**Department of Electrical and Computer Engineering**

**North South University**



**Junior Design Project**

Malarial Cell Classification Using Deep Learning

|  |  |
| --- | --- |
| **Fahim Faisal Ratul** | **ID # 1712697642** |
| **Hasibul Hossain** | **ID # 1821534042** |
| **Reajus Salahin Shyon** | **ID # 1822187042** |

**Faculty Advisor**

**Dr. Mohammad Monirujjaman Khan**

**Professor**

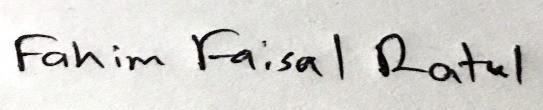
**ECE Department**

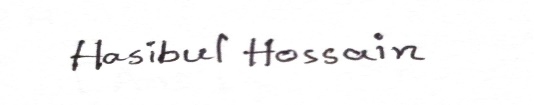
**Fall, 2021**

**DECLARATION**

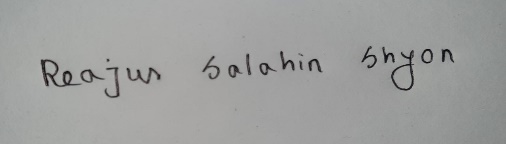
This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

**Students’ name & Signature**

1. **Fahim Faisal Ratul**
2. **Hasibul Hossain**



1. **Reajus Salahin Shyon**

****

**APPROVAL**

The capstone project entitled “**Malarial Cell Classification using deep learning**” by **Fahim Faisal Ratul (ID#1712697642), Hasibul Hossain (ID#1821534042),** and **Reajus Salahin Shyon (ID #1822187042)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on November and has been accepted as satisfactory.

|  |
| --- |
|  |
| **Supervisor’s Signature** |
|  |
| **Dr. Mohammad Monirujjaman Khan**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |
| **Department Chair’s Signature** |
| **Dr. Rezaul Bari**  **Associate Professor**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |

**ACKNOWLEDGMENT**

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The junior project program is very helpful to bridge the gap between the theoretical knowledge and real-life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjaman Khan for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 299 in which we could really work on this project and materialize it the way we have dreamt of.

We thank our friends and family for their moral support to carve out this project and always offer their support.

**ABSTRACT**

Malaria, which is caused by Plasmodium parasites, is a blood disease spread by the bite of a female Anopheles mosquito. There are almost 240 million instances in the United States. Each year, the illness affects roughly 40% of the population. A third of the world's population is in jeopardy. Macroscopic examinations generally take a long time. Examine thick and thin blood smears to diagnose a sickness or a condition's cause and discover what makes people vulnerable. Nonetheless, the accuracy of a smear depends on the quality of the smear and the knowledge of the situation. Parasite and non-parasite cells are classified and counted. The gold standard for diagnosis is manual assessment. It needs several actions to be completed. Furthermore, this procedure is used even when it comes to analysis, and it leads to erroneous and late conclusions. In the hands of professionals, our project's goal is to reduce human dependency, and the model is responsive. Malaria analysis is carried out automatically. A deep learning subcategory Convolutional Neural Networks (CNNs) are a kind of model that guarantees end-to-end results that are exceptionally adaptable and sophisticated in the extraction and classification of attributes. The accuracy and the techniques used have unwavering quality, speed, and affordability. Examinations are crucial to the disease's treatment and prevention, utter annihilation. We compared the overall results of this research. The performance of a CNN-based DL model that has been pre-trained as distinct extractors gets us closer to categorizing parasites and non-parasite cells to help in the detection of advanced illness. The best model layers for extracting attributes from the experimentally underlying records are determined. The information included in the dataset contains a wide range of parasitic and non-parasite blood image samples. To achieve a precise result, we've used Size, color, form, and cell size are all dominant traits. Count the number of photos to aid in the categorization process. CNN's that have been pre-trained are being utilized as a potential technique for the output of attribute extraction, which may be used to determine this. It has statistical support. As a result of these changes, automation in the pursuit of a cure, microscopy might be a huge help as a low-cost, simple, and reliable approach for identifying malaria. With the help of CNN, our model can recognize the image model and conclude whether it’s malarial affected or not.

