MALARIA PARASITE DETECTION FROM HUMAN BLOOD SMEAR IMAGES USING DEEP LEARNING TECHNIQUES

BY

TEOW YI JIA

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Malaria Parasite Detection from human blood smear images using deep learning techniques

Academic Session: JUNE 2023

_____TEOW YI JIA_____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:
C-10 Jermal Court,
5A Jalan Bagan Jermal
10250 Penang

Sayed Ahmad Zikri Bin Sayed Aluwee

Supervisor's name

Date: _____13/9/2023_____ **Date**: __14/9/2023_____

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 13/9/2023

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ***Teow Yi Jia*** (ID No: ***2000323***) has completed this final year project entitled "*Malaria parasite detection from human blood smear image using deep learning technique*" under the supervision of *Dr Sayed Ahmad Zikri Bin Sayed Aluwee* (Supervisor) from the Department of *Computer Science*, Faculty of *Information and Communication Technology*, and *Dr Goh Chuan Meng* (Co-Supervisor)* from the Department of *Computer Science*, Faculty of *Information and Communication Technology*.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*Teow Yi Jia*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**MALARIA PARASITE DETECTION FROM HUMAN BLOOD SMEAR IMAGES USING DEEP LEARNING TECHNIQUES** " is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___ _____

Name : ____Teow Yi Jia_____

Date : ____13/9/2023_____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors Dr Sayed Ahmad Zikri Bin Sayed Aluwee who has given me this bright opportunity to work on this project and spent much of his valuable time supporting and guiding me.

I must also thank my parents, who have provided love, support, and encouragement throughout my university life. I want to thank my friends as well for providing suggestions and encouragement that make me feel possible to finish the whole project on time.

ABSTRACT

Malaria is a deadly disease caused by a parasite that is transmitted to humans through the bite of an infected mosquito. The standard method of diagnosing malaria involves a graphic examination of human blood smears under a microscope by medical experts to determine parasite-infected red blood cells. However, this method is ineffective, and the diagnosis is dependent on the knowledge and experience of the examiner which is still lack in some places especially in rural area. Faultless identification of medical imaging has become a crucial factor in medical diagnosis and decision-making with the significant development in deep learning research. Even though malaria can be fatal, most cases of illness and fatalities are frequently preventable if there is an accurate detection. Therefore, automated parasite detection technologies are highly needed to decrease the rate of false detection.

The aim of this study is to investigate various deep learning methods that can be employed to identify the presence of the malaria parasite in human blood cells. Additionally, the objective is to develop a convolutional neural network (CNN) based on deep learning techniques to detect malaria in medical cell images through image classification. This paper covers various aspects of malaria detection, including image pre-processing, feature extraction, and classification.

Finally, the study discusses the potential for future research in deep learning-based malaria detection, including the use of transfer learning, ensemble models, and other deep learning techniques. Overall, the study highlights the promising results of deep learning-based malaria detection and its potential to revolutionize malaria diagnosis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DL	Deep Learning
RBC	Red Blood Cell
CNN	Convolutional Neural Network
RDT	Rapid Diagnostic Test
MP	Malaria Parasite
ReLU	Rectified Linear Unit
VGG	Visual Geometry Group
LSH	Locality Sensitive Hashing
RGB	Red, Green Blue
BN	Batch Normalization

Chapter 1

Introduction

Female Anopheles mosquitoes, which have been infected with parasites from feeding on an infected person's blood, can transmit the potentially fatal disease malaria to humans through their bites. Malaria is a severe illness that poses a significant risk to human life [1]. These mosquitoes are the only ones that can spread malaria, and they become infected with the parasite by feeding on the blood of an infected person. Once within the mosquito, the parasites grow and multiply before eventually travelling to the salivary glands and becoming prepared to infect the next victim of the mosquito's bite.

Once an individual is bitten by an infected mosquito, the malaria parasites invade the bloodstream and initiate an attack on the red blood cells responsible for transporting oxygen. The earliest symptoms of malaria may be modest and challenging to identify as malaria; they often develop a few days or weeks following the mosquito bite. However, the illness can result in significant complications and even death if left untreated. This is because these deadly parasites can survive in a person's body for more than a year without presenting symptoms.

In rural communities with insufficient access to medical care and diagnostic equipment, malaria is more common. The World Health Organization (WHO) estimates that there were approximately 241 million cases of malaria worldwide in 2020, with the African region accounting for the highest number of cases. The South-East Asia region was the second highest contributor to the disease, with almost 5 million cases and 9000 fatalities in 2020.[2]

Prevention and treatment of malaria are key to reducing its impact. Preventive measures include using insecticide-treated bed nets and mosquito repellent, as well as eliminating standing water where mosquitoes breed. Prompt diagnosis and treatment of malaria with effective antimalarial drugs can cure the disease and prevent further spread. However, access to these treatments can be a challenge in some areas, and new methods for preventing and treating malaria are continually being researched.

1.1 Problem Statement and Motivation

Time-consuming and inaccurate traditional techniques

The present diagnostic procedures for malaria, such as microscopic analysis of blood smears, are time-consuming, labour-intensive, and error prone.[3] This can result in delayed treatment and increased risk for patients. Therefore, a more precise and efficient technique of diagnosis is essential for the disease's appropriate management and therapy.

Existing malaria detection systems based on deep learning techniques often require large amounts of labelled data to achieve high accuracy. However, obtaining such data can be challenging, particularly in resource-limited settings where the disease is endemic. As a result, stronger and more effective deep learning models are required, ones that can produce high accuracy results with just a little quantity of labelled data.

Inconsistent of staining methods

Cell morphology, imaging conditions, and differences in staining methods can all have an impact on how well malaria detection systems operate. Therefore, there is a need for standardization of sample preparation and imaging protocols to ensure consistent and reliable results across different settings and laboratories.

Highly dependent on the expertise and experience of the microscopist

Malaria is a prevalent illness, primarily in countries with low and middle income, where there are restricted healthcare resources. There is a need for a low-cost and portable diagnostic tool that can accurately detect malaria in resource-limited settings.

Motivation

This project aims to introduce an innovative and accurate method for screening and diagnosing malaria parasites using deep learning techniques that are straightforward and effective. The motivation of this project is to help medical experts accurately and rapidly diagnose patients who have been infected by Malaria Parasite. To do so, the expected outcome in this project is to develop an automating Malaria Parasite detection model through classification of human blood smears images.

1.2 Objectives

The primary goal of research on malaria detection is to create methods that are precise, effective, and dependable for promptly identifying and diagnosing malaria. The goal is to improve patient outcomes by enabling prompt treatment, reducing morbidity and mortality rates, and preventing the spread of the disease.

Deep learning-based approaches have emerged as a promising solution for malaria detection due to their ability to automatically learn and extract meaningful features from large datasets of infected and uninfected blood smear images.

The specific research objectives of malaria detection using deep learning may include:

- Develop a robust deep learning models for accurate classification of infected and uninfected blood smear images.
- To decrease the burden and the amount of time required for detecting an enormous medical image dataset.

Investigating the feasibility of deploying deep learning models in resource-limited settings where malaria is prevalent.

1.3 Project Scope and Direction

The system proposed will be developed by using Python programming language and PyTorch library on Google Collaboratory. In this project, deep learning techniques will be used for the classification of healthy or infected blood smear images. The scope of this project is to develop, train and evaluate DL model architecture to provide a high accuracy and performance of the MP detection. In conclusion, the primary focus of this study is the automatic detection of MP using medical image.

1.4 Contributions

The following stated is a summary of contributions of the research that is being proposed:

- Demonstrates the potential of deep learning techniques, particularly transfer learning for automatically detecting malaria parasites in images of human blood smears.

- Contributes to the development of a system that can diagnose malaria automatically., which can enhance the speed and accuracy of the diagnostic process.
- Can serve as a useful tool for healthcare providers in diagnosing malaria and can also be adapted for use in telemedicine and mobile health applications.
- Has the potential to enhance the diagnosis and treatment of malaria, which remains a significant public health issue in various regions of the world.

1.5 Report Organization

The background information, problem statement regarding Malaria Parasite is provided in chapter 1. Additionally, the problem statement is used to discuss the project's aim and scope. The project scope, contributions also stated in this chapter.

Chapter 2 discusses the review of a few research publications, analysts' studies of the current methodologies, and comparisons of their effectiveness, strengths, flaws, accuracy, and others.

In Chapter 3, it was discussed the block diagram, how to pre-process datasets and models, as well as what kind of datasets were prepared. Moreover, the architecture of the model also discussed.

Chapter 4 discussed tools used in this project and the implementation issues and challenges.

In chapter 5, the system evaluation is discussed. Finally, chapter 6 concludes the entire project.

Chapter 2

Literature Review

2.1 Previous Works on malaria parasite detection using deep learning techniques.

2.1.1 Malaria Parasite Detection using Residual Attention U-Net[4]

The study proposed the use of a residual attention U-Net for biomedical image segmentation, which was customized to work with fewer training images and produce more precise segmentations. The U-Net design comprises two primary elements, namely the contraction path (encoder), which captures the context of the image, and the expansion path (decoder), which utilizes transposed convolutions to enable precise localization. The researchers utilized a dataset of blood smear microscopic images that had been annotated by professional microscopists. They chose U-Net because it provides a simple convolutional network design that integrates a portion of the encoder feature with the decoder, which is not typical of traditional CNNs. Moreover, the output of the U-Net model is then passed to the following phase to aid the decoder in recovering any lost features during maxpooling.

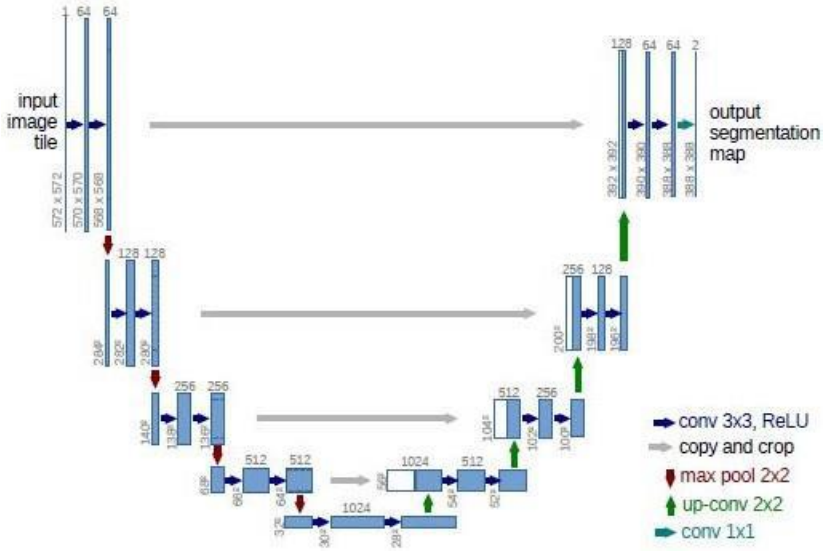


Figure 2.1.1.1 U-net computing paradigm model [4]

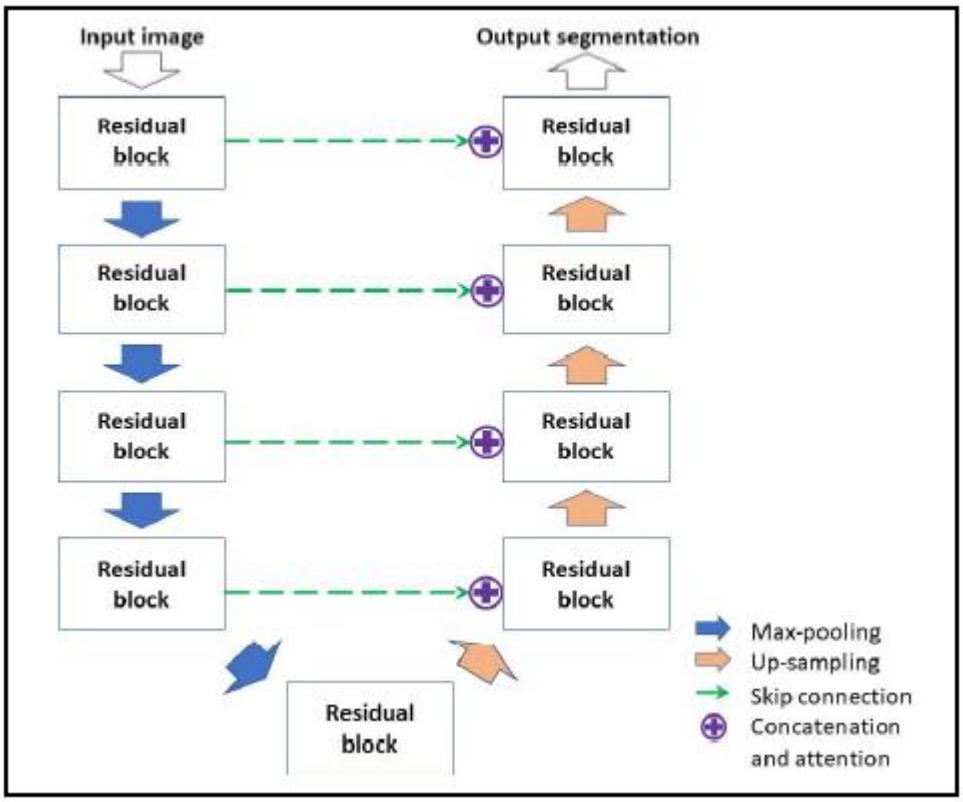


Figure 2.1.1.2 Proposed residual attention U-net model [4]

As stated earlier, the U-net model only needs four convolution and max-pooling layers for feature extraction. Next, the model also employs soft attention and Locality Sensitive Hashing (LSH)-based attention to reduce memory inefficiency. Data augmentation techniques were also used to enhance the model's performance while maintaining the semantics of each image. The results show that the Residual Attention U-Net model achieves high accuracy and outperforms existing methods, the model achieved a training accuracy of 96.87% and a validation accuracy of 96.91%. Authors suggest that their approach could be extended to other medical image analysis tasks, indicating its potential for medical image analysis tasks.

2.1.1.2 Strengths and Weaknesses

The strength of this paper is that the performance of the model is thoroughly evaluated in this paper, including its sensitivity, specificity, and F1 score. The study was conducted on a dataset of microscopic images of blood smears, which were annotated by expert microscopists. The model demonstrated its potential for use in clinical settings by achieving high accuracy and outperforming current methods.

However, because the Residual Attention U-Net model is only tested on one dataset in this research, it may not apply well to other datasets.

2.1.2 Malaria Detection using Deep learning [5]

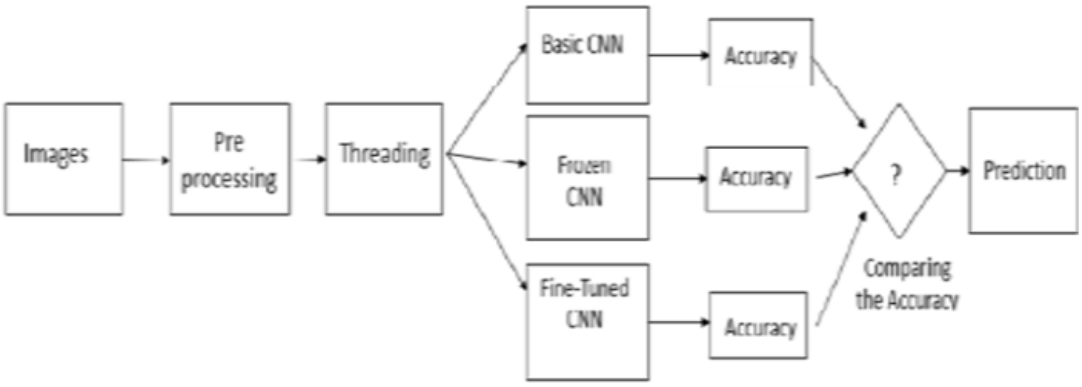


Figure 2.1.2.1 Block Diagram of Proposed Work [5]

In their paper titled "Malaria Detection with Deep Learning," Shekar et al. (2020) present a method based on deep learning for identifying malaria using microscopic images of blood samples. The authors collected a total of 27,558 images from the internet and used them to train their model. The images were split into train and test sets using Sklearn, and OpenCV was used to extract the parameters of the images. During pre-processing, contour detection was used on the cells to identify any dark or black patches within them, which were then circled using a curve drawn close to the spot.

After contouring the image, threading occurs with the help of the Thread Pool Executor attribute. For two-dimensional images, Conv2D is used in a basic CNN. Both the basic CNN and the frozen CNN are arranged in the same order, and if the accuracy is inadequate, the next CNN model is tried. The attributes will change when implementing the frozen CNN. The VGG19 model is used in this study because it uses a deep neural network with 19 layers to effectively process images and train the model. The Fine-Tuned CNN is the final stage of the CNN, where a trained neural network's weights are used to initialize a new model that is trained on data from the same domain while fine-tuning. The Fine-Tuned CNN offers higher accuracy compared to the other two CNN models because the train and validation accuracy lines intersect, providing an accurate model result.

2.1.2.2 Strengths and Weaknesses

The authors deploy a Convolutional Neural Network (CNN), a well-known method for picture classification problems, as part of their deep learning strategy. The CNN is trained

on a large dataset of blood cell images, which is a strength of the paper. However, the paper does not include an analysis of the computational requirements of the proposed model. This information is important for practical considerations, such as the feasibility of deploying the model on resource-constrained devices.

2.1.3 Deep Learning Based Approach for Malaria Detection in Blood Cell Images [6]

The authors of this paper put forth a technique centered on a Convolutional Neural Network (CNN) for detecting malaria using cell images. They employed a 5-fold validation technique to train the CNN model. Comparison between both the conventional training and k-fold cross validation are carried out based on performance metrics.

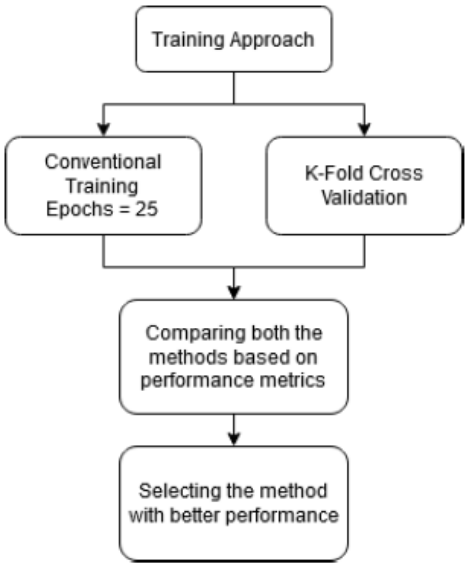


Figure 2.1.3.1 Training Methodology [6]

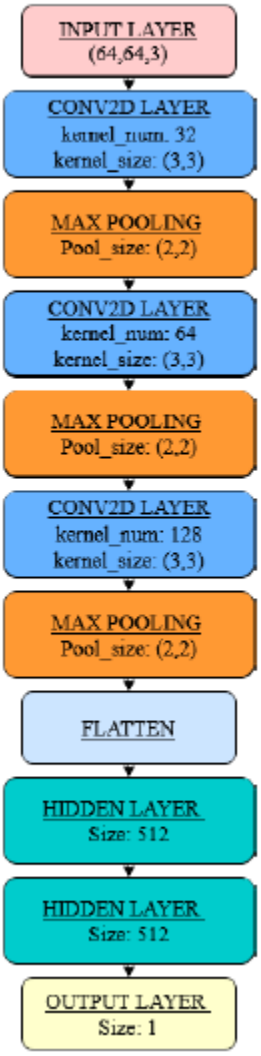


Figure 2.1.3.2 Proposed Model Architecture [6]

According to the figure above, every one of the three convolution layers in this configuration is followed by a max pooling operation, usually referred to as a max pooling layer. The convolution layers make use of the Rectified Linear Unit (ReLU) activation function. Feature maps are given activation functions to enhance a network's non-linearity. Any negative values in an activation map are all removed via ReLU activation and replaced with 0. Next, 1 neuron and sigmoid activation function are present in the model's output layer. It gives a value between 0 and 1 for binary classification. The Adam optimizer is used to compile the model, and its learning rate is set to 0.01. After that, the model is ready to be trained on the dataset. During the training phase, the model acquires the capacity to link inputs with outputs. The primary aim of the model training process is to determine a set of weights that are optimal for addressing the given problem. An epoch

is known as the neural network goes over the complete training set once, both forward and backward. After 25 epochs, the model achieved a training accuracy of 99.80%, whereas the validation accuracy remained steady at 94.56%.

K-fold Cross Validation is a method that helps create a less biased model compared to other techniques. In this method, the dataset is divided into K partitions or folds. Each fold is utilized K-1 times for training the model. In the specific case mentioned, a 5-fold cross-validation technique was employed, and the model was trained for 20 epochs within each fold. This strategy resulted in an impressive training accuracy of 99.95% and a validation accuracy of 99.61%. The training accuracy also increased by 0.15% when compared to the earlier method [6]. Performance comparison of both approaches can be seen on the table below. Therefore, authors of this paper choose k-fold cross validation method over the conventional method.

Metrics	Conventional Training	K-Fold CV
Training Accuracy	99.80%	99.95%
Validation Accuracy	94.56%	99.61%
Test Accuracy	94.48%	99.44%
F1 Score	95.54%	99.40%
AUC Score	95.48%	99.40%
Sensitivity	95.90%	99.92%
Specificity	95.12%	99.90%

Table 2.1.3.1 Performance comparison of both approaches [6]

2.1.3.2 Strengths and Weaknesses

The authors use a transfer learning approach, which involves fine-tuning a pre-trained CNN model (VGG16) on the malaria dataset. This method has proven to be successful in numerous image classification tasks, especially when dealing with limited dataset sizes. However, the dataset used in the paper is relatively small, with only 12,497 images. While this is a reasonable size for a medical imaging dataset, it may limit the generalizability of the model to other datasets or real-world scenarios.

2.1.4 Transfer Learning with ResNet-50 for Malaria Cell-Image Classification [7]

The primary objective of this research is to improve diagnostic accuracy by using transfer learning to classify malaria-infected cells. Transfer learning refers to a process in which a pre-existing neural network is first trained on a base dataset, and then the knowledge gained from this training is used to train a separate neural network on a different dataset and task. If the features prove suitable for both the base and target tasks, this method will be successful in comparison to solely using the base task. Pre-trained models have performed well in tasks related to image classification when used with similar dataNeural networks train each layer to perform the desired task while acquiring low- or high-level features. However, in residual learning, the ResNet model endeavors to acquire residual Relu activations rather than features. ResNet is a dependable pre-trained model for classifying medical images, according to previous research [7].

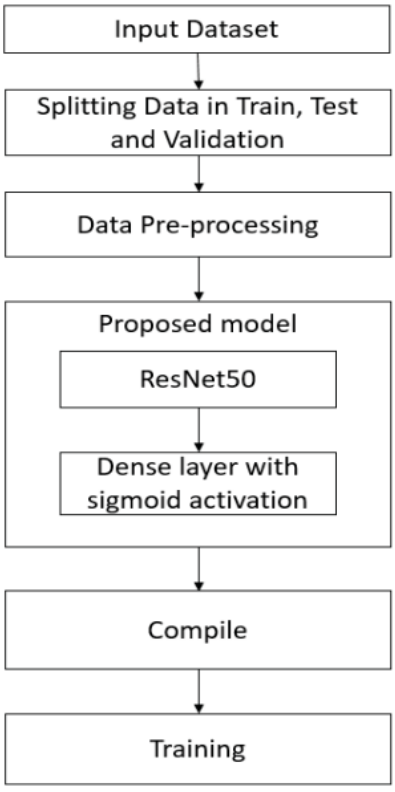


Figure 2.1.4.1 Proposed architecture [7]

First, the model's input dataset is an RGB image. Next, the data is split into three subsets: train, test, and validation. The model uses pre-trained weights to train the input data, with only the dense layer learning through backpropagation. The suggested model comprises a ResNet50 layer with pre-trained weights and a dense layer that employs sigmoid

activation as the final layer. Specific layers like Batch Normalization (BN) should remain trainable because the dataset might have distinct mean and variance values compared to the pre-trained weights. During the training phase of the model, the top layers of ResNet50 are not kept frozen, while the other layers are kept frozen. The process of fine-tuning the top layers is important as it adapts to the new dataset's mean or variance. To address this issue, auto-tuning is utilized for the BN layers in ResNet50, which is one of the top layers that should not be frozen. For binary classification of images in this task, the Sigmoid activation function is used as it performs well. The resulting model achieves a training accuracy of 95.91% and a validation accuracy of 95.4%.

2.1.4.2 Strengths and Weakness

The paper proposes the use of transfer learning with ResNet-50 for malaria cell-image classification, which has been shown to be effective in other image classification tasks. The results of the study demonstrate that the proposed model achieves high accuracy in malaria cell-image classification and outperforms other traditional machine learning methods. However, the study does not provide a detailed analysis of the feature maps generated by the deep learning model, which can provide insights into how the model makes its predictions.

2.1.5 Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images.[8]

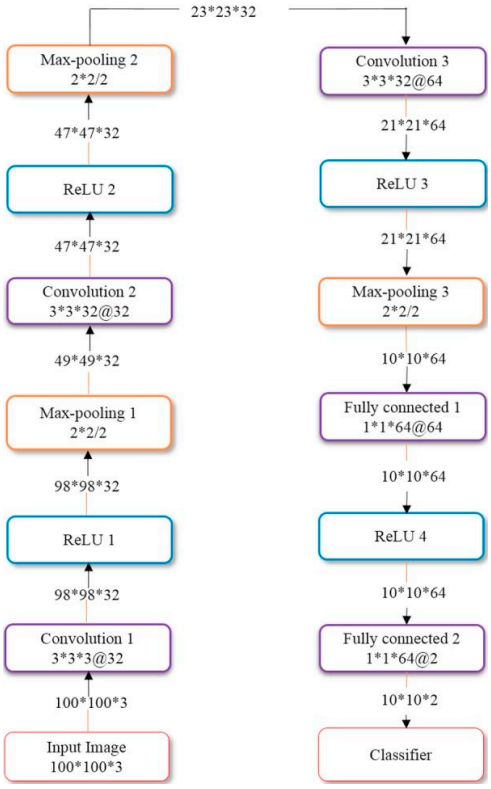


Figure2.1.5.1 Architecture of the customised model [8]

Rajaraman et al. (2018) investigate the applicaiton of pre-trained convolutional neural networks (CNNs) to improve the detection of malaria parasites in thin blood smear images. The research aimed to address the challenge of identifying and distinguishing malaria parasites in these images, a task that is labour-intensive and typically demands the skills of trained individuals.

The authors begin by introducing the problem of malaria and the need for accurate and efficient diagnosis. In their discussion, they mention the application of deep learning techniques for analyzing images and emphasize the effectiveness of pre-trained Convolutional Neural Networks (CNNs) for extracting features in medical image analysis. In this study, various pre-trained CNNs including VGG-19, ResNet50, and Inception-V3 were used as feature extractors to improve the detection of malaria parasites in thin blood smear images. The authors compared the performance of these pre-trained CNNs with traditional feature extraction methods such as SIFT and HOG, and with a baseline model which used a random forest classifier with handcrafted features.

The results show that the CNN-based models outperform the traditional feature extraction methods and the baseline model, achieving an accuracy of up to 96.25%. The study also

demonstrates the effectiveness of transfer learning, showing that pre-trained CNNs can be fine-tuned to further improve their performance [8].

In summary, the research underscores the promise of utilizing pre-trained CNNs as feature extractors in the field of medical image analysis. It highlights their effectiveness in the detection of malaria parasites within thin blood smear images. This could have noteworthy implications for enhancing the precision and efficiency of malaria diagnosis, particularly in regions where resources are scarce, and access to skilled professionals is limited.

2.1.5.2 Strengths and Weakness

The article proposes a novel approach for malaria detection that utilizes three-dimensional morphological reconstruction and neural network classification of optimal features. However, the proposed method requires a relatively large number of computational resources, which could limit its applicability in resource-limited settings with limited access to high-performance computing infrastructure.

2.2 Limitation of Previous Studies

- **Training data issues**

Training data is a crucial factor in the performance of deep learning models. Insufficient training data can limit the performance of a model, as the model may not have enough examples to learn from. In such cases, the model may be overfit to the available data or fail to generalize to new data. Furthermore, training datasets with imbalanced class distribution, where one class contains notably more instances than others, can lead to the development of biased models that exhibit subpar performance when it comes to the minority classes.

- **Vanishing Gradient**

The challenge of the vanishing gradient problem arises when, during the training of deep neural networks, the gradients become exceedingly small in the initial layers of the network. This can cause slow convergence or even prevent the network from converging.

- **Limited applicability**

For instance, Inception-v3 is designed for processing RGB images of fixed size, and it may not perform well on other types of data or images with different resolutions or colour spaces. Next, Residual Attention U-Net model is only tested on one dataset in this research, it may not apply well to other datasets.

Chapter 3

System Design

This chapter discuss the proposed deep learning techniques for Malaria cell detection. The proposed method includes data augmentation, customize ResNet50 model, Fine-tune, train, and evaluate the model.

3.1 System Design

3.1.1 Block Diagram of Overall Process

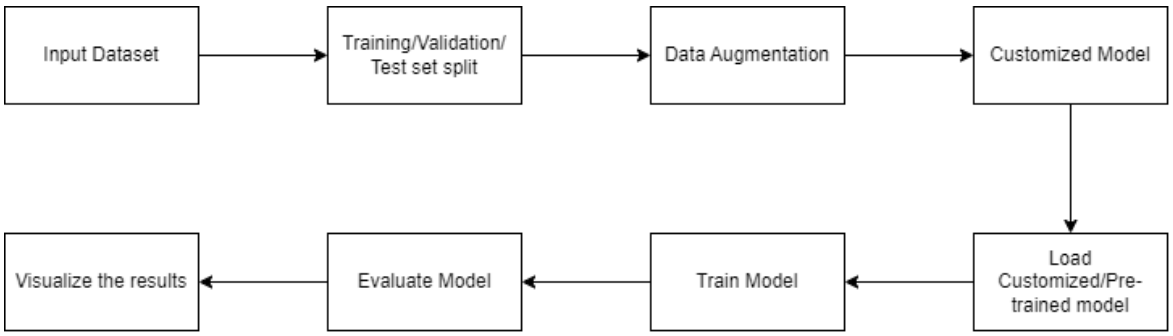


Figure 3.1.1 Block diagram of overall process

3.1.2 Data Source

The dataset used in the study was obtained from the NIH website, as mentioned on the Kaggle website. There are 27,558 cell images in the dataset, divided into two categories: infected and uninfected. Expert microscopists annotated the images after they were taken with a microscope at various magnifications [9].

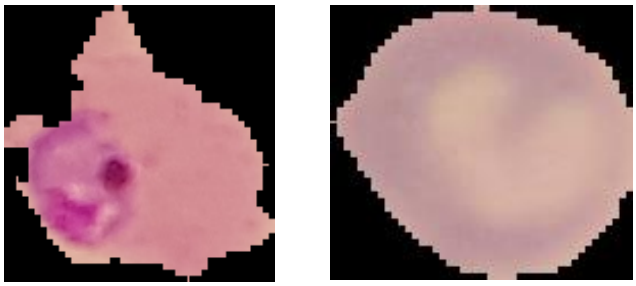


Figure 3.1.2 Sample of parasitized and uninfected cell

3.1.3 Splitting dataset

The dataset consists of a total of 27,558 cell images, which are categorized as either infected or uninfected. To prepare the data for training and evaluation of a deep learning model, the dataset was split into three subsets: train, test, and validation. The dataset was separated into three parts: 70% for training, 20% for testing, and 10% for validation.

Splitting data into training, testing, and validation sets is a standard approach in machine learning, and it is typical to allocate most of the data for training purposes. In this study, the training dataset contained 19,285 cell images, comprising both infected and uninfected cells. The rest of the images, i.e., 8,273, were divided into testing and validation subsets. The testing subset had 5,510 images, while the validation subset contained 2,755 images.

3.1.4 Data Augmentation

Transforms in this project include rotation, random resizing and cropping, vertical flipping, conversion to a PyTorch Tensor object, and normalization of pixel values. Rotation is performed randomly with an angle between -30 and 30 degrees, which can help the model learn to recognize objects regardless of their orientation in the input images. Random resizing and cropping are applied to capture variations in the size and aspect ratio of objects in the images. Vertical flipping is performed randomly with a 50% probability to introduce variety into the images. The final two transforms, conversion to a PyTorch Tensor object and normalization of pixel values, are essential for processing the image data within a PyTorch model. This transformation converts the input image to a Tensor object, while the Normalize transform standardizes the pixel values to have a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. This normalization step helps to reduce the impact of differences in lighting and color between images, which can improve the accuracy of the model. These transforms are designed to introduce variations into the input images, which can help the model become more robust and accurate.

3.1.5 Load Model

In this project, we have loaded three different models, each serving a distinct purpose and offering a range of capabilities. A more detailed model architecture will be explained in Chapter 3.2.

1. Customized Model

The customized model is tailored to the specific requirements of our project. We designed this model architecture from scratch, allowing us to fine-tune its structure to match the unique characteristics of our dataset and task. By creating a custom model, we have the flexibility to experiment with various layers, activations, and configurations to optimize its

performance. This model serves as a blank canvas, ready to be trained and adapted to the intricacies of our problem.

2. ResNet-18

ResNet-18 is a widely recognized convolutional neural network architecture renowned for its efficacy in image classification tasks. Its design comprises multiple convolutional layers that incorporate skip connections, which enhances its suitability for deep learning tasks. Additionally, it is a network architecture that can be constructed entirely from the ground up, allowing for customization and adaptation to specific image classification requirements.

3. ResNet-50

ResNet-50 is another variant of the ResNet architecture but deeper and more complex than ResNet-18. It offers a higher capacity to capture intricate features in images, making it suitable for more demanding tasks and larger datasets. We have loaded a pre-trained ResNet-50 model, which provides a stronger starting point for transfer learning. While it may require more computational resources, ResNet-50 can potentially achieve even better performance on challenging image analysis tasks.

By incorporating these three models into our project, we can employ a combination of customizability, transfer learning, and model complexity to address a wide spectrum of challenges. Depending on the specific requirements of our task and the available computational resources, we have the flexibility to choose the most suitable model or even combine their strengths to achieve optimal results.

3.1.6 Model Training

To train a deep learning model, allowing it to learn from a labelled dataset. The model architecture can be customized according to the specific classification problem. This pipeline is designed for training deep learning models, particularly those used in classification tasks. The function takes several important parameters, including the model to be trained, data loaders for training and validation datasets, the number of training epochs, learning rate, and a patience threshold for early stopping.

Within the training loop, the code adheres to best practices for deep learning training:

Loss Function and Optimizer: It initializes the loss function as cross-entropy loss and uses the Adam optimizer for gradient-based optimization. Cross-entropy loss is suitable for classification tasks, making it a commonly used choice.

Device Selection: The code checks for the availability of a GPU (CUDA) and automatically assigns the appropriate device for training. This dynamic device selection enables GPU acceleration when possible, improving training speed.

Training and Validation Loops: The training loop iterates through each epoch, tracking training loss, accuracy, and the number of correctly classified samples. Simultaneously, the validation loop assesses model performance on a separate validation dataset, calculating validation loss and accuracy.

Early Stopping Mechanism: To prevent overfitting, the code implements early stopping. If the validation loss does not improve for a specified number of consecutive epochs (controlled by the patience parameter), training is halted early. This helps ensure that the model does not continue to learn noise in the data and maintains better generalization.

Logging and Progress Display: After each epoch, the code logs and displays training and validation metrics, including loss and accuracy. This allows for real-time monitoring of model training progress and performance.

Result Storage: The function returns lists containing training and validation losses, as well as training and validation accuracies. These results can be further analysed or visualized to gain insights into model behaviour.

3.1.7 Evaluate Model

The objective of model evaluation in machine learning is to assess how well a trained model performs its classification task. Two essential tools for this purpose are the confusion matrix and the classification report.

The confusion matrix provides a detailed breakdown of the model's predictions, categorizing them into true positives (correctly predicted positives), true negatives (correctly predicted negatives), false positives (incorrectly predicted positives), and false negatives (incorrectly predicted negatives). This matrix facilitates the computation of multiple metrics like accuracy, precision, recall, specificity, and the F1-score, providing a deeper understanding of the model's ability to differentiate between distinct classes.

The classification report offers a concise summary of key classification metrics derived from the confusion matrix. It includes precision (the ability to make correct positive predictions), recall (the ability to identify all actual positives), and the F1-score (a balance between precision and recall). Additionally, it provides support counts for each class, accuracy, and macro, weighted, and micro averages. These metrics enable us to evaluate the model's performance on a per-class basis and account for class imbalances.

Together, the confusion matrix and classification report are valuable tools for understanding a model's strengths and weaknesses, identifying any bias or misclassification tendencies, and making informed decisions about model improvement or deployment in real-world applications. A more detail calculation will be mentioned in chapter 5.

3.1.8 Testing Model

First, the code ensures that the model is in evaluation mode, which means that certain operations like dropout and batch normalization are disabled to ensure consistent and deterministic predictions during inference. Next, it defines a utility function called `get_class_name(index)`, which converts class indices (0 for Parasitized and 1 for Uninfected) into their corresponding class names. This function will be used to label the predicted and actual classes. The interactive aspect of the viewer is enabled using the `ipywidgets` library. It creates a widget that allows you to interactively select an index, which corresponds to an image in your test dataset. The selected index determines which image the viewer will display and predict.

Within the `show_images` function, the code performs the following steps:

1. It checks whether the selected index corresponds to an infected (class 0) or uninfected (class 1) image.
2. It retrieves the image and its true label from the test dataset based on the selected index.
3. Using the trained model, it predicts the label for the selected image. This involves passing the image through the model and determining both the predicted label and the confidence score associated with the prediction.
4. The code then creates a subplot and displays the selected image within it.

5. It determines and displays both the actual (true) label and the predicted label for the image. These labels are styled using HTML to distinguish between them. Specifically, predicted labels are displayed in red if they represent Parasitized and in green if they represent Uninfected.

6. Finally, the output area is updated and displayed, showing the image, its prediction, and the associated confidence score. The interactive widget allows user to easily select different images from dataset to view and analyse.

In summary, this interactive image viewer provides an intuitive and informative way to assess how well the model performs on individual images. It visually presents the model's predictions alongside actual labels, making it a valuable tool for inspecting and understanding the model's performance on specific cases within the test dataset.

3.2 System Architecture

3.2.1 Custom Model Architecture

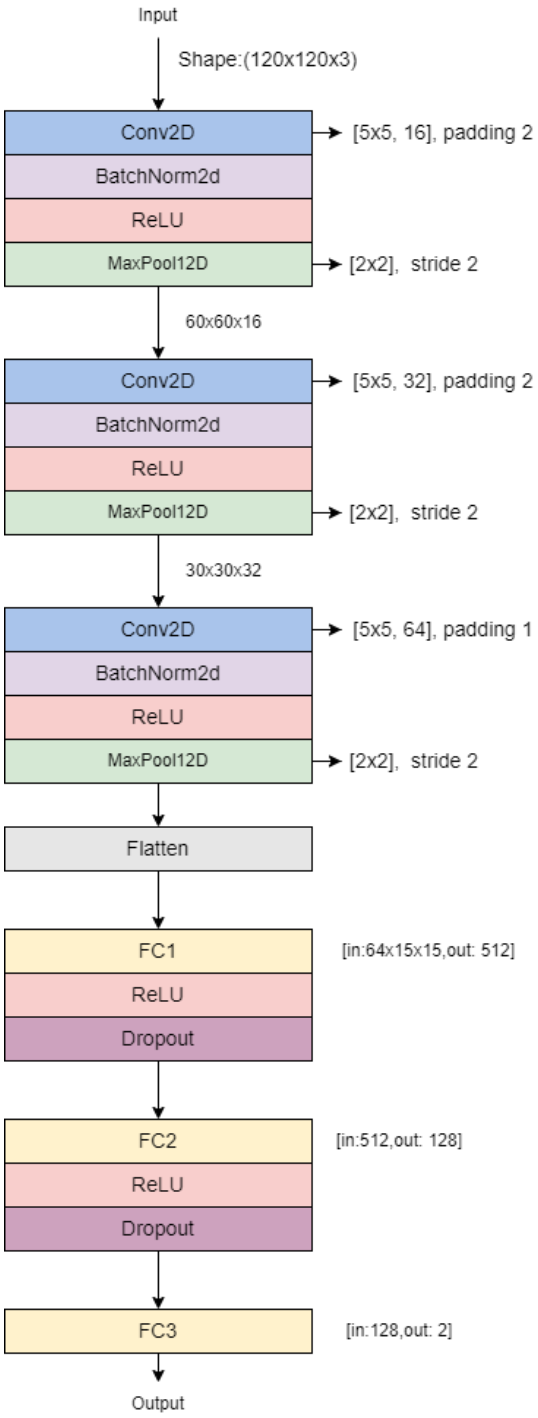


Figure 3.2.1 Customized model architecture diagram

The architecture presented here is a Convolutional Neural Network (CNN) tailored for image classification tasks. It comprises several layers, each serving a specific purpose in feature extraction and classification. The input to this model consists of colour images with three channels, representing the Red, Green, and Blue (RGB) components.

The initial layer, known as Layer 1, commences the network's operations. It features a convolutional layer with 16 output channels. This layer employs a 5x5 kernel with padding set to 2, preserving the spatial dimensions of the input. Following the convolution, Batch Normalization is applied to normalize the output, followed by Rectified Linear Unit (ReLU) activation to introduce non-linearity. Subsequently, a MaxPooling layer with a 2x2 kernel and a stride of 2 reduces the spatial dimensions. Layer 2 builds upon the previous layer's output. It includes another convolutional layer with 32 output channels, identical kernel size, padding, and activation function as Layer 1. Like before, Batch Normalization, ReLU activation, and MaxPooling further process the data. Layer 3 continues the feature extraction process with a 3x3 kernel convolutional layer, producing 64 output channels. The pattern of Batch Normalization, ReLU activation, and MaxPooling remains consistent. After these convolutional layers, the network transitions to fully connected layers for classification. The output from Layer 3 is flattened into a vector, which is then processed by a series of fully connected layers. The first fully connected layer reduces the features to 512 dimensions, followed by a ReLU activation and a dropout layer to mitigate overfitting. A second fully connected layer further reduces the dimensionality to 128, followed by another ReLU activation and dropout. The final fully connected layer, serving as the output layer, produces the logits for binary classification.

In summary, this CNN architecture follows a sequence of convolutional, normalization, activation, and pooling layers for feature extraction, followed by fully connected layers for classification. Dropout layers are incorporated to enhance generalization. Overall, it is a robust model for image classification tasks, particularly binary classification.

3.2.2 ResNet18 Model Architecture

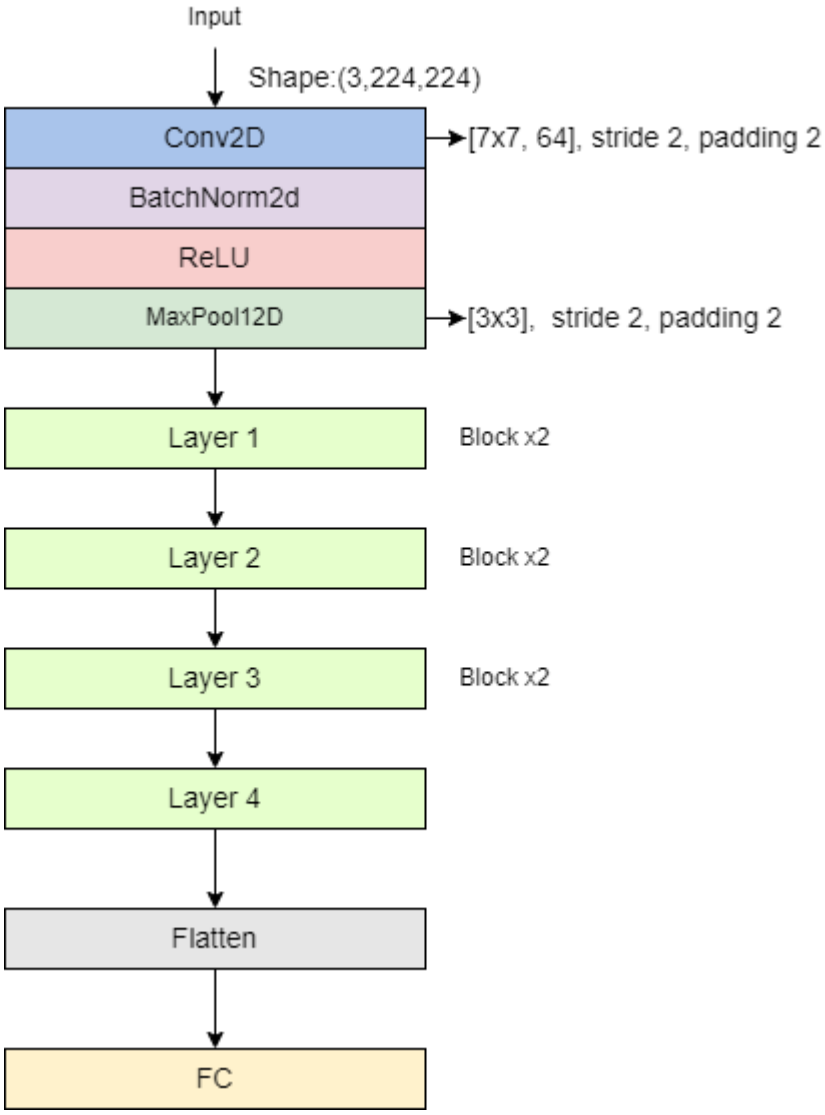


Figure 3.2.2 ResNet18 model architecture diagram

The network begins with an initial convolutional layer, which takes input images, typically of size 224x224 pixels. This layer applies a 7x7 convolution with 64 output channels, followed by batch normalization and rectified linear unit (ReLU) activation. A stride of 2 is used, reducing the spatial dimensions of the feature maps.

Following the initial convolution, a max-pooling layer with a 3x3 kernel and a stride of 2 is applied, further reducing the spatial dimensions of the feature maps.

The core of ResNet-18 consists of four residual blocks, each containing multiple basic blocks. A residual block is defined by its skip connections, allowing gradients to flow effectively during training. Let's focus on the first two residual blocks as an example:

Residual Block 1 consists of two basic blocks:
- Basic Block 1: This block includes two convolutional layers with batch normalization and ReLU activation.
- Basic Block 2: Similar to the first block, it contains two convolutional layers with batch normalization, ReLU activation, and a skip connection. The key feature here is the skip connection, which directly adds the input to the block to the block's output.

Residual Block 2 follows a similar structure but incorporates the output of Residual Block 1 as part of its skip connection. This means that the output of Residual Block 1 is added to the output of Residual Block 2, allowing information to flow from earlier layers to deeper layers of the network. After the residual blocks, a global average pooling layer reduces the spatial dimensions to 1x1. The resulting tensor is flattened and passed through a fully connected layer to produce the final classification output.

In summary, ResNet-18's strength lies in its residual connections, which facilitate training very deep networks while maintaining gradient flow. This architecture has demonstrated remarkable performance in various computer vision tasks and serves as a foundation for more advanced variants in the ResNet family. Visualizing this architecture with its intricate skip connections can provide a clearer understanding of its design and effectiveness in deep learning applications.

3.2.3 ResNet50 Model Architecture

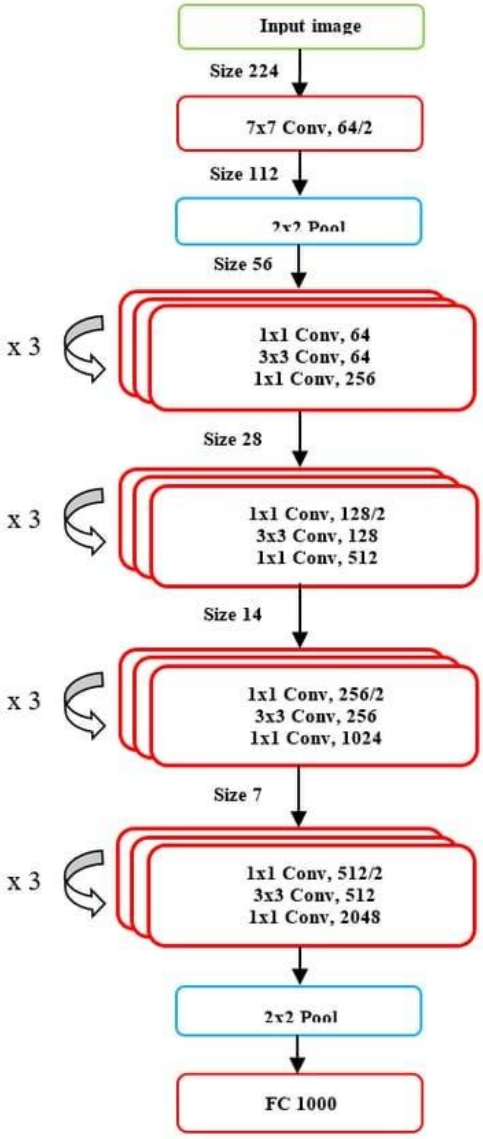


Figure 3.2.3 ResNet50 Model Architecture [12]

ResNet50, developed by Kaiming He and his team in 2015, is a CNN architecture widely utilized for image classification tasks. The name ResNet is derived from the use of residual connections in the model, which help alleviate the issue of vanishing gradients in deep neural networks. ResNet50 is a well-known and frequently used deep learning model in the computer vision community, particularly for image classification tasks [13].

An image is provided as input to the ResNet50 model, which undergoes a series of operations, including convolutional layer, batch normalization, ReLU activation, and max pooling. The architecture consists of 50 layers and is based on the idea of residual learning, which aims to solve the issue of vanishing gradients in deep neural networks.

Following the initial block, the output is fed through several residual blocks, each comprising three convolutional layers with batch normalization and ReLU activation. In these blocks, the shortcut connection omits the convolutional layers and instead directly combines the input with the output obtained from the final convolutional layer in the block.

There are a total of 5 residual blocks in ResNet50, each with increasing numbers of filters (64, 128, 256, and 512) and decreasing spatial dimensions due to max pooling. After the final residual block, the output is passed through average pooling to reduce the spatial dimensions to a 1x1 feature map. This feature map is then flattened and passed through a fully connected layer with softmax activation to produce the final classification output.

ResNet50 is composed of various blocks, with each block containing numerous convolutional layers, batch normalization, and ReLU activation functions. The blocks are created to progressively decrease the spatial size of the input while simultaneously augmenting the number of channels. The final output of the network is a softmax function, which produces a probability distribution over the possible classes.

ResNet50 has demonstrated remarkable results on various benchmark datasets, including ImageNet, which comprises more than a million images categorized into 1000 classes. The architecture has been used in a wide range of computer vision applications, including object detection, segmentation, and image captioning.

3.3 Deep Learning techniques used

3.3.1 Transfer Learning

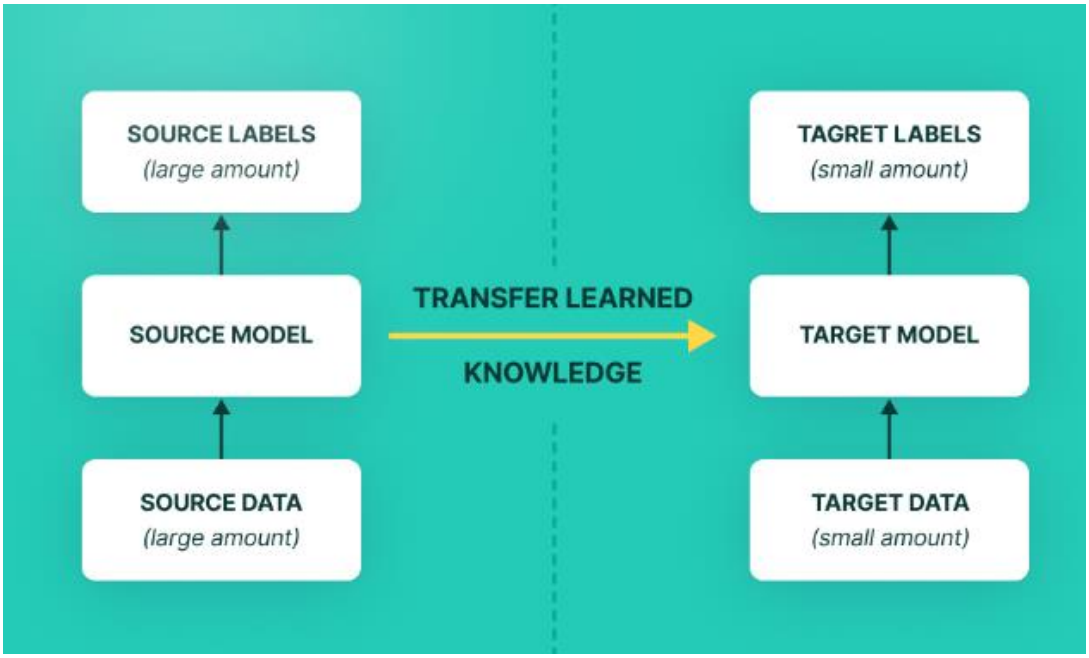


Figure 3.3.1 Transfer Learning [10]

Transfer learning is a machine learning technique that involves using a model trained on one task to improve the learning efficiency and accuracy of a model trained on a related but different task. Rather than beginning with a blank slate, transfer learning involves transferring the pre-existing knowledge of a pre-trained model to a new model, enabling it to learn from the pre-trained model's characteristics and modify them to suit the new undertaking [11]. The application of transfer learning has become prevalent in several domains, including natural language processing and computer vision, where pre-trained models like BERT have achieved impressive results on different tasks. By using transfer learning, researchers can build models that require less data and training time while still performing well on multiple tasks.

The transfer learning approach used in this project allows the model to take advantage of the pre-trained ResNet50's knowledge of general image features, while still being able to learn the specific features required to classify the cell images as parasitized or uninfected. The training process updates only the parameters of the last fully connected layer, allowing for faster training and better performance compared to training the entire network from scratch.

3.3.2 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a widely utilized optimization algorithm in deep learning, including in ResNet50. During the training phase of ResNet50, SGD is employed to adjust the network's weights.

ResNet50 takes a mini-batch of input images and passes them through the network to compute the predicted outputs during training. To quantify the disparity between predicted outputs and actual labels, a loss function such as cross-entropy is employed. SGD employs backpropagation to calculate the gradient of the loss function about the weights of the network. This gradient indicates the direction in which the weights should be updated to reduce the loss. The weights are updated using the gradient and a learning rate. The magnitude of the weight update during training is governed by the learning rate. This process is reiterated for several mini-batches until the entire training set is utilized. This constitutes one epoch of training. ResNet50 typically uses a variant of SGD called stochastic gradient descent with momentum (SGDM) which includes an exponentially decaying average of the gradients to update the weights. This helps to smooth out the weight updates and prevent the optimization process from getting stuck in local minima. Additionally, ResNet50 also uses a technique called weight decay, which adds a penalty term to the loss function to encourage the weights to have small magnitudes. This helps to prevent overfitting and improve generalization performance.

By using SGD with momentum and weight decay, ResNet50 can optimize its weights effectively and achieve state-of-the-art performance in image classification tasks.

Chapter 4

Experiment/Simulation

4.1 Tools to Use

Description	Specifications
Model	Acer Nitro AN515-55
Processor	Intel Core i5-10300H CPU @ 2.50GHz, 2496 Mhz
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650Ti
Memory	8GB DDR4 SDRAM
Storage	512 GB SSD

Table 4.1 Hardware used in this project

Software utilized for this project are listed as below:

- Google Drive

  Google Drive is a cloud-driven platform created by Google that enables users to store, retrieve, and collaborate on various types of files across devices. It integrates seamlessly with other Google tools especially Google Colab, making it convenient for collaborative work and accessing files online. With Google Drive, dataset can store and access which can be directly used in Google Colab.

- Google Colaboratory (Google Colab)

  Google Colab and Google Drive are tightly integrated, enabling seamless collaboration and code sharing in a cloud-based environment. This synergy simplifies data analysis and code development, making it a valuable ecosystem for individuals and teams working with Python and related tasks in a collaborative setting.

- Software Libraries
  - Torch, or PyTorch, is a Python-based open-source framework primarily used for deep learning and AI. It offers dynamic computation graphs, automatic differentiation, GPU support, and a rich set of pre-built neural network components. It's popular in research and industry for tasks like image recognition, NLP, and reinforcement learning due to its flexibility and strong developer community.
  -

4.2 Implementation Issues and Challenges

- **Memory usage**

The free version of Google Colab comes with limited storage space. These models has many parameters, which can make it memory intensive. This can be a problem for devices with limited memory. In addition, users are restricted by limited storage capacity in Google Drive, with only 15GB available per user account. As a result, saving the dataset images and training parameters by the model may necessitate a significant amount of storage space.

- **Computational complexity**

Training large deep learning models can be challenging on the free version of Google Colab as it offers access to a restricted amount of computing resources, which may cause a bottleneck in the process. The models in this project are deep network with many layers, which makes it computationally expensive to train and evaluate.

- **Overfitting**

ResNet50 can be prone to overfitting, where it memorizes the training data instead of generalizing to new data. Overfitting may result from using an extremely high number of epochs to train the model. On the other hand, underfitting may occur if the number of epochs is too low.

4.3 Project Timeline

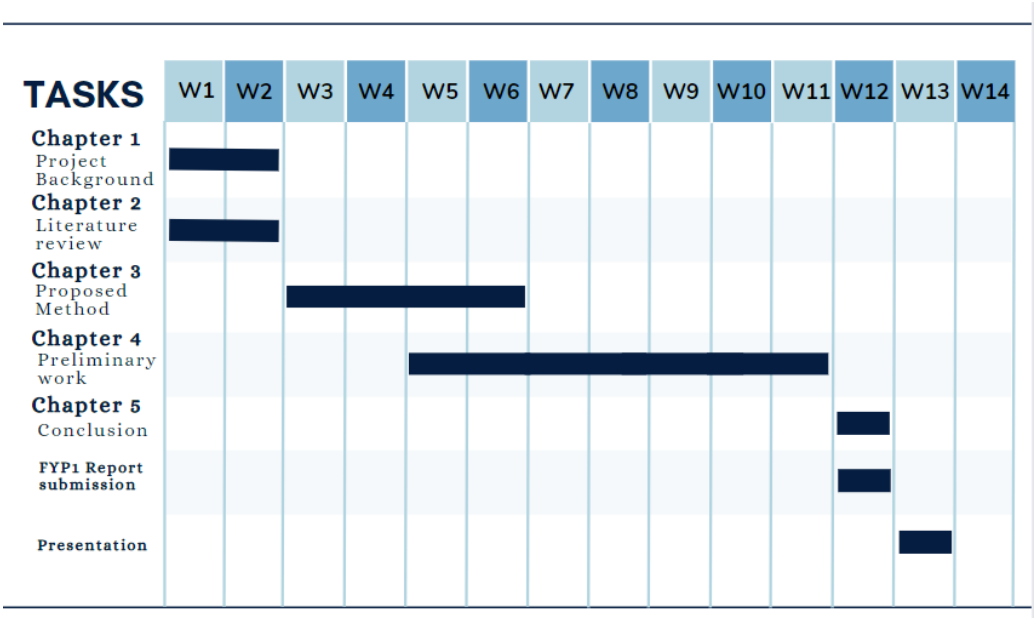


Figure 4.3.1 Project Timeline for FYP 1

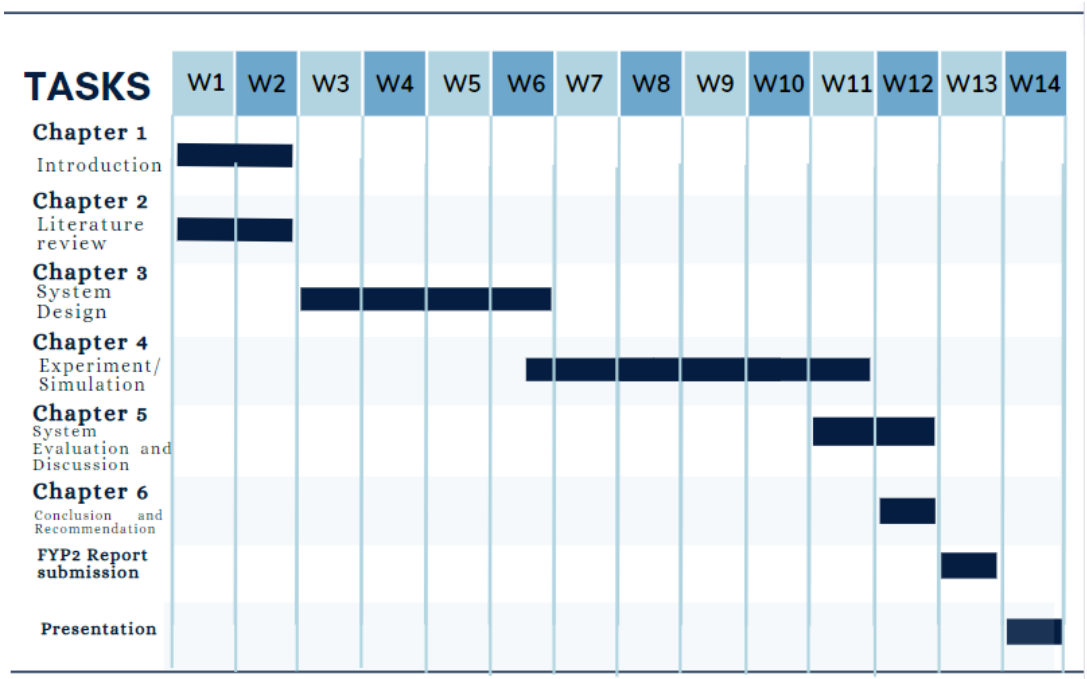


Figure 4.3.2 Project Timeline for FYP 2

Chapter 5

System Evaluation and Discussion

5.1 Model Performance Definition

The following equations will utilize the values mentioned to compute their outcomes:

- True Positives (TP): This represents the count of instances that have been accurately identified as positive, meaning they were correctly classified as malaria-positive.
- True Negatives (TN): This signifies the count of instances that have been correctly recognized as negative, indicating they were correctly classified as malaria-negative.
- False Positives (FP): This denotes the count of instances that have been incorrectly classified as positive, essentially, they were falsely categorized as malaria-positive when they were not.
- False Negatives (FN): This indicates the count of instances that have been inaccurately categorized as negative, signifying they were missed or overlooked as malaria-negative when they were malaria-positive.

- Accuracy

Accuracy serves as a widely utilized evaluation metric in the realm of machine learning and classification. It quantifies the fraction of accurately categorized instances or data points among the entire datasets. In a binary classification context, accuracy is calculated as:

$$Accuacy = \frac{TP + TN}{TP + FP + FN + TN}$$

- Confusion Matrix

A confusion matrix is a useful tool for evaluating the performance of a classification model, such as one used for malaria detection. It provides a summary of how well the model is performing in terms of making correct and incorrect predictions. A typical confusion matrix for a binary classification problem like malaria detection has the following components:

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

- Precision, Recall and F1 Score

Precision measures the accuracy of the positive predictions made by the model. In the context of malaria parasite detection, it quantifies how many of the predicted malaria parasite instances are true positive cases (correctly identified parasites) out of all the predicted positive instances. Precision is calculated as:

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

where TP is the number of correctly detected parasites and FP is the number of non-parasite regions incorrectly classified as parasites. A high precision value signifies that the model produces minimal false positive predictions, a critical aspect in medical image analysis to prevent unnecessary treatments or additional tests for patients.

Recall, sometimes referred to as sensitivity or the true positive rate, gauges the model's capacity to accurately detect all true positive cases (in this context, malaria parasites) among all the positive instances present in the dataset. Recall is calculated as:

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

where TP is the number of correctly detected parasites and FN is the number of parasite regions that were missed or not detected).

In malaria detection, high recall is crucial because missing even a single parasite can have serious consequences for a patient's health. Therefore, it's important to minimize false negatives.

The F1 score is the harmonic mean of precision and recall and provides a balanced measure of a model's performance, especially in situations where class imbalances exist.

F1 Score is calculated as:

$$F1\ Score = \frac{2(Precision\ x\ Recall)}{(Precision + Recall)}$$

It balances the trade-off between precision and recall. A high F1 score indicates that the model achieves both high precision and high recall simultaneously, making it a suitable metric for malaria parasite detection, where both false positives and false negatives should be minimized.

In the context of malaria parasite detection, a model with high precision ensures that most of the identified parasites are accurate, reducing the likelihood of false alarms. High recall ensures that the model doesn't miss any parasites, minimizing the chances of false negatives. The F1 score provides a single metric that considers both precision and recall, helping to strike a balance between these two important aspects of detection accuracy.

5.2 Training Results and Analysis

5.2.1 Custom Model

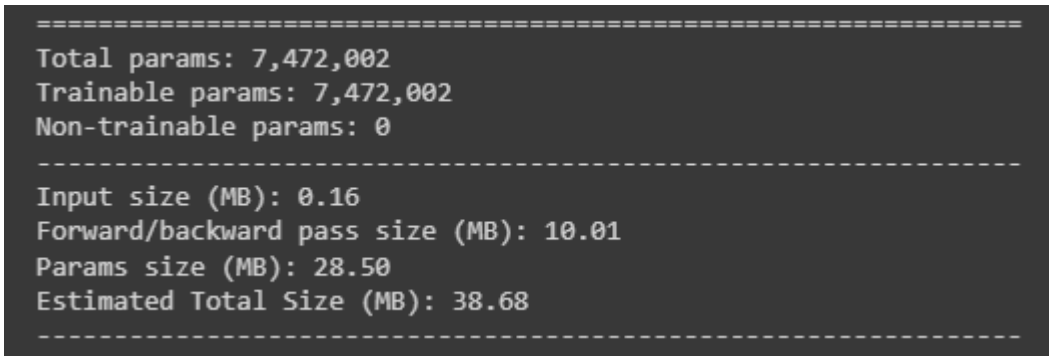


Figure 5.2.1 Summary of customized model

The list of parameters in a neural network model, separated into trainable and non-trainable parameters, is provided above:

Total params indicates how many parameters there are in the entire neural network. The weights and biases connected to the layers of the neural network are referred to as parameters in the context of deep learning models. To train the network to accomplish a certain task, these parameters are discovered. For this custom model, it has a total of 7,472,002 parameters.

Trainable params: These are the parameters that the model will learn or update during the training process. They include the weights and biases of the neural network layers. All 7,472,002 parameters in this model are trainable, meaning that the optimisation algorithm can change them to reduce the loss function and enhance the model's performance on training data. This also implies that for the model to effectively complete its duty, they will be updated and taught during the training phase.

Non-trainable params: These are parameters that are not updated during training. In most cases, non-trainable parameters are associated with layers that perform operations without learnable weights. For example, Batch Normalization layers may have non-trainable parameters for scaling and shifting that are computed based on the training data but remain

fixed during inference. In this model, there are no non-trainable parameters, indicating that all parameters are subject to training updates.

Metrics	Values
Training accuracy	95.18
Validation accuracy	93.55
Training Loss	0.1432
Validation Loss	0.1792

Table 5.2.1 Training Performance

The calculated average training and validation metrics offer valuable insights into the model's performance and training process. On average, the model achieves a training accuracy of approximately 95.18%, indicating that it effectively classifies the training data with high accuracy during the training phase. This reflects the model's ability to learn and fit the training dataset well. Furthermore, the average validation accuracy of around 93.55% suggests that the model generalizes reasonably well to unseen validation data. This is a crucial indication that the model can make accurate predictions on new, previously unseen examples, demonstrating its robustness. In terms of average training and validation losses, the model shows an average training loss of 0.1432 and an average validation loss of 0.1792. Lower loss values indicate that y the model is effectively minimizing errors during both training and validation, further supporting its overall good performance. These results collectively indicate that the model performs well in terms of accuracy and generalization, making it a promising candidate for the given task.

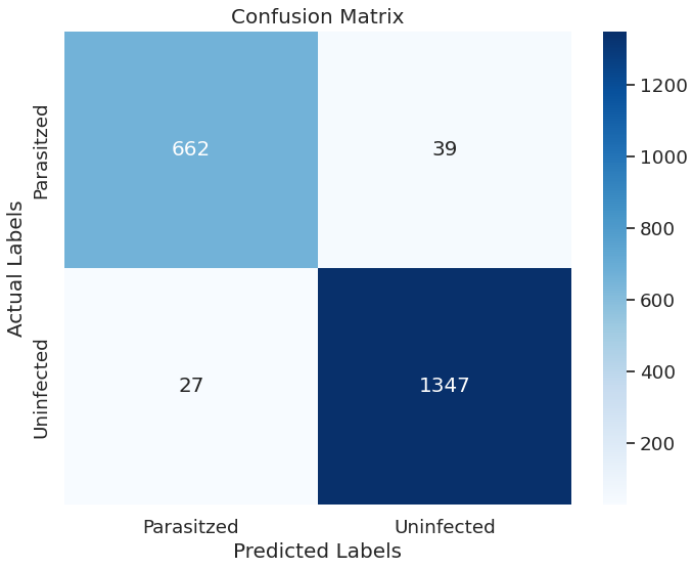


Figure 5.2.2 Confusion Matrix

Report:

	precision	recall	f1-score	support
0	0.96	0.94	0.95	701
1	0.97	0.98	0.98	1374
accuracy			0.97	2075
macro avg	0.97	0.96	0.96	2075
weighted avg	0.97	0.97	0.97	2075

Figure 5.2.3 Classification Report

Precision which measures the accuracy of the model's positive predictions for each class, indicates that the model achieves 96% accuracy for class 0 and 97% accuracy for class 1. The F1-score, which balances precision and recall, is 0.95 for class 0 and 0.98 for class 1, demonstrating the model's ability to strike a balance between precision and recall for both classes. The support values represent the number of samples in each class, with 701 samples for class 0 and 1374 samples for class 1. Next, Macro average calculates metrics independently for each class and then computes the average. In this case, the macro-averaged precision is 0.97, macro-averaged recall is 0.96, and macro-averaged F1-score is 0.96. Macro average gives equal weight to each class, making it useful when you want to evaluate the model's performance across different classes without considering class imbalances. Furthermore, weighted-average precision, recall, and F1-score, on the other hand, take into account class imbalances. They are all 0.97, indicating strong overall model performance. Weighted average assigns more weight to classes with more samples, providing a better representation of the model's performance in real-world scenarios where class sizes are unequal.

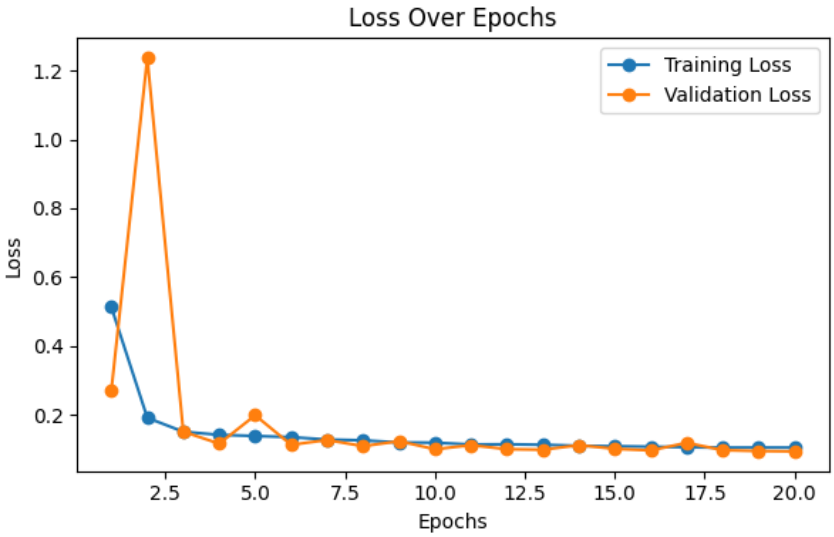


Figure 5.2.4 Graph of Validation loss and Train loss

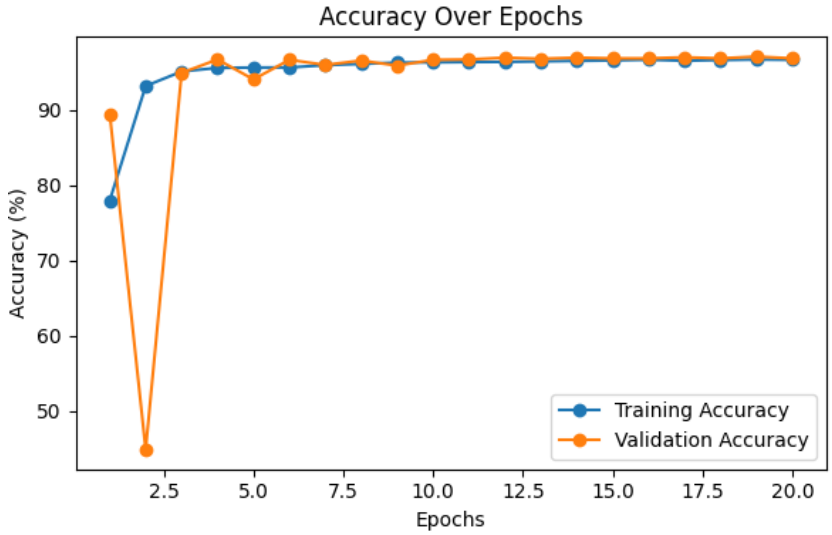


Figure 5.2.5 Graph of Validation accuracy and Train accuracy

5.2.2 ResNet18

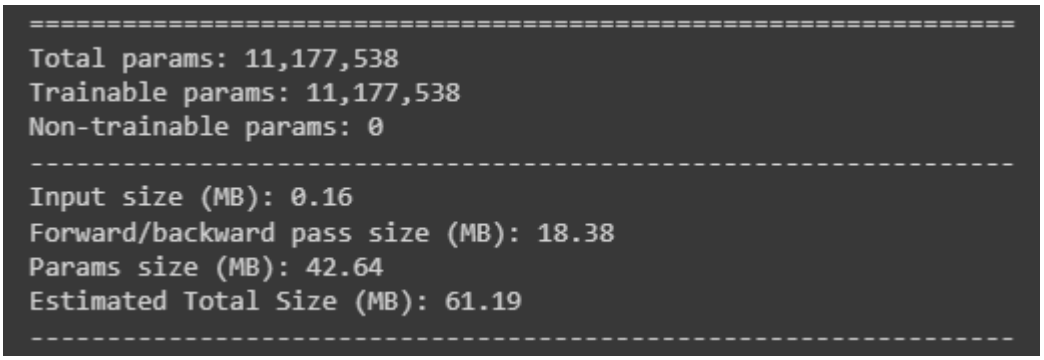


Figure 5.2.6 Summary of ResNet18

Metrics	Values
Training accuracy	92.67
Validation accuracy	95.30
Training Loss	0.1903
Validation Loss	0.1503

Table 5.2.2 Training Performance

On average, the model achieves a training accuracy of approximately 92.67%, signifying its ability to correctly classify the training data with a high degree of accuracy during the training process. This means that the model has effectively learned from the training dataset. Furthermore, the average validation accuracy of around 95.30% demonstrates that, the customized model generalizes well to unseen validation data. In terms of average training and validation losses, the model exhibits an average training loss of 0.1903 and an average validation loss of 0.1503. Lower loss values are generally desirable as they indicate that the model is minimizing errors during both training and validation.

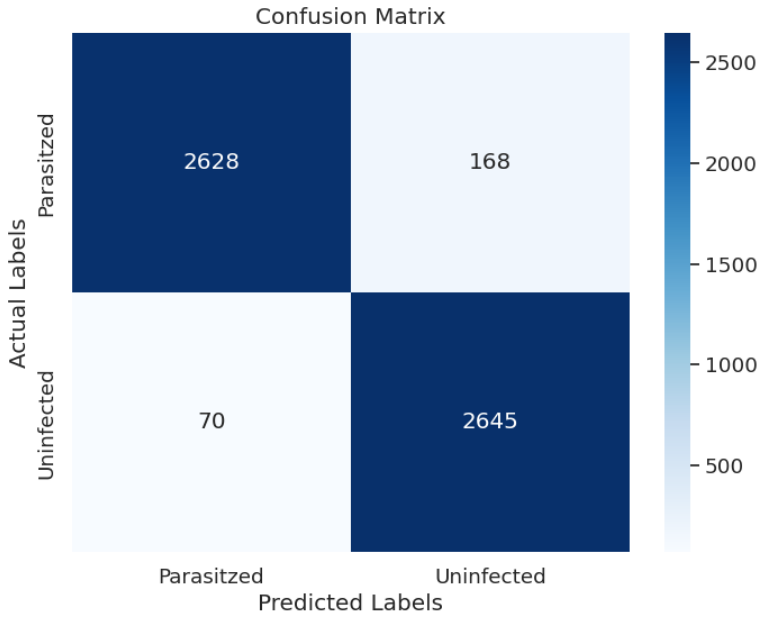


Figure 5.2.7 Confusion Matrix

Report:

	precision	recall	f1-score	support
0	0.98	0.94	0.96	2796
1	0.94	0.98	0.96	2715
accuracy			0.96	5511
macro avg	0.96	0.96	0.96	5511
weighted avg	0.96	0.96	0.96	5511

Figure 5.2.8 Classification Report

The classification report shows the model's performance in a binary classification task. It has precision values of 0.98 for class 0 (Parasitized) and 0.95 for class 1 (Uninfected), indicating high accuracy in predicting both classes. The recall scores indicate that the model is capable of accurately identifying positive instances, with a score of 0.95 for class 0 and 0.98 for class 1. Additionally, the F1-score, which provides a balance between precision and recall, is 0.96 for both classes. Overall accuracy is 0.96, reflecting the proportion of correctly classified samples. The macro and weighted averages confirm strong model performance.

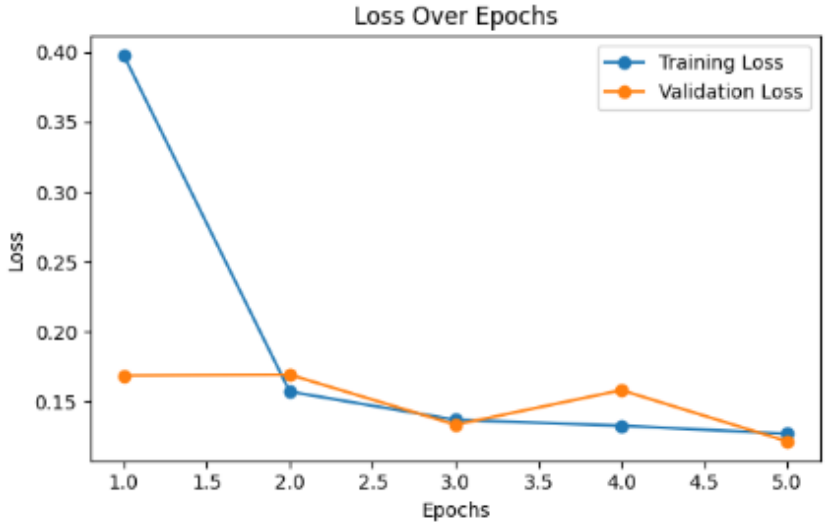


Figure 5.2.9 Graph of Validation loss and Train loss

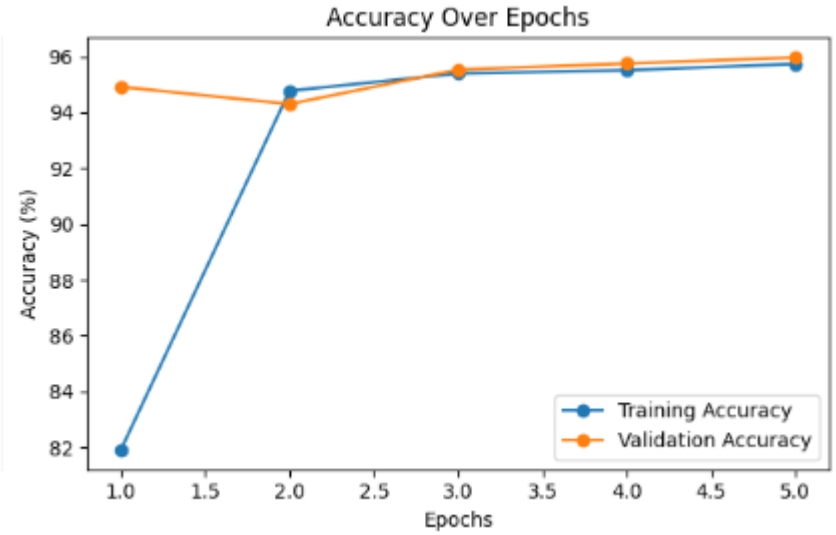


Figure 5.2.10 Graph of Validation accuracy and Train accuracy

5.2.3 ResNet50

```
================================================================
Total params: 23,512,130
Trainable params: 4,098
Non-trainable params: 23,508,032
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 286.55
Params size (MB): 89.69
Estimated Total Size (MB): 376.82
----------------------------------------------------------------
```

Figure 5.2.11 Summary of ResNet50

The model has many total parameters, but only a small fraction of them (4,098) are trainable, while the majority (23,508,032) are non-trainable. This suggests that a significant portion of the model consists of pre-trained layers or modules, which can be beneficial for transfer learning or fine-tuning on specific tasks while keeping most of the pre-trained knowledge intact.

Metrics	Values
Training accuracy	92.36
Validation accuracy	92.97
Training Loss	0.2005
Validation Loss	0.1875

Table 5.2.3 Training Performance

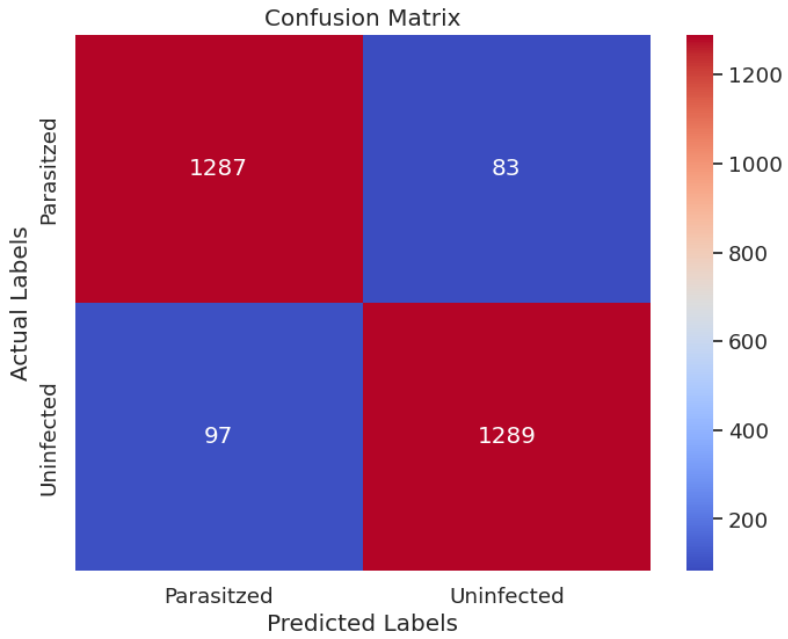


Figure 5.2.12 Confusion Matrix

	precision	recall	f1-score	support
0	0.93	0.94	0.93	1370
1	0.94	0.93	0.93	1386
accuracy			0.93	2756
macro avg	0.93	0.93	0.93	2756
weighted avg	0.93	0.93	0.93	2756

Figure 5.2.13 Classification Report

The classification report presented here provides a concise summary of the performance evaluation of a binary classification model. The model is made to distinguish between classes 0 and 1 in the current case. With class 0 at 0.93 and class 1 at 0.94, the precision values for both classes are high, demonstrating that the model correctly predicts positive outcomes for both classes. Like this, both classes' recall values—class 0 at 0.94 and class 1 at 0.93—are high, indicating that the model can successfully identify most real positive cases. The F1-scores for both classes are 0.93, which demonstrates a balanced performance in terms of precision and recall. The accuracy of the model across the entire dataset is 93%, reflecting its overall effectiveness in correctly classifying samples. The macro-averaged and weighted-averaged metrics further support the model's robust performance, considering any class imbalances that may exist in the dataset.

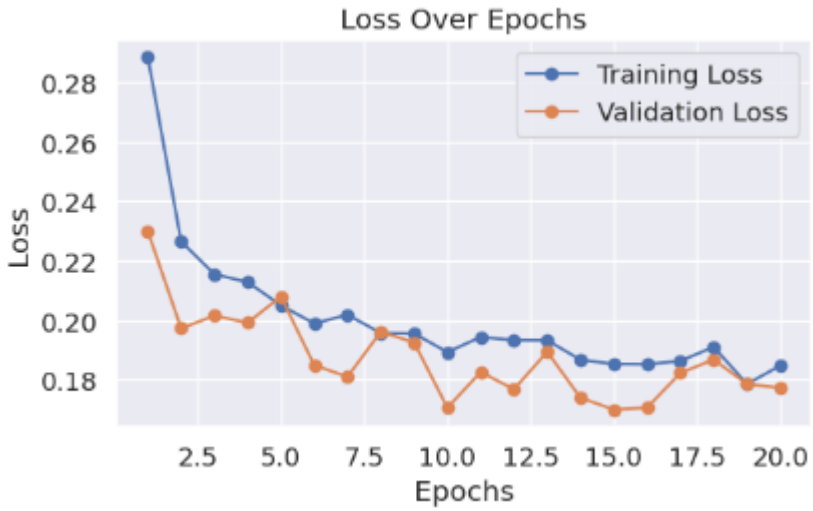


Figure 5.2.14 Graph of Validation loss and Train loss

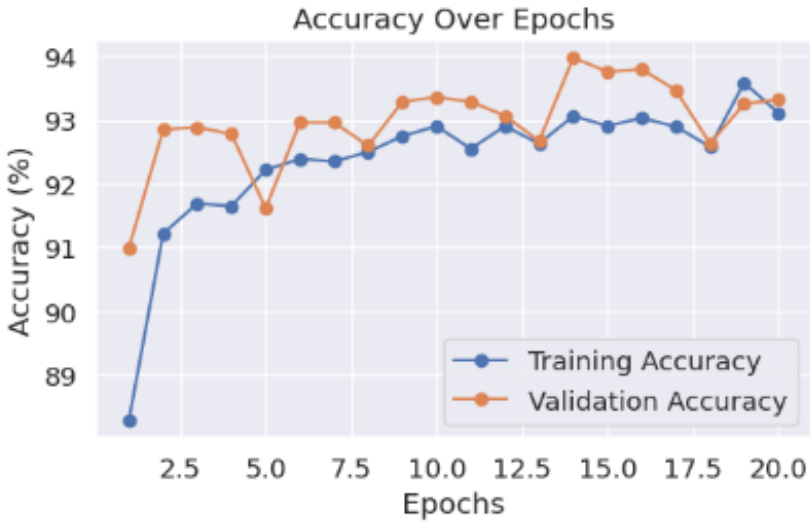


Figure 5.2.15 Graph of Validation accuracy and Train accuracy

5.3 Comparison between Models

The overall of the three model is shown in Figure 5.2.16.

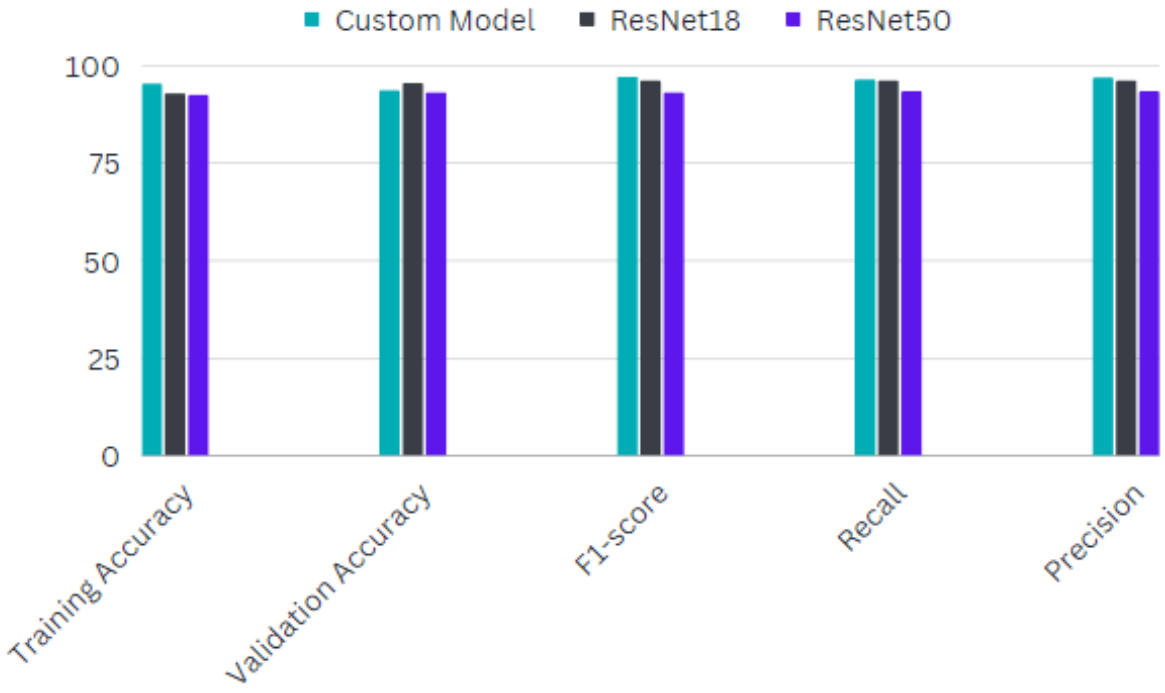


Figure 5.2.16 Performance matrices of various models

In conclusion, the malaria parasite detection task was addressed using three different models: a custom model, ResNet-18, and ResNet-50. Various evaluation metrics were employed to assess the performance of each model, ultimately leading to the selection of the model that exhibited the best overall performance. These findings clearly show that the custom model beat ResNet-18 and ResNet-50 in terms of accuracy during training,

validation, F1-score, recall, and precision. Notably, the custom model achieved the highest validation accuracy, demonstrating its efficacy in detecting malaria parasites. ResNet-18 also demonstrated strong performance, particularly in terms of validation accuracy and F1-score. On the other hand, while ResNet-50 yielded reasonable results, it fell slightly behind other two models in terms of accuracy and overall performance.

In practice, the choice of the best model may depend on various factors, including computational resources, training time, and deployment requirements. However, based on the evaluation metrics, the custom model stands out as the top performer in this malaria parasite detection task.

5.4 Example Output

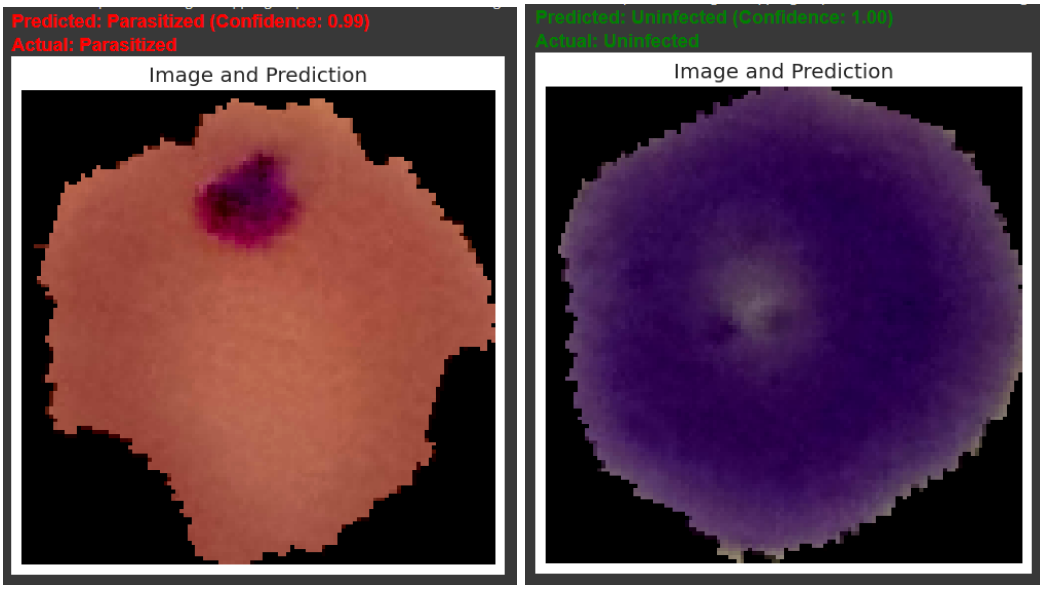


Figure 5.2.17 Images of Correctly Predicted Cell

Images of cells from the dataset that were correctly classified by the deep learning model. These cells were predicted as parasite or uninfected by the model, and these predictions matched the actual labels or ground truth.

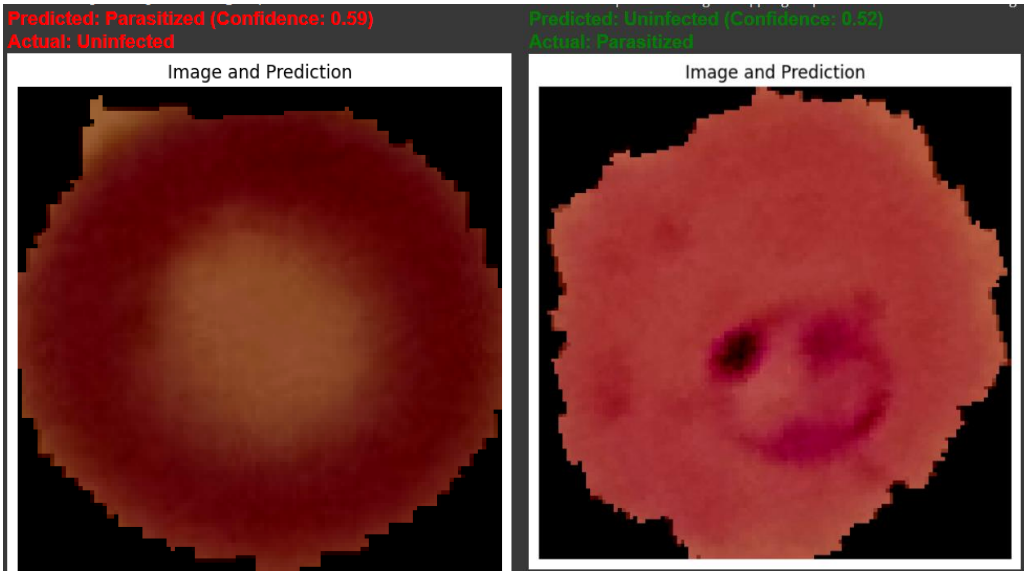


Figure 5.2.18 Images of Wrongly Predicted Cell

These cells were predicted as one class by the model, but the actual labels or ground truth indicated a different class. These are often referred to as "false positives" or "false negatives" depending on the specific misclassification.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

In conclusion, malaria is a major global source of morbidity and mortality, and early diagnosis is essential for effective treatment and prevention. The conventional diagnostic techniques are frequently labor-intensive and prone to mistakes. Traditional diagnostic methods have limitations and can lead to diagnostic errors. The objective of this project was to utilize deep learning methods to tackle the challenges associated with detecting malaria parasites in blood smear images. The motivation behind this project was to develop a deep learning system capable of accurately classifying and detecting malaria parasites in blood smear images. Such a system has the potential to enhance diagnostic accuracy, reduce the workload of healthcare professionals, and expedite the diagnostic process. By utilizing pre-trained models like ResNet, transfer learning is employed to enhance the performance of the model and minimize training time. Through the pre-trained models on a dataset comprising human blood smear images, the model can acquire the ability to effectively detect the existence of malaria parasites with high accuracy. The novelty of this project is the use of transfer learning to optimize the deep learning model for malaria parasite detection. A novel idea is the creation of an automatic malaria diagnosis system that can enhance the precision and swiftness of diagnostic procedures, especially in areas where resources are scarce. The successful implementation of this project holds the promise of earlier detection of malaria, potentially saving lives and reducing the burden of this disease.

6.2 Recommendations

In considering future research directions, it is essential to expand the horizons of these models by training and evaluating them on diverse datasets. Additionally, there is room for enhancing model performance through the incorporation of advanced deep learning techniques like ensemble classifiers and decision algorithms. A primary objective for the future is to develop a more resilient and efficient model that can deliver high accuracy across various test datasets while reducing the time required for training.

REFERENCES

[1] "Frequently Asked Questions (FAQs) about Malaria", *Centers for Disease control and Prevention*, 2022. [Online]. Available: https://www.cdc.gov/malaria/about/faqs.html

[2] “Fact Sheet about Malaria.” 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/malaria

[3] "Malaria Diagnostic Tests", *Centers for Disease Control and Prevention*, 2022. [Online]. Available: https://www.cdc.gov/malaria/diagnosis_treatment/diagnostic_tools.html#:~:text=A%20Rapid%20Diagnostic%20Test%20(RDT,available%20in%20the%20United%20States.

[4] C. K. Tan, C. M. Goh, S. A. Zikri Bin Sayed Aluwee, S. W. Khor and M. T. Chai, "Malaria Parasite Detection using Residual Attention U-Net," 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2021, pp. 122-127, doi: 10.1109/ICSIPA52582.2021.9576814.

[5] G. Shekar, S. Revathy and E. K. Goