in particular weights, to the target network model (Cheng & Malhi, 2017). The last fully connected layers are then trained with initial random weights on the new dataset. Note that, although the dataset is different than the one the network was trained

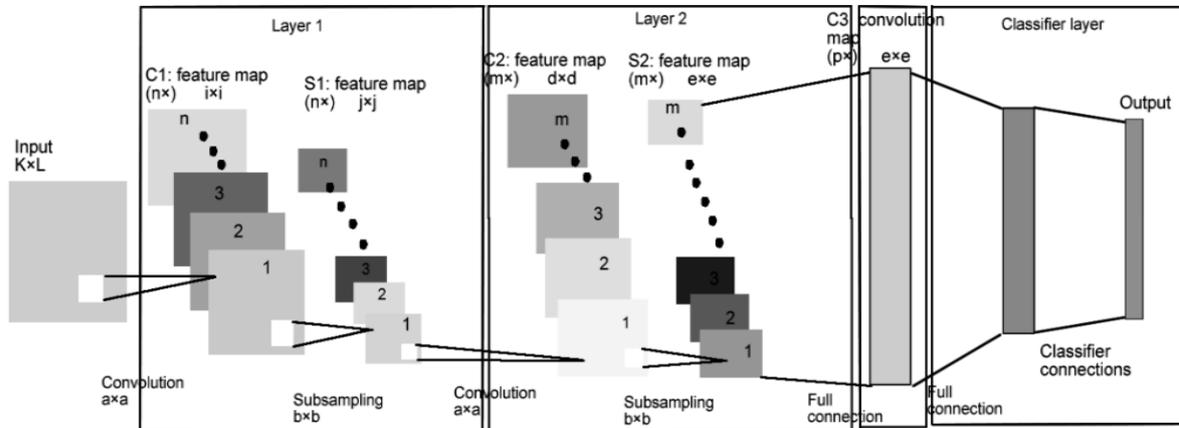


Figure 2.5: A Full Convolutional Neural Network Layers (LeNet)

on, the low-level features are similar. Thus, the parameter's transfer of the pre-trained model may provide the new target model with a powerful feature extraction capability and reduce its training computations speed and memory cost.

Transfer learning has been used extensively in medical imaging and it showed a great efficacy in terms of accuracy, training time, and error rates (Lei et al., 2018). In this research, one different pre-trained model has been employed for the classification of Malaria cells into 2 different classes. This convolutional neural network is: GoogleNet.

2.7.1 GoogleNet (Inception)

GoogleNet is a deep convolutional neural network that was proposed at ILSVRC in 2014 (a.k.a. Inception V1) from Google (Szegedy et al., 2015). This network was able of achieving the least top-5 error rate of 6.67%. This error was very close to the one achieved by humans when they were forced to have the same challenge. GoogleNet architecture is inspired by the typical CNN "LeNet"; however, this network has more features and novel elements such as the inception module found throughout the network layers. GoogleNet, uses also batch normalization, RMSprop, and image distortions just like other networks. The most important and different part and improvement of this network compared to other deep convolutional neural networks is the inception module. This module is a combination of different small convolutions that are done throughout the network layers in order to

reduce the number of hyperparameters to 4 million. Note that number of parameters of previous network like AlexNet was 60 million.

The network architecture is seen in Figure 2.12. As seen the network consists of 22 layers which makes very deep but with less number of parameters compared to other networks due to its inception modules.

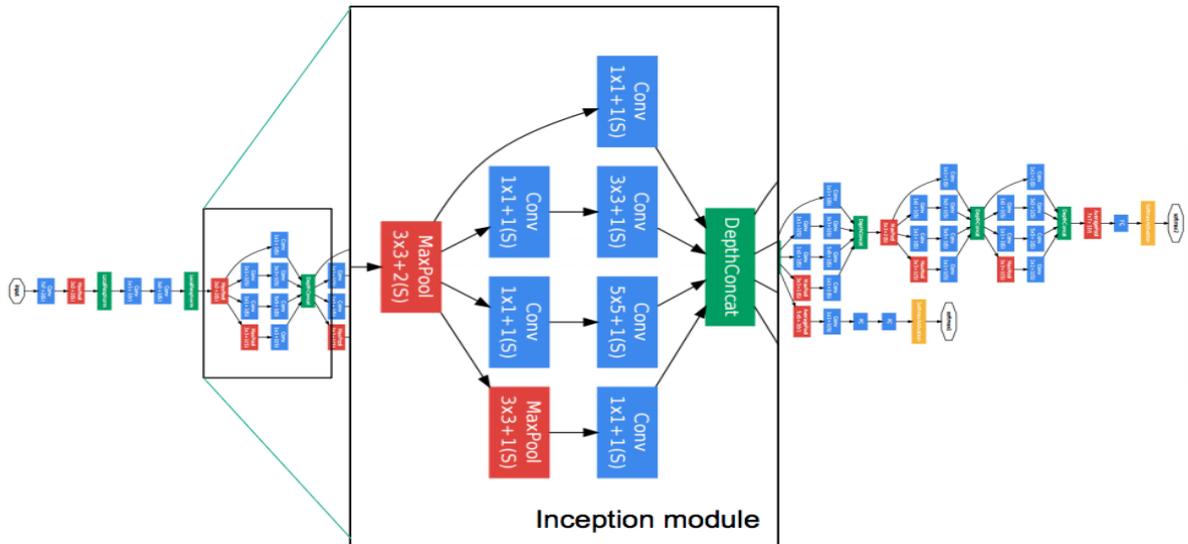


Figure 2.6: GoogleNet architecture and its inception module (Szegedy et al., 2015)

2.7.2 Inception Module

The Inception Module is based on a pattern recognition network which mimics the animal visual cortex. After presenting several examples of images, the network gets used to small details; middle sized features or almost whole images if they come up very often (Szegedy et al., 2015). Each layer of the deep network reinforces some features it thinks is there and passes on to the next. If it has been trained to recognize faces, the first layer detects edges, the second overall design, the third eyes, mouth, nose, the fourth the face, the fifth the mood, for instance.

Working with inception means they do not feed the trained network a real image, but random noise. It tries to Figure out if there is something it is acquainted with in this image.

Then, they backpropagate the firing force of the last layer reconstructing an interpreted version of the input bits. This new image is then presented to the network again and the process is iterated. It works exactly like humans looking at the clouds and finding sheep, faces, monsters. It's pattern reinforcement.

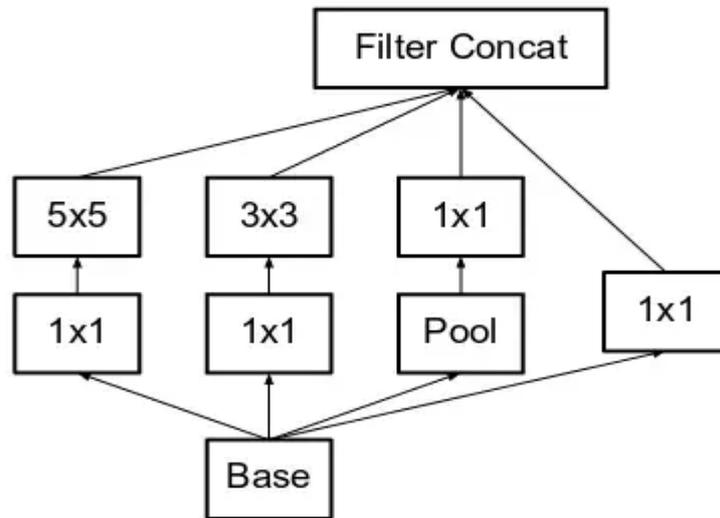


Figure 2.7: Inception module in GoogleNet (Szegedy et al., 2015)

2.7.3 AlexNet

AlexNet is the first convolutional neural network that achieved the highest classification accuracy at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 (Krizhevsky et al., 2012). This deep structure is comprised of eight main layers; the first five layers are mainly convolutions, while the last three are fully connected layers. Each convolutional layer is followed by an activation function layer, i.e. Rectified Linear Units layer (ReLU), proposed to improve the performance of the network by making the training faster than equivalents of “tanh” activation functions. After each convolution layer, a max-pooling is used in AlexNet, in order to reduce the network size.

Moreover, a dropout layer is added after the first two fully connected layer which helps to reduce the number of neurons and prevent overfitting (Srivastava et al., 2014). Finally, a

Softmax layer is added after the last layer to classify the input given data. Figure. 2.8 shows the structure of the AlexNet.

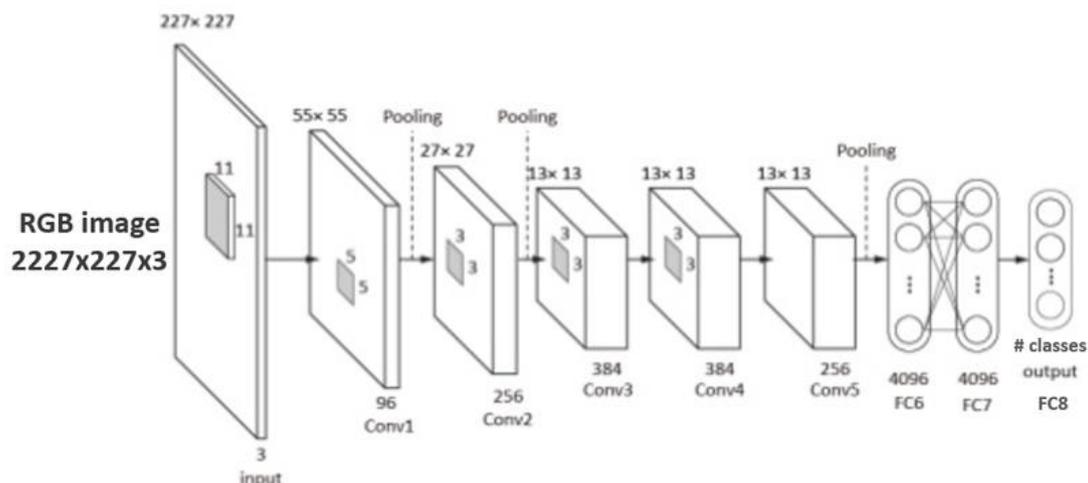


Figure 2.8: Blockdiagram of AlexNet

2.7.3 VGG-16

The VGG-16 is deep convolutional neural network that was proposed at ILSVRC in 2014 (Simonyan et al., 2014), and was able to achieve the least error rate. This network consists of 16 main layers, among them 13 are convolutional layers while the remaining are fully connected layers. Unlike the AlexNet, all of the convolutions layers of VGG-16 have the same filter size. Moreover, ReLU layers, max pooling layers, fully connected layers, and dropout layers are also used in the VGG-16. Figure 2.9 shows the structure of the VGG-16.

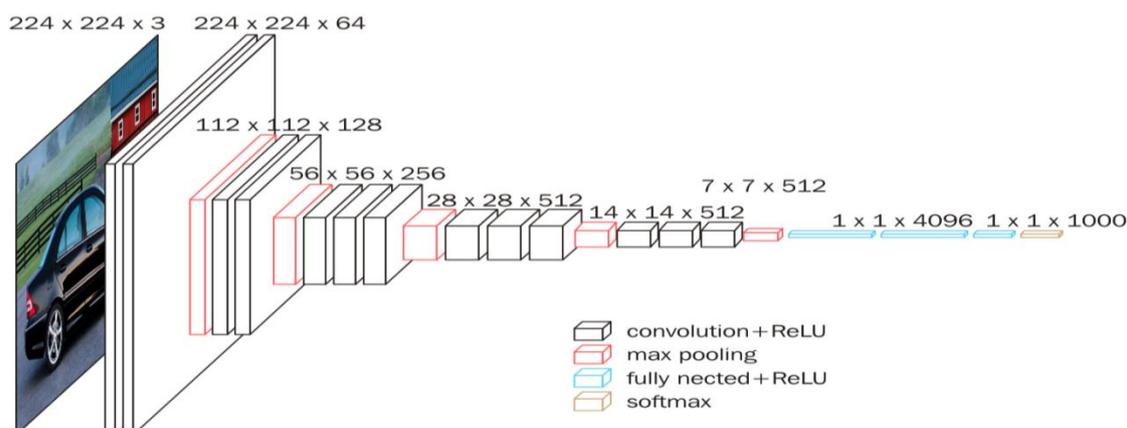


Figure 2.9: VGG-16 Model Architecture

CHAPTER 3

MATERIALS AND METHODS

3.1 Methodology

Image processing and manual features extraction of images can be so complex and time consuming. Therefore, there was a big need of networks that can extract features from images automatically through its layers. This was the motivation of creating deep learning networks. The depth of networks is made to extract low and high level features without any feature engineering techniques.

Many deep networks were created, however, the best deep network for features extraction from domain space is the deep convolutional neural network (DCNN). This is due to its depth in which convolutions, pooling, regularization, and normalization are applied to images, which allows the extraction of different levels of abstractions of input data.

In practice, the training of deep convolutional neural networks created from scratch is a tedious task. This is because CNNs are deep, which means many hyperparameters to be trained in addition to filters learning and weights update and calculation of errors which requires long time. Moreover, CNNs need large datasets in order to be trained and to not overfit. This can be an issue since it is relatively difficult to find some large datasets especially in the medicine field.

Recently, deep networks architectures are presented. Those networks are convolutional neural networks with different architectures and number of layers such as AlexNet, VGG-NET, GoogleNet etc... These networks are trained on ImageNet; a public dataset of millions of general images used to train the new models to classify 1000 classes. After training those models have obtained great generalization capabilities in classifying 1000 objects.

Researchers found that the solution of these problems is to fine-tune those pre-trained models instead of training CNNs from scratch. They also found that this storing of knowledge gained by a network can lead to better results as the pre-trained models are

trained on millions of data and gained great features extraction capabilities due to their learned filters.

Transfer learning is to use a pre-trained model in order to fine-tune it to classify a new task in addition to the images it was trained to classify. This is achieved by storing its knowledge gained by training it on one problem and applying it to classify new images using its trained filters and parameters (Figure 3.1).

In this thesis, transfer learning is applied to solve the problem of Malaria cells classification into 2 types. GoogleNet is fine-tuned in this work in order to add a new classification task to its functions, as shown in Figure 3.1 and 3.2. This network contains 3 classification layers named as Loss1, Loss2, and Loss3. Those are fully connected layers and they are retrained during fine-tuning of the network to its new classification task which is Malaria cells classification.

Figure 3.1 shows the fine-tuning process of GoogleNet to be trained to classify Malaria cells. As seen in the Figure, all layers are fixed except the last three layers which are classification layers. Those three layers are retrained as they represent the fully connected layer, which is a traditional feedforward neural network. Those three layers are retrained using backpropagation learning algorithm in which error is reduced and weights are updated until the network reaches a minimum square error with a high classification rate.

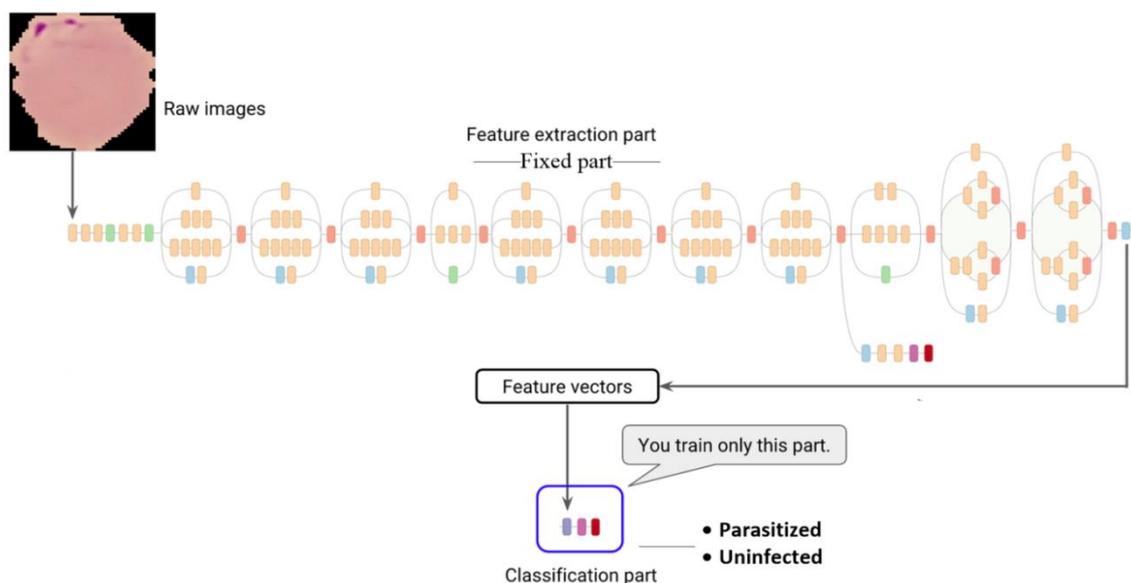


Figure 3.1: Fine-tuning of GoogleNet to classify Malaria cells

Figure 3.2 shows a block diagram of the whole process of fine-tuning GoogleNet to classify malaria into infected and uninfected images. As seen, images are first used for fine-tuning the network but first pre-trained weights and parameters should be frozen as they are already well trained using millions of images; which is the main aim of transfer learning. As seen in the block diagram, network is first trained for its new task; which is here Malaria classification and then it is tested using new unseen images in order to measure its capability of generalizing accurate diagnosis of new images.

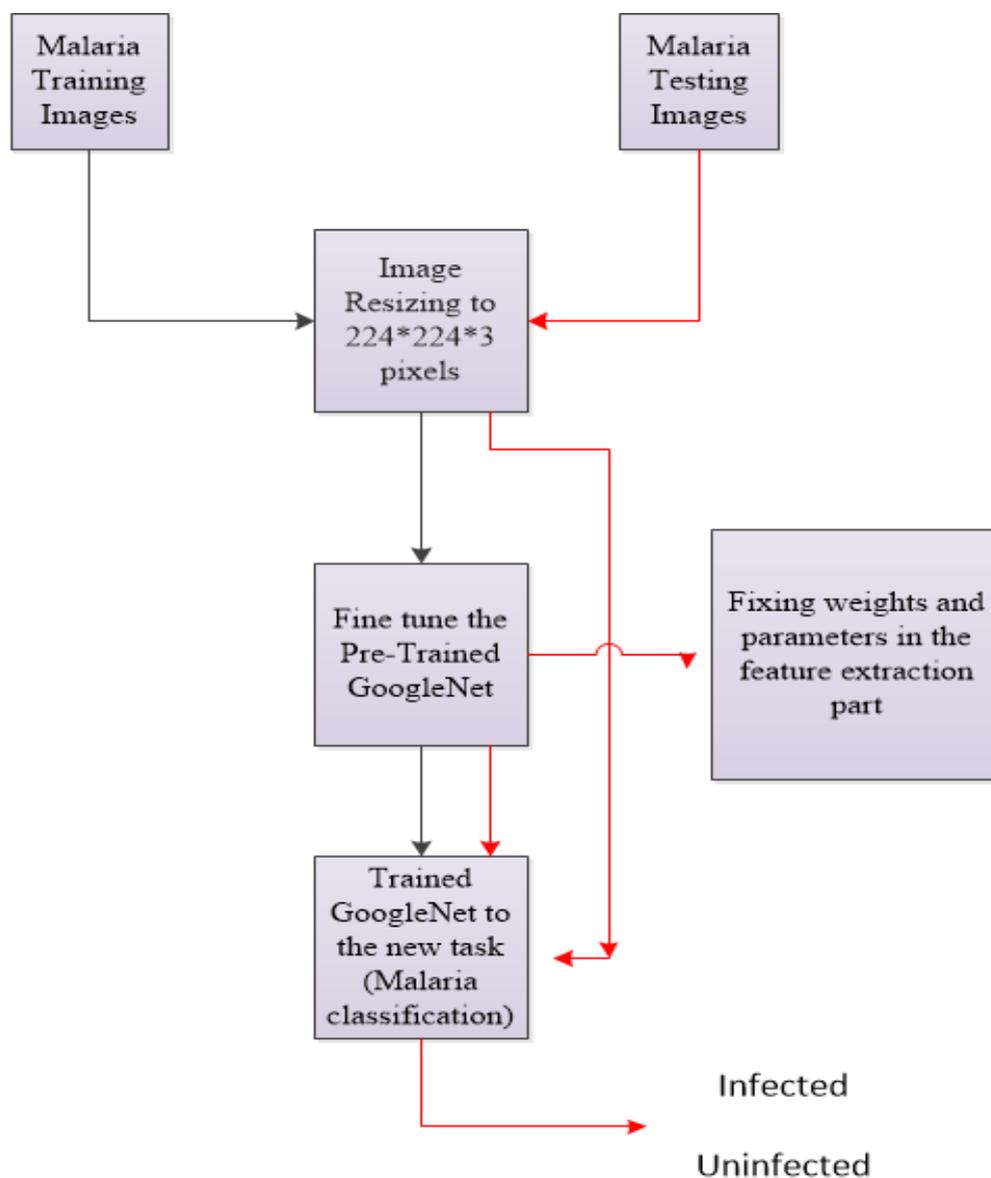


Figure 3.2: Block diagram of the whole training and testing process of GoogleNet

3.2 Data

The employed model is trained and tested using Malaria cells images collected from a public database. This dataset is taken from kaggle (Kaggle, 2017) which is an online community and environment for machine learning researchers and a place for machine learning competitions. This dataset contains 27558 images of Malaria cells which includes infected and uninfected cells with their cell types or labels. The dataset is divided into approximately 13779 images of each type of the 2 different blood cells (Table 3.1).

Table 3.1: Dataset description

Total Number of data	Infected	Uninfected
27558	13779	13779

Figure 3.4 shows a sample of the dataset images of the 2 different normal and Malaria blood cells.

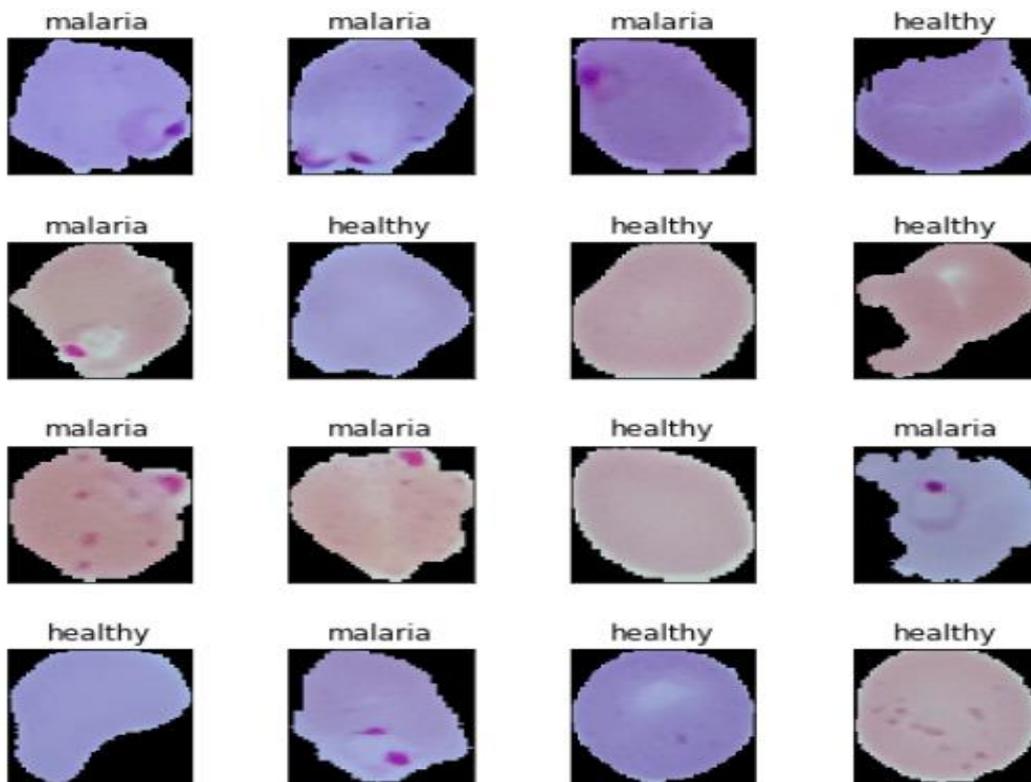


Figure 3.3: Sample of dataset images of the two different Malaria cells

3.2 Training of Pre-trained Model GoogleNet

In this work, the employed model is trained and tested using Matlab environment. The networks were simulated on a Windows 64-bit desktop computer with an Intel Core i7 4770 central processing unit (CPU) and 8 GB random access memory. Note that there was no graphical processing unit (GPU) available in the used desktop.

The pre-trained model used in this research were trained and tested on a ratio of 60% of the available data. The performance of the networks was then evaluated using a held out test set of the remaining 40% of the data. Note that images were all resized in order to fit the GoogleNet input which requires input images to be of size 224*224 pixels. Loss and accuracy of each model were calculated as follows:

$$\text{Loss} = -(1/n) \sum_{i=1}^n \log(P(CC)) \quad (3.1)$$

$$\text{Accuracy} = \frac{CC}{T} \quad (3.2)$$

where $P(CC)$ is the probability of the correctly classified images, n is the number of images, and T is the total number of images during the training and/or testing phases.

GoogleNet is a pre-trained model architecture used in this research. It is a convolutional neural network winning in the ILSVRC 2014 competition. As shown in Figure.3.2, the network mainly consists of 22 layers including inception modules which are used to reduce number of parameters of the network to 4 million. The last three layers of the network are the fully connected layers and they are denoted as Loss1, Loss2, Loss3. At the last layer, there is Softmax activation function that is used to show the output as probabilities.

Note that the publicly available weights of the network trained against the ILSVRC14 are used in this transfer learning based research. As we are using a pre-trained model, its final

fully connected layer Loss3 was removed and a new layer was added and it has 2 output neurons corresponding to the 2 Malaria cells categories. Note that the weights of this layer are initialized at random. On the other hand, the other layers are remained in the network but their weights were frozen to act as a feature extractor. These weights are already trained on millions of images to extract high level features of the input data. For training, a batch size of 200 images for each iteration is used via stochastic gradient descent (Wijnhoven & Dewith, 2010). Also, the learning rates for the fully connected Loss1, Loss2, and Loss3 layers were fixed at 0.001, 0.001, and 0.01, respectively during training. Consequently, this allows the network to learn faster for the final fully connected layer. Moreover, the network is fine-tuned using 60% of the available data, and 20 epochs are set to train the network.

Table 3.2: GoogleNet learning parameters

GoogleNet	
Learning parameters	Values
Training ratio	60%
Learning rates (Loss1, Loss2, Loss3)	0.001, 0.001, 0.01
Number of epochs	20
Training accuracy	97.5%
Training time	1.5 hours
Achieved mean square error (MSE)	0.017

Table 3.2 shows the learning parameters of the GoogleNet model. It can be seen that the network has achieved a training accuracy of 97.5% in approximately 1.5 hours and 20 epochs.

Figure 3.4 shows the variations of accuracy in terms of epochs increase. It is seen that the network' accuracy was improving with the increase of epochs during training and testing until a minimum square error and accuracies of 97.5% and 95% are achieved, respectively.

This achieved accuracy can be considered as good and also, the network required relatively good time to achieve such results.

3.3 Testing of GoogleNet

Once trained, GoogleNet was tested on 40% of the available data which includes the 2 types of Malaria cells. Figure 3.5 shows the testing accuracy variations in terms of epochs increasing. It can be seen that the network performed well in testing in which it was capable of reaching an accuracy of 95%

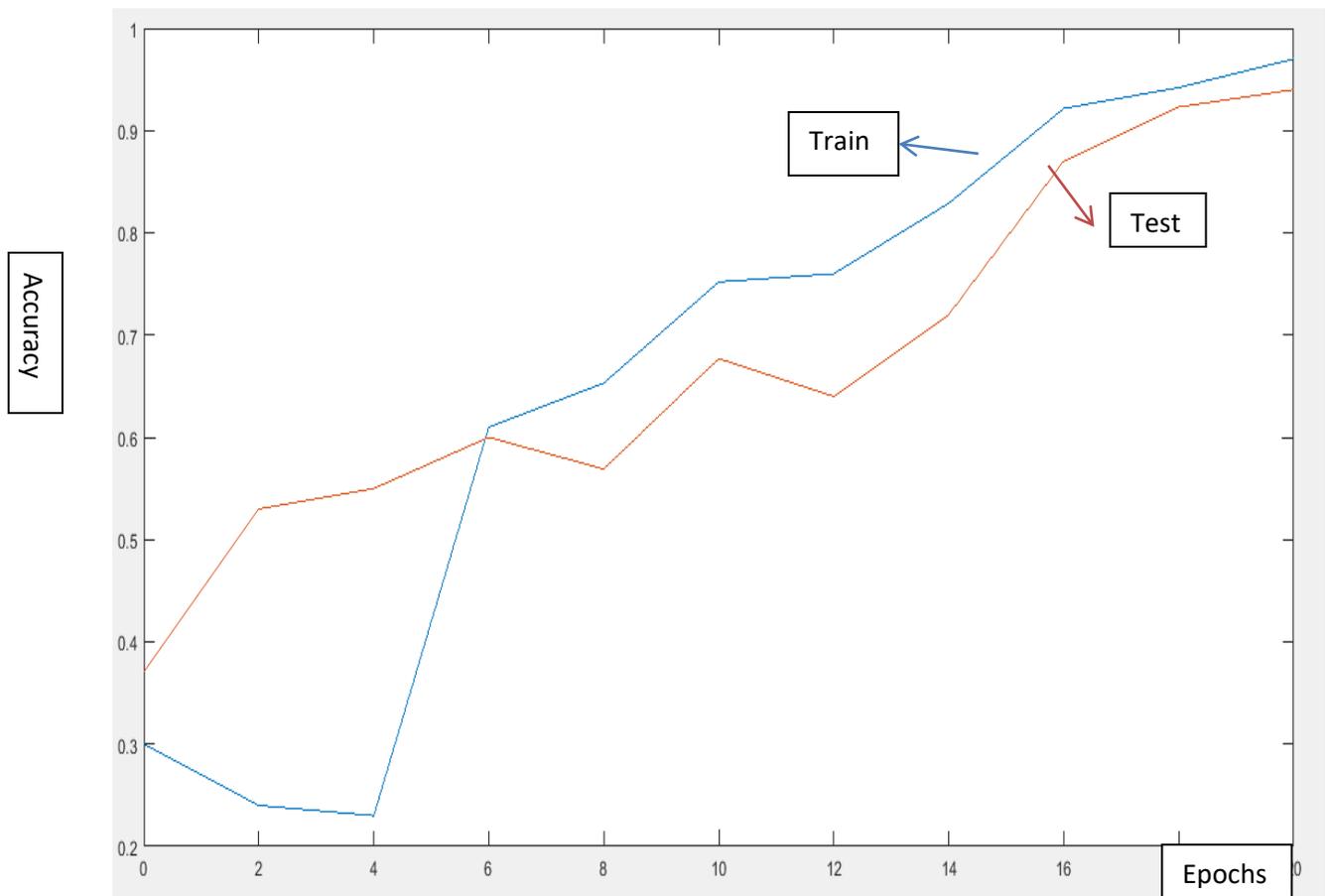


Figure 3.4: Accuracy variations with the change of Epochs

Table 3.3 shows the results of the performance of the fine-tuned and tested GoogleNet during both training and testing.

Table 3.3: Testing performance of GoogleNet

GoogleNet	Data Number	Accuracy
Training Accuracy	60% of data	97.5%
Testing Accuracy	40% of data	95%

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results and Discussion

Upon training, the employed pre-trained model is tested on 40% of the available data. Table 4.1 shows the performances of the model during training and testing. As seen, GoogleNet achieved 97.5% training accuracy; however, it was not capable of achieving such accuracy during testing, where it scored a lower recognition rate of 95%. On the other hand, this testing accuracy is also satisfactory as this classification task is tedious since there is a similarity between all Malaria cells. This might have made the network fall into local minima. Moreover, GoogleNet required 20 epochs to achieve such accuracy, which is relatively good to achieve such accuracy and a minimum square error of 0.017. In contrast, to achieve this accuracy and to reach that small error the network required a long training time of 1.5 hours in order to converge and fine-tune. This time is obviously due to the depth of network as it contains many hidden layers. Also, it is because of the number of images which can be considered large number.

Figure 3.4 shows the learning curve of the fine-tuned GoogleNet. This Figure shows the accuracy variations with respect to the Epochs increasing during training and testing of network. It can be seen that network was trained well; however, the increase of depth of GoogleNet makes it more difficult to train, i.e. it required longer time and more epochs to reach the minimum square error (MSE) and converge. Furthermore, it is important to mention that this difference in time and epochs number of GoogleNet ended up with a low MSE and good accuracy.

Table 4.1: Performances of the model during training and testing

Number of images	Number of Epochs	Error reached (mse)	Training time	Training accuracy	Testing accuracy
60% of data (train)	20	0.017	1.5 hours	97.4%	95%
40% of data (test)					

For more understanding of the network learning performance and to have insight into the different levels features learned by the employed models, we sought to visualize the learned kernels or features in the convolutional layers.

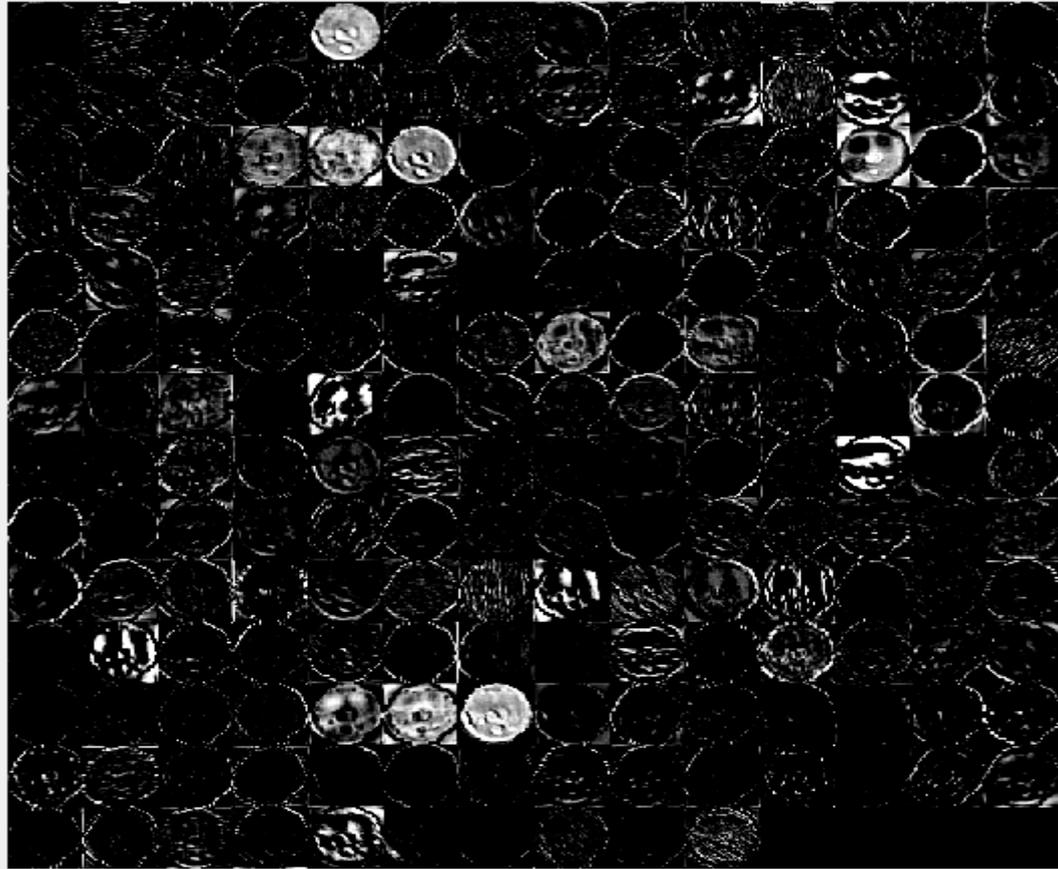


Figure 4.1: Learned filters of GoogleNet This image is produced by the network and it shows the learned feature maps of the network. those images represents different parts of images like corners, edges, objects etc..

Figure 4.1 shows the visualizations of extracted features of Malaria images by the GoogleNet pre-trained model. It is seen that different levels of abstractions are extracted during each layer which helps the network in learning the exact and appropriate features that distinguish the two different classes of infected and uninfected Malaria cells.

4.2 Performance Evaluation Metrics

These metrics are derived from classification of the tested sampling images, as shown in Table 4.2 its being derived by a contingency table which is called confusion matrix, Accuracy indicates the percentage of rightly classified image samples, without considering their class labels. For a binary classification that concludes on positive and negative

classes, Sensitivity is the percentage of correctly classified samples, Specificity is the number of correctly negative samples classified.

$$Accuracy = \frac{(TP + TN)}{TN + TP + FP + FN}$$

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

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

Table 4.2: Performance metrics of the networks

Network Model	GoogleNet
Accuracy	95%
Sensitivity	95%
Specificity	96%
Misclassified	7%

Figure 4.2 shows the ROC curve which is the model selection metric for bi–multi class classification problem. This curve is a probability curve for the two classes of the proposed classification system. ROC shows how good the model is for distinguishing the given classes, in terms of the predicted probability.

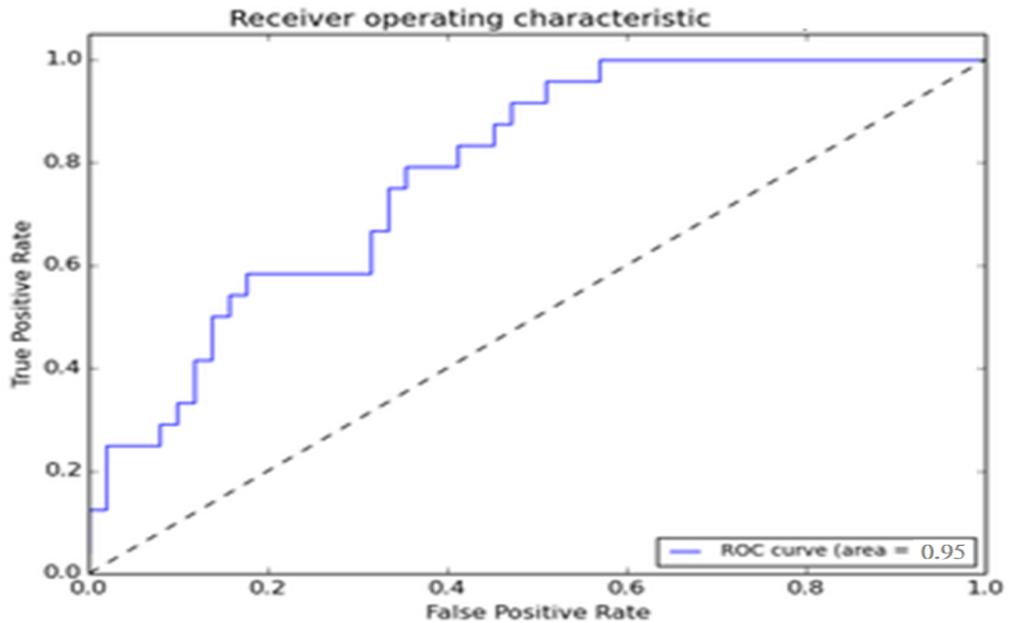


Figure 4.2: ROC Curve

4.3 Results Comparison

A comparison of the developed networks employed in this work with some earlier works is shown in Table 4.3. Firstly, it is seen that the employed pre-trained CNNs achieved high recognition rates compared to other deep networks, which is obviously due to their powerful efficiency in extracting the important features from input images. The pre-trained convolutional neural networks (GoogleNet) employed within this work achieved higher accuracies than other earlier work that used a conventional neural network (Yu et al., 2017), which was built from scratch. Furthermore, it is important to note that the networks gained a better generalization capability compared to those other methods and networks used for Malaria cells classification such as BPNN (Das et al., 2011) and other machine learning and image processing techniques used in (Das et al., 2013).

It is also remarkable that our method achieved higher accuracies than other related works which also used convolutional neural networks such as in (Kaewkamnerd et al., 2012) and (Sorgeddrager, 2018).

Table 4.3: Results comparison with other works

	Classification objectives	Classifier used	Accuracy
(Das et al., 2011)	2 types (Infected and uninfected)	Backpropagation neural network	88.78%
(Das et al., 2013)	2 types (Infected and uninfected)	SVM	84%
Our method	2 types (Infected and uninfected)	GoogleNet	95%

	Classification objectives	Classifier used	Accuracy
(Kaewkamnerd et al., 2012)	2 types (Infected and uninfected)	CNN based device his own design	90%
(Sorgedragger, 2018)	2 types (Infected and uninfected)	CNN his own design	92%
Our method	2 types (Infected and uninfected)	GoogleNet	95%

CHAPTER 5

CONCLUSION

In this research, transfer learning based neural networks were employed. GoogleNet is used in this research to be the classifier of this work. Transfer learning is used in this work due to the power of pre-trained models in extracting features and achieving high classification rates. Their features learned on a source task are transferred to a new task, Malaria cells dataset, in order to learn the classification of Malaria cells into infected and uninfected. In order to fine-tune GoogleNet, its features extraction layers were all frozen so that their learned filters can be used. However, its last three layers were retrained since these are the classification layers and they should be learned to classify the Malaria cells into 2 types.

GoogleNet was fine-tuned on 60% of the data and tested using the remaining 40%. During training network achieved an accuracy of 97.4%, while 95% was achieved during testing

It was concluded that GoogleNet, a complex very deep architecture of achieved a significantly higher classification accuracy when distinguishing between normal and abnormal Malaria cells images, as compared to that of other methods such as CNN created from scratch, SVM, and decision tree. Furthermore, GoogleNet network learned features visualization demonstrates that mid and high level features are learned effectively by the model.

Overall, it can be stated that the transfer of knowledge from a well-trained convolutional network to extract the rightful features and an accurate of identification of new unseen Malaria images, is possible. Thus, it can be stated that the GoogleNet can be a good classifier for the Malaria classification task, with a small margin of errors.

REFERENCES

- Pandit, P., & Anand, A. (2016). Artificial neural networks for detection of malaria in RBCs. *arXiv preprint arXiv:1608.06627*.
- Anand, A., Chhaniwal, V. K., Patel, N. R., & Javidi, B. (2012). Automatic identification of malaria-infected RBC with digital holographic microscopy using correlation algorithms. *IEEE Photonics Journal*, 4(5), 1456-1464.
- Parveen, R., Jalbani, A. H., Shaikh, M., Memon, K. H., Siraj, S., Nabi, M., & Lakho, S. (2017). Prediction of Malaria using Artificial Neural Network. *International Journal of Computer Science and Network Security*, 17(12), 79-86.
- Vijayalakshmi, A. (2019). Deep learning approach to detect malaria from microscopic images. *Multimedia Tools and Applications*, 1-21.
- Dong, Y., Jiang, Z., Shen, H., Pan, W. D., Williams, L. A., Reddy, V. V., ... & Bryan, A. W. (2017, February). Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)* (pp. 101-104).
- Pan, W. D., Dong, Y., & Wu, D. (2018). Classification of Malaria-Infected Cells Using Deep Convolutional Neural Networks. In *Machine Learning-Advanced Techniques and Emerging Applications*. IntechOpen.
- Das, D. K., Ghosh, M., Pal, M., Maiti, A. K., & Chakraborty, C. (2013). Machine learning approach for automated screening of malaria parasite using light microscopic images. *Micron*, 45, 97-106.
- Das, D., Ghosh, M., Chakraborty, C., Maiti, A. K., & Pal, M. (2011, November). Probabilistic prediction of malaria using morphological and textural information. In *2011 International Conference on Image Information Processing* (pp. 1-6). IEEE.

- Kaewkamnerd, S., Uthaipibull, C., Intarapanich, A., Pannarut, M., Chaotheing, S., & Tongshima, S. (2012, December). An automatic device for detection and classification of malaria parasite species in thick blood film. In *Bmc Bioinformatics*(Vol. 13, No. 17, p. S18). BioMed Central.
- Sorgedragger, R. (2018). Automated malaria diagnosis using convolutional neural networks in an on-field setting. *MA, Delft University of Technology*.
- M. I. Razzak, “Malarial parasite classification using recurrent neural network,” vol. 9, no. 2, pp. 69–79, 2015.
- World Health Organization, “Disease burden of malaria,” <http://www.who.int/mediacentre/factsheets/fs094/en/>.
- Kaggle, <https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria>
- F. B. Tek, A. G. Dempster and I. Kale, “Parasite detection and identification for automated thin blood film malaria diagnosis,” *Journal of Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 21–32, January 2010.
- D. K. Das, M. Ghosha, M. Pal, A. K. Maiti, and C. Chakraborty, “Machine learning approach for automated screening of malaria parasite using light microscopic images,” *Journal of Micron*, vol. 45, pp. 97–106, Feb 2013.
- Cheng, P. M., & Malhi, H. S. (2017). Transfer learning with convolutional neural networks for classification of abdominal ultrasound images. *Journal of digital imaging*, 30(2), 234-243.
- Elsalamony, H. A. (2016). Detection of some anaemia types in human blood smears using neural networks. *Measurement Science and Technology*, 27(8), 085401.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115.

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). Berlin: Germany.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105). Nevada: USA.
- LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.
- Lei, H., Han, T., Zhou, F., Yu, Z., Qin, J., Elazab, A., & Lei, B. (2018). A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning. *Pattern Recognition*, 79, 290-302.
- Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.
- Manik, S., Saini, L. M., & Vadera, N. (2016, July). Counting and classification of Malaria cell using Artificial Neural Network (ANN). In *Proceedings of the IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems* (pp. 1-5). Delhi Technological University: India.
- McKenna, S. J., Ricketts, I. W., Cairns, A. Y., & Hussein, K. A. (1993, May). A comparison of neural network architectures for cervical cell classification. In *Proceedings of the Third International Conference on Artificial Neural Networks* (pp. 105-109). Lodz: Poland.
- Preston, K. (1987). High-resolution leukocyte analyzers: retrospective and prospective. *Applied optics*, 26(16), 3258-3265.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211-252.

- Shitong, W., & Min, W. (2006). A new detection algorithm (NDA) based on fuzzy cellular neural networks for Malaria cell detection. *IEEE Transactions on information technology in biomedicine*, 10(1), 5-10.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sobrevilla, P., Montseny, E., & Keller, J. (1999, July). Malaria cell detection in bone marrow images. In *Proceedings of the 18th International Conference of the North American* (pp. 403-407). Barcelona: Spain.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015, June). Going deeper with convolutions. *Proceedings of the International Conference on Computer Vision and Pattern Recognition* (pp. 1-9). Boston: USA.
- Ushizima, D. M., Lorena, A. C., & De Carvalho, A. C. P. L. F. (2005, November). Support vector machines applied to Malaria cell recognition. In *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems* (pp. 6-39). San Diego: California.
- Wijnhoven, R. G., & de With, P. H. N. (2010, August). Fast training of object detection using stochastic gradient descent. In *Proceedings of the 20th International Conference on Pattern Recognition* (pp. 424-427). Istanbul: Turkey.
- Xiaomin, Y., Limin, L., & Yu, W. (1994). Automatic classification system for leukocytes in human blood. *Chinese Journal of Computers*, 2, 130–136.
- Yu, W., Chang, J., Yang, C., Zhang, L., Shen, H., Xia, Y., & Sha, J. (2017, October). Automatic classification of leukocytes using deep neural network. In *Proceedings of the 12th International Conference on ASIC* (pp. 1041-1044). Berlin: Germany.

APPENDIX 1

GOOGLENET SOURCE CODE

- Source code

```
dataFolder = 'C:\Users\Toshiba\Documents\MATLAB\Data';
categories = {'Infected', 'Uninfected'};
imds = imageDatastore(fullfile(dataFolder, categories), ...
    'LabelSource', 'foldernames');

[trainingImages,validationImages] =
splitEachLabel(imds,0.8,'randomized');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
net = googlenet;
% Extract the layer graph from the trained network and plot the layer
graph.

lgraph = layerGraph(net);
figure('Units','normalized','Position',[0.1 0.1 0.8 0.8]);
plot(lgraph)

lgraph = removeLayers(lgraph, {'loss3-classifier','prob','output'});
numClasses=4;
% numClasses = numel(categories(trainingImages.Labels));
newLayers = [

fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor',20,'Bi
asLearnRateFactor', 20)
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classoutput')];
lgraph = addLayers(lgraph,newLayers);

lgraph = connectLayers(lgraph,'pool5-drop_7x7_s1','fc');

figure('Units','normalized','Position',[0.3 0.3 0.4 0.4]);
plot(lgraph)
ylim([0,10])

options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',3, ...
    'InitialLearnRate',1e-4, ...
    'ValidationFrequency',3, ...
    'ValidationPatience',Inf, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(trainingImages,lgraph,options);
[predictedLabels, probs] = classify(net,validationImages);
accuracy = mean(predictedLabels == validationImages.Labels)
```

```
idx = randperm(numel(validationImages.Files),4);
figure
for i = 1:4
    subplot(2,2,i)
    I = readimage(validationImages,idx(i));
    imshow(I)
    label = predictedLabels(idx(i));
    title(string(label) + ", " + num2str(100*max(probs(idx(i),:)),3) +
"%");
end
```

APPENDIX 2

ACTIVATIONS SOURCE CODE

- **Code**

```
load G1
im=imread('M1.jpeg');
act2 = activations(net,im,'conv2-norm2');
act2 = reshape(act2,size(act2,1),size(act2,2),1,size(act2,3));
act2_scaled = mat2gray(act2);
tmp = act2_scaled(:);
lim = stretchlim(tmp);
lim(1) = 0;
tmp = imadjust(tmp,lim);
act2_stretched = reshape(tmp,size(act2_scaled));
clf
montage(act2_stretched)
title('Activations from the conv2-relu_7x7 layer','Interpreter','none')
```