**Table of Content**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | Page |
|  | **Chapter 1: Overview**…………………………………………………. | | 9 |
| 1.1 | Introduction………………………………………………………………………… | | 10 |
| 1.2 | Project Details……………………………………………………………………… | | 11 |
| 1.3 | Project Goals………………………………………………………………………. | | 13 |
| 1.4 | Summary…………………………………………………………………………… | | 14 |
|  |  |
|  |  |
|  | **Chapter 2: Methodology** ………………………………………….... | | 15 |
| 2.1 | Introduction………………………………………………………………………. | | 16 |
| 2.2 | Motivation towards our project…………………………………………………… | | 16 |
| 2.3 | Methodology……………………………………………………………………… | | 16 |
| 2.4 | Materials and tools………………………………………………………………… | | 16 |
| 2.5 | Block diagram………………………………………………….……….................. | | 17 |
| 2.6 | System architecture……………………………………………………………....... | | 20 |
|  |  | |  |
|  | **Chapter 3: Design results and analysis** | | 21 |
| 3.1 | Introduction……………………………………………………………… | | 22 |
| 3.2 | System related to our project………………………………………………. | | 22 |
| 3.3  3.4 | Problem with our current project……………………………………………  Proposed solution……………………………………………………….... | | 23  24 |
|  | **Chapter 4: Conclusion**……………………………………………… | | 28 |
|  | **Future work** | |  |
|  | **Bibliography**………………………………………………………… | | 31 |
|  | **Appendix: Software Listing**……………………………...………… | | 35 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure caption | Page No. |
|  |  |  |
| 1.1 | Malaria affected countries | 13 |
| 2.1 | Classifications with original data with CNN architecture | 18 |
| 2.2 | Classifications with data after filters with CNN architecture | 19 |
| 2.3 | Layers | 19 |
| 2.4 | System architecture | 20 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table No. | Table caption | Page No. |
| 1 | Model Comparison | 27 |

**CHAPTER 1**

**Introduction**

**1.1 Introduction**

Malaria is a potentially fatal disease caused by a parasite that infects a specific type of mosquito that feeds on humans. Malaria typically causes severe illness, including high fevers, shaking chills, and flu-like symptoms. Malaria is a potentially fatal disease, but illness and death can usually be avoided. Five species of Plasmodium (single-celled parasites) can infect humans and cause illness:

Plasmodium falciparum (or P. falciparum),

Plasmodium malariae (or P. malariae),

Plasmodium vivax (or P. vivax),

Plasmodium ovale (or P. ovale) and

Plasmodium knowlesi (or P. knowlesi).

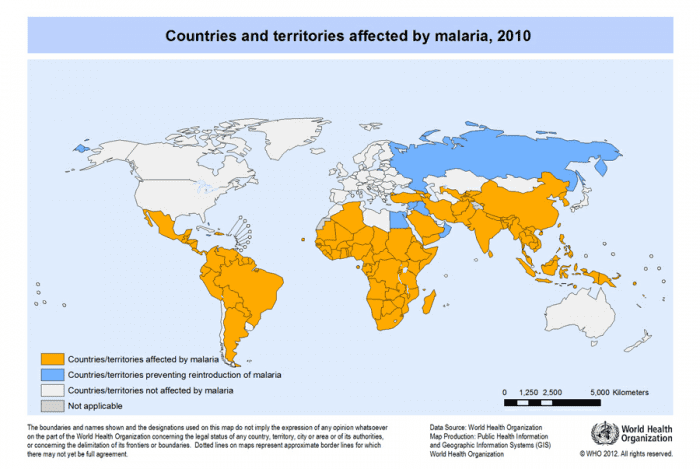
Malaria caused by the falciparum parasite may be fatal. Severe falciparum malaria patients may have liver and renal failure, seizures, and comas. Infections with P. vivax and P. ovale, although rarely severe, usually result in less serious sickness. However, the parasites may stay latent in the liver for months, causing symptoms to resurface months or even years later.

Malaria is an old illness; allusions to what was almost certainly malaria can be found in Chinese documents from about 2700 BC, Mesopotamian clay tablets from 2000 BC, Egyptian papyri from 1570 BC, and Hindu scriptures from the sixth century BC. Such historical documents must be treated with care, but as we get into the twentieth century, we are starting to tread on more solid footing. Early Greeks, such as Homer (about 850 BC), Empedocles of Agrigentum (around 550 BC), and Hippocrates (circa 400 BC), we're well aware of the distinctive ill health, malarial fevers, and enlarged spleens found in people living in swampy areas. For over 2500 years, malaria fevers were thought to be caused by miasmas rising from marshes, and it is usually assumed that the name malaria originates from the Italian mal'aria, which means spoiled air, but this has been questioned. The World Health Organization predicts that 229 million clinical cases of malaria occurred globally in 2019, with 409,000 people dying from the disease, the majority of whom were children in Africa. Malaria is a major economic burden on many countries because it causes so much sickness and death. Because many malaria-affected countries are already impoverished, the illness perpetuates a vicious cycle of sickness and poverty.

**1.2 Project Details**

Malaria is spread by mosquito bites from infected female Anopheles mosquitos. Only Anopheles mosquitoes may spread malaria, and they must have been infected with a blood meal from an infected human before. When a mosquito bites an infected individual, a minute quantity of blood is drawn, which includes minuscule malaria parasites. When the mosquito takes its next blood meal, about a week later, these parasites mix with the insect's saliva and are injected into the person who has been bitten. Because the malaria parasite is located in an infected person's red blood cells, malaria may also be transferred by blood transfusion, organ transplant, or the shared use of blood-contaminated needles or syringes. Malaria may also be passed from a woman to her unborn child before or after birth (this is known as "congenital" malaria).

Plasmodium falciparum is the most prevalent kind of malaria that causes severe and life-threatening illness; it is found in many African nations south of the Sahara Desert. People who are often bitten by P. falciparum-infected mosquitos are at the greatest risk of contracting malaria. Malaria is more likely to kill those who have little or no immunity to it, such as small children, pregnant women, and travelers from malaria-free areas. This illness is more likely to strike poor people in rural regions that lack access to health care. As a consequence of all of these circumstances, an estimated 90% of malaria fatalities occur in Africa south of the Sahara, with the majority of these deaths occurring in children under the age of five. Malaria may be treated with medication. You may also take medication to reduce your chances of contracting the condition.

Figure 1.1:Malaria affected countries

**1.3Project Goals**

If someone has the malaria virus within them, then they have to do a blood test in the hospital. To detect malaria, the doctor has to see a smear image of the blood to determine whether the blood is affected by malaria or not. If it is affected, then he is malaria positive, and if not, he is negative. But the maximum time between the images of a healthy blood smear image and an affected blood smear image is very much the same. It’s hard to identify the malaria virus from the image it produces in the naked eye. For that, the patient has to test his blood multiple times to see if he has malaria or go home knowing that he is not affected, but the reality is the opposite. Sometimes the malaria virus can be found in a very minor position. So, it’s not possible to detect many times. If it remains undetected, then it can spread and people can die.

**1.4 Summary**

With the help of deep learning, our model can detect the malaria virus from the blood smear image very precisely. Our model can analyze and identify which blood reports are affected and which are not. So, it will be very easy to detect, and many people’s lives can be saved by this deep learning model.

Our project is for the common people. This system can detect malaria in cell very accurately and quickly. So, for this many people’s life can be save. They can get proper treatment.

**CHAPTER**

**Methodology**

**2.1 Introduction**

In this chapter we discuss the motivation due to which we thought of implementing this system. We will also discuss in this chapter as to why we have chosen the medical field apart from all other fields to work it.

**2.2 Motivation towards our project**

With the help of deep learning, our model can detect the malaria virus from the blood smear image very fine. Our model can analyze and identify which blood report is affected and which is not. So, it will be very easy to detect and many people’s lives can be saved by this deep learning model.

**2.3 Methodology**

Data for this paper were obtained from the Kaggle dataset. Data classes are classified into two types. This information is used to diagnose malaria. The data in the first class is non-parasitic, while the data in the second class is parasitic. There are 3730 parasitic and 3000 healthy data points in the data set. The CNN methods, which have become increasingly popular in recent years, were used to classify malaria images. The application was created in MATLAB and used the models Alex Net, ResNet50, DenseNet201, Vgg19, Google Net, and Inceptionv3. The original data was first classified into six different architectures, after which the Gauss and Median filters were applied to the data set.

**2.4 Materials and Tools**

Anaconda and Jupyter notebook will be used in this project. Python are going to be our programing language of choice. Anaconda Navigator can look up packages on Anaconda.org or in a local Anaconda Repository. It is compatible with Windows, macOS, and Linux. The Jupyter Notebook may be a free and open-source web application that allows you to create and share documents with live code, equations, visualizations, and narrative text. Data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and lots of other applications are possible. Data for this paper was obtained from the Kaggle dataset. Data classes are classified into two types. This information is used to diagnose malaria. The data in the first class is non-parasitic, while the data in the second class is parasitic. The techniques for data collection and preprocessing are covered in the subsections that follow. The data used to support the research outcome are freely available at

<https://www.kaggle.com/itsdaniyal/malerial-cell-classification-dataset>

**2.5 Block Diagram**

Deep learning is the ability of computers to process and learn from data. The main difference between deep learning models and traditional neural networks is that deep learning models have multiple layers. In 2012, deep learning was put on hold. Deep learning's popularity skyrocketed after the Deep Learning model won the ImageNet competition in 2012. One of the reasons that deep learning has recently gained popularity is the development of cards with faster processing speeds. As data volumes increased, so did the proclivity for deep learning.

Fully Connected

AlexNet

Vgg19

GoogleNet

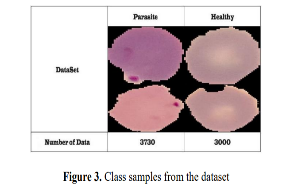
ResNet50

SoftMax

DenseNet201

Classification

Parasite



+

+

Dataset

Healthy

Inceptionv3

Fig 2.1: Classifications with original data with CNN architecture

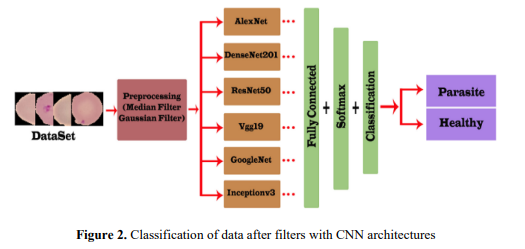


Fig 2.2: Classifications with data after filters with CNN architecture

Color

Constancy

Resample to

50 x 50

Input

Image

Classification Layer

SoftMax Layer

Fully Connected Layer

Convolutional Layers 1-6

Parasitized

Uninfected

Fig 2.3: Layers

**2.6 System Architecture**

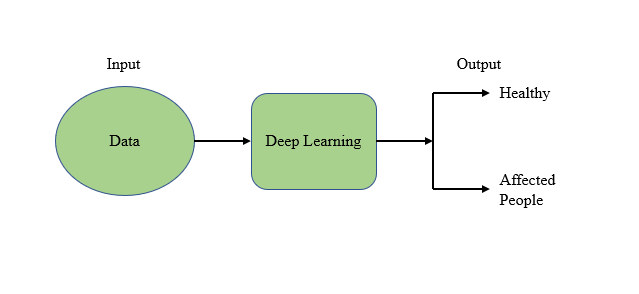
****

Fig 2.4: System architecture

**2.7 System Architecture**

The convolution layer is the primary building block of a CNN. It is in charge of the vast majority of the network's computational load. This layer performs a dot product on two matrices, one of which is a set of learnable parameters known as a kernel and the other being the restricted section of the receptive field. The kernel is smaller than a picture but has more depth. This means that if the image has three (RGB) channels, the kernel height and width will be small, but the depth will be large.

If we have an input of size W x W x D and a given number of kernels with a spatial size of F, stride S, and amount of padding P, we can calculate the size of the output volume using the following formula:

= (1)

**CHAPTER 3**

**Design results and analysis**

-

**3.1 Introduction**

In this chapter we discuss the types of medical systems that currently exist in the market. We also focus on the loopholes that the current system entails and a proper justification will be provided as to why our system is the ideal one in the current circumstances.

**3.2 Systems related to our project**

Using computing algorithms for cost-effective solutions to support interoperable healthcare [1] in lowering diseases has been a major focus of research in recent decades. Neto et al. [2], for example, suggested a simulator for replicating epidemiological events in real-time. For malaria identification and classification, Kaewkamnerd et al. [3] presented a five-phase image analysis approach. Anggraini et al. [4] created a program that uses image segmentation techniques to separate blood cells from their surroundings. In addition, Rajaraman et al. [5] created feature extractors for uninfected and parasitized blood cell categorization using pre-trained CNN-based deep learning models to aid illness detection. Using the underlying data, the researchers employed an experimental technique to find the best model layers. Two fully connected dense layers and three convolutional layers make up the CNN model. The performance is tested by extracting features from uninfected and parasitized blood cells using VGG-16, AlexNet, Xception, DenseNet-121, and ResNet-50. [6] and Liang et al. [7] both offer exclusively CNN-based malaria classifiers, in contrast to [8].

MOMALA [9] is a smartphone and microscope-based application designed to swiftly and inexpensively detect malaria. On a standard blood-smeared slide, the MOMALA app can detect the presence of malaria parasites. The blood smear is photographed using a phone camera mounted to the microscope's ocular, which is subsequently analyzed. Currently, the application is heavily reliant on microscopes that are large, bulky, and difficult to carry.

[10] developed a mobile app that captures photographs of blood samples and detects malaria almost instantly. We can evaluate blood samples without consulting microscope technicians by using a smartphone app. The program works by clamping a smartphone to the eyepiece of a microscope, then analyzing blood sample photographs and painting a red circle around malaria parasites A lab worker examines the case afterward. The extraction of meaningful features is critical to the efficiency of any machine learning approach. The bulk of computer-assisted diagnostic systems that use machine learning models for image analysis generate choices based on manually developed characteristics [11]– [12]. The approach also needs computer vision ability to assess the variation in picture size, color, background, angle, and location of interest. Deep learning algorithms may be effectively employed to overcome the limitations of a hand-engineered feature extraction technique [13]. To uncover hierarchical feature relations in raw picture data, deep learning models use a sequence of successive layers with hidden nonlinear processing units. Nonlinear decision-making, learning difficulty, feature extraction, and classification are all aided by low-level features abstracted from higher-level traits [14]. Furthermore, when dealing with large volumes of data and processing resources, deep learning models outperform kernel-based approaches like Support Vector Machines (SVMs), making them extremely scalable [15].

**3.3 Problems with the current systems**

The problems associated with manual diagnosis argue in favor of automating the malaria diagnosis process. The automation of the diagnosis process will ensure accurate disease diagnosis and, as a result, holds the promise of delivering dependable health-care to resource-limited areas. As a result, rural areas that lack specialized infrastructure and trained personnel can greatly benefit from automated diagnosis. Automating malaria diagnosis entails adapting conventional microscopy methods, expertise, practices, and knowledge to a computerized system structure. Malaria detection at an early stage is critical for ensuring proper diagnosis and increasing the chances of cure.

Due to the severity and number of fatalities claimed by this disease, it is reasonable to accept the possibility of minor implementation errors introduced by an automated system. An automated system consists of streamlined image processing techniques for initial filtering and segmentation, as well as a pattern recognition suite. Previous research has found that the degree of agreement between clinicians on the severity of the disease in a given patent's sample is very low. As a result, a computer-assisted system as a decision support system can be critical to a faster and more reliable diagnosis. It can help provide a benchmark and a standardized method of measuring the disease's level of infection.

There isn't a large enough, high-quality image dataset of pathologically annotated cell images to fully train multiple-layer neural networks. As a result, we collaborated with a pathology team to create a dataset. After preprocessing the data, we randomly selected a large number of cell images and sent them to pathologists at the University of Alabama at Birmingham. The entire dataset of slide images has been evenly divided into four segments. Each pathologist is assigned two segments, ensuring that each cell image is viewed and labeled by at least two experienced pathologists. One cell image can only be considered infected and included in our final dataset if all of the reviewers mark it positively; otherwise, it is excluded.

**3.4 Proposed Solution**

To solve that problem, we want to implement deep learning. Our model based on deep learning can solve this problem accurately. As anyone who has witnessed firsthand knows, healthcare delivery in low-resource settings is fundamentally different from more affluent settings. Artificial Intelligence, including Machine Learning and more specifically Deep Learning, has made amazing advances over the past decade. Significant resources are now dedicated to problems in the field of medicine, but with the potential to further the digital divide by neglecting underserved areas and their specific context. In the general case, Deep Learning remains a complex technology requiring deep technical expertise. Deep learning, also known as hierarchical learning or deep structured learning, is a type of machine learning that uses a layered algorithmic architecture to analyze data. In deep learning models, data is filtered through a cascade of multiple layers, with each successive layer using the output from the previous one to inform its results. Deep learning models can become more and more accurate as they process more data, essentially learning from previous results to refine their ability to make correlations and connections. Deep learning is loosely based on the way biological neurons connect with one another to process information in the brains of animals. Similar to the way electrical signals travel across the cells of living creates, each subsequent layer of nodes is activated when it receives stimuli from its neighboring neurons. In artificial neural networks (ANNs), the basis for deep learning models, each layer may be assigned a specific portion of a transformation task, and data might traverse the layers multiple times to refine and optimize the ultimate output. These “hidden” layers serve to perform the mathematical translation tasks that turn raw input into meaningful output.

**3.5 Result and Analysis**

Our model provided 96 percent accuracy and 97 percent validation accuracy in the 10th epoch after training with the train generator, validation generator, step per epoch=8, and 10 epochs. The training accuracy was quite low in the first few epochs, starting at 91% and increasing to 96 percent after the tenth epoch. The validation accuracy began at 95% and ended at 0.9716 after the tenth epoch. VGG19 has a train accuracy of 96 percent and a validation accuracy of 97 percent, with a train loss of 10% and a validation loss of 9%.

**3.5.1 Model Accuracy**

The accuracy history plot shows that the train's accuracy increased rapidly after each epoch. The accuracy was 91 percent in the first epoch and increased with each epoch. The model's validation accuracy was 95 percent and increased until the last epoch. The model accuracy plot shows that an increasing line has been drawn for training accuracy and a line that is around 95 percent –97 percent accuracy all the time during the epoch for test accuracy. The model accuracy and model loss are depicted in Figs. 3 (a), 3 (b) AND 3 (c) respectively.

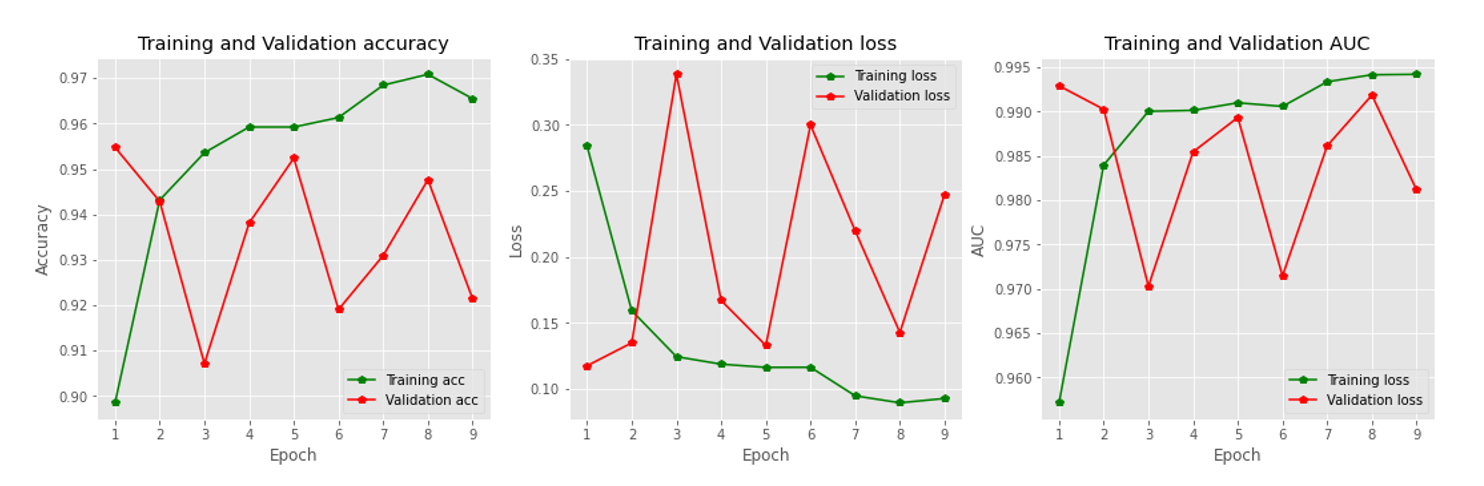


Figure 3: (a) Model accuracy, (b) model loss (c) Model AUC

It can be assumed from the model loss plot that both the lines of training loss and test loss have gradually decreased. The train loss was 20 percent after the first epoch and 10 percent after ten epochs. After first epoch, the validation loss was 12%, and after 10 epochs, it was 8%. Figure 5 depicts a plot of the model loss.

**3.5.2 Model Comparison**

In this study, the DenseNet201, ResNet50, AlexNet, Vgg19, GoogleNet, and Inceptionv3 pretrained models were compared to various previous models. In this study, InceptionV3, DenseNet201, AlexNet, GoogleNet and VGG19 produced better results in accuracy and efficiency than the models used in previous studies. The accuracy of the pretrained models improved noticeably. Table 1 shows a comparison of various models and data sets.

**Table 1:** Model Comparison

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | | | | | This paper (model name) | Accuracy (%) | Reference paper (model name) | Accuracy (%) | |  | | | | | VGG-19 | 97.16 | Ref [19] (VGG-19) | 85.6 | | Inceptionv3 | 96.92 | (Inception V3) | 93.3 | | ResNet50 | 91.95 | (ResNet50) | 92.8 | | GoogleNet | 95.00 | (GoogleNet) | 92.5 | | AlexNet | 90.45 | (AlexNet) | 87.3 | | DenseNet201 | 96.11 | (DenseNet201) | 94.3 | |  | | | | |  | | | | |

Except for ResNet50, all of the models in Table 1 are extremely accurate. Out of all the pretrained models, DenseNet201, ResNet50, AlexNet, Vgg19, GoogleNet, and Inceptionv3 have consistently produced smooth results from the start and have nearly the highest accuracy of all the other research.

**CHAPTER 14**

**CONCLUSION**

Conclusion:

Malaria is a global disease that has claimed the lives of millions of people. We briefly described the workflow for classifying red blood cell images and went over the data augmentation methods we proposed to address the problem of training deep convolutional neural networks with a small dataset. Our model is based on the VGG. VGG19 is overly complex for the task; a simplified version works better in this application and avoids overfitting. The classification accuracy associated with training, validating, and testing with various combinations of the original dataset and significantly augmented datasets were then compared. Our observations show that deep learning can be used effectively to detect malaria parasites on thin blood smears. The proposed model outperforms or is comparable to previous research. We demonstrated that it is not necessary to use very deep neural networks to achieve high accuracy and that shallower versions may be preferable. We relabeled incorrectly classified images in the dataset. More work is needed to create a deep learning model capable of distinguishing between true parasites, impurities, and artifacts in the same way that a human expert would. However, the presented model correctly identifies the majority of cases. A well-executed system would require several factors to communicate with one another. This entails the characteristics of the microscope, the type of staining used, the slide preparation mechanism, as well as image exploration and machine learning software. The use of images with varying characteristics in each piece of research increases the difficulty in determining which method is best to use. This implies that the method proposed in this field of study is highly dependent on the image's characteristics. Because deep learning has gained widespread acceptance, a gigantic amount of support has taken the lead in data acquisition efforts. In addition, annotated data image repositories for preparation are now widely understood.

Future work:

We believe that the findings of this study will help to develop valuable mobile-based solutions to address treatment reliability and a lack of medical expertise. As an immediate extension of this work, we will consider using image augmentation on the training data to alleviate overfitting further, as well as testing different adaptive variants of the SGD optimizer to see how they affect performance results. In the future, we hope to improve predictions by employing ensemble methods such as model stacking.

**BIBLIOGRAPHY**

[1] M. Masud, M. S. Hossain, and A. Alamri, “Data interoperability and multimedia content management in e-health systems,” IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 6, pp. 1015–1023, 2012.

[2] O. B. Leal Neto, C. M. Albuquerque, J. O. Albuquerque, and C. S. Barbosa, “The schisto track: a system for gathering and monitoring epidemiological surveys by connecting geographical information systems in real time,” JMIR Mhealth Uhealth, vol. 2, no. 1, article e10, 2014.

[3] S. Kaewkamnerd, C. Uthaipibull, A. Intarapanich, M. Pannarut, S. Chaotheing, and S. Tongsima, “An automatic device for detection and classification of malaria parasite species in thick blood film,” BMC Bioinformatics, vol. 13, Supplement 17, p. S18, 2012.

[4] D. Anggraini, A. S. Nugroho, C. Pratama, I. E. Rozi, A. A. Iskandar, and R. N. Hartono, “Automated status identification of microscopic images obtained from malaria thin blood smears,” in Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, July 2011.

[5] S. Rajaraman, S. K. Antani, M. Poostchi et al., “Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images,” PeerJ, vol. 6, article e4568, 2018.

[6] G. P. Gopakumar, M. Swetha, G. S. Siva, and G. 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, vol. 11, no. 3, 2018.

[7] Z. Liang, A. Powell, I. Ersoy et al., “CNN-based image analysis for malaria diagnosis,” in 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 493–496, Shenzhen, China, December 2016.

[8] S. Rajaraman, S. K. Antani, M. Poostchi et al., “Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images,” PeerJ, vol. 6, article e4568, 2018.

[9] “MOMALA,” <https://momala.org/malaria-diagnosis/>.

[10] “This New App Helps Doctors Diagnose Malaria in Just 2 Minutes,” <https://www.globalcitizen.org/en/content/app-diagnose-malaria-uganda>.

[11] N. E. Ross, C. J. Pritchard, D. M. Rubin, and A. G. Dusé, “Automated image processing method for the diagnosis and classification of malaria on thin blood smears,” Medical & Biological Engineering & Computing, vol. 44, no. 5, pp. 427–436, 2006.

[12] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. R. Thoma, “Image analysis and machine learning for detecting malaria,” Translational Research, vol. 194, pp. 36–55, 2018

[13] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, “Automatic fruit classification using deep learning for industrial applications,” IEEE Transactions on Industrial Informatics, vol. 15, no. 2, pp. 1027–1034, 2019.

[14] M. Usama, B. Ahmad, J. Wan, M. S. Hossain, M. F. Alhamid, and M. A. Hossain, “Deep feature learning for disease risk assessment based on convolutional neural network with intra-layer recurrent connection by using hospital big data,” IEEE Access, vol. 6, pp. 67927–67939, 2018.

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

[16] Y. Zhang, Y. Qian, D. Wu, M. Shamim Hossain, A. Ghoneim, and M. Chen, “Emotion-aware multimedia Systems security,” IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 617–624, 2019.

[21] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, “Edge intelligence in the cognitive internet of things: improving sensitivity and interactivity,” IEEE Network, vol. 33, no. 3, pp. 58–64, 2019.

[22] W. David Pan, Yuhang Dong and Dongsheng Wu Submitted: June 13th 2017Reviewed: November 13th 2017Published: September 19th 2018 DOI: 10.5772/intechopen.72426

[23] Kaustubh Chakradeo, Michael Delves, Sofya Titarenko, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:15, No:2, 2021

DOI:10.5281/zenodo.4569849 Conference: International Conference on Machine Learning Methods for Computer Vision At: New York

**APPENDIX**

**SOFTWARE LISTING**

!unzip '/content/drive/MyDrive/Training/299 Malaria/malaria.zip'

import os

len(os.listdir('/content/malaria/Parasitized')), \

len(os.listdir('/content/malaria/Uninfected'))

import os

os.makedirs('/content/drive/MyDrive/Training/299 Malaria/Dataset')

!cp -r '/content/malaria/Parasitized' -d '/content/drive/MyDrive/Training/299 Malaria/Dataset'

!cp -r '/content/malaria/Uninfected' -d '/content/drive/MyDrive/Training/299 Malaria/Dataset'

!pip install split-folders

import splitfolders

splitfolders.ratio('/content/drive/MyDrive/Training/299 Malaria/Dataset', output="splitted\_data", seed=1337, ratio=(.7, 0.2,0.1))

!zip -r '/content/splitted\_data.zip' '/content/splitted\_data'

!cp '/content/splitted\_data.zip' -d '/content/drive/MyDrive/Training/299 Malaria'

!cp '/content/drive/MyDrive/Training/299 Malaria/splitted\_data.zip' -d '/content'

!unzip '/content/splitted\_data.zip'

!pip install scikit-plot

# import all libraries

# Train/Test Libraries

import os

import numpy as np

import tensorflow as tf

from collections import Counter

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix

import scikitplot

from sklearn.metrics import roc\_curve, auc

from sklearn.metrics import classification\_report

data\_aug\_train = tf.keras.preprocessing.image.ImageDataGenerator(

height\_shift\_range = 0.15,

width\_shift\_range = 0.15,

rotation\_range = 10,

shear\_range = 0.1,

fill\_mode = 'nearest',

zoom\_range = 0.2)

train\_generator = data\_aug\_train.flow\_from\_directory(

'/content/splitted\_data/train',

target\_size=(224, 224),

batch\_size = 32,

class\_mode ='categorical',

color\_mode = 'rgb',

classes = ['Parasitized', 'Uninfected'],

seed = 2,

shuffle = True,

interpolation = 'lanczos'

)

data\_aug\_val = tf.keras.preprocessing.image.ImageDataGenerator()

val\_generator = data\_aug\_val.flow\_from\_directory(

'/content/splitted\_data/val',

target\_size =(224, 224),

batch\_size =32,

class\_mode ='categorical',

color\_mode = 'rgb',

classes= ['Parasitized', 'Uninfected'],

seed = 2,

shuffle = True,

interpolation = 'lanczos'

)

data\_aug\_test = tf.keras.preprocessing.image.ImageDataGenerator()

test\_generator = data\_aug\_test.flow\_from\_directory(

'/content/splitted\_data/test',

target\_size = (224, 224),

batch\_size = 2756,

class\_mode ='categorical',

color\_mode = 'rgb',

classes = ['Parasitized', 'Uninfected'],

shuffle = False,

interpolation = 'lanczos'

)

train\_generator.class\_indices, val\_generator.class\_indices

update 2:

import tensorflow as tf

import numpy as np

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D

from tensorflow.keras.layers import TimeDistributed

from tensorflow.keras.layers import Dense, Flatten, Dropout,BatchNormalization,Activation

from tensorflow.keras.regularizers import l2

from keras.callbacks import ModelCheckpoint

import glob,os

data\_aug\_train = tf.keras.preprocessing.image.ImageDataGenerator(

rescale=1./255,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

horizontal\_flip=True,

# rotation\_range=15,

# shear\_range=0.2,

# zoom\_range=0.25,

)

data\_aug\_valid = tf.keras.preprocessing.image.ImageDataGenerator(

rescale=1./255)

train=data\_aug\_train.flow\_from\_directory(directory='/content/splitted\_data/train',

target\_size=(128,128),

color\_mode='rgb',

classes=['Parasitized','Uninfected'],

shuffle=True,

class\_mode='binary',

batch\_size=64)

valid=data\_aug\_valid.flow\_from\_directory(directory='/content/splitted\_data/val',

target\_size=(128,128),

color\_mode='rgb',

classes=['Parasitized','Uninfected'],

shuffle=False,

class\_mode='binary',

batch\_size=64)

from tensorflow.keras.applications.vgg19 import VGG19

p\_model = VGG19(

include\_top=False,

weights='imagenet',

input\_shape=(128, 128, 3)

)

p\_model.trainable=True

p\_model.summary()

model = Sequential()

model.add(BatchNormalization(input\_shape=(128, 128, 3)))

model.add(p\_model)

model.add(Flatten())

# finalize with standard Dense, Dropout...

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.3))

model.add(Dense(1, activation='sigmoid'))

model.summary()

from tensorflow.keras.optimizers import Adam,RMSprop

model.compile(loss='binary\_crossentropy',

optimizer=Adam(lr=0.0001),

metrics=['accuracy'])

modelPath = '/content/drive/MyDrive/Training/299 Malaria/saved\_models\_VGG\_19/Pretrained VGG19'

if not os.path.exists(modelPath):

os.makedirs(modelPath)

print('Model Directory Created')

else:

print('Model Directory Already Exists')

checkpoint = ModelCheckpoint(modelPath, monitor='val\_accuracy', verbose=1, save\_best\_only=True)

history = model.fit(

train,

validation\_data = valid,

validation\_steps = len(valid),

shuffle = True,

steps\_per\_epoch = len(train),

epochs = 10,

callbacks=[checkpoint],

verbose = 1)

import pickle

with open('/content/drive/MyDrive/Training/299 Malaria/saved\_models\_VGG\_19/Pretrained VGG19/saved\_model.pb', 'wb') as file\_pi:

pickle.dump(history.history, file\_pi)

import matplotlib.pyplot as plt

plt.style.use('ggplot')

def plot\_history(history):

acc = history['accuracy']

val\_acc = history['val\_accuracy']

loss = history['loss']

val\_loss = history['val\_loss']

#auc = history['auc']

#val\_auc = history['accuracy']

x = range(1, len(acc) + 1)

plt.figure(figsize=(18, 5))

plt.subplot(1, 2, 1)

plt.plot(x, acc, 'b', label='Training acc',marker = 'p',color='green')

plt.plot(x,val\_acc, 'r', label='Validation acc',marker = 'p',color='red')

plt.title('Training and Validation accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.legend(loc='lower right' )

plt.subplot(1, 2, 2)

plt.plot(x, loss, label='Training loss',marker = 'p',color='green')

plt.plot(x, val\_loss, label='Validation loss',marker = 'p',color='red')

plt.title('Training and Validation loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

# plt.savefig('curve.jpg',dpi=600)

#plt.subplot(1, 3, 3)

#plt.plot(x, auc, 'b', label='Training loss',marker = 'p',color='green')

#plt.plot(x, val\_auc, 'r', label='Validation loss',marker = 'p',color='red')

#plt.title('Training and Validation AUC')

#plt.xlabel('Epoch')

#plt.ylabel('AUC')

#plt.legend()

#plt.savefig('curve.jpg',dpi=600)

# plot\_history(modelHistory.history)

import pickle

with open('/content/drive/MyDrive/Training/299 Malaria/saved\_models\_VGG\_19/Pretrained VGG19/saved\_model.pb', 'rb') as file\_pi:

dct=pickle.load(file\_pi)

plot\_history(dct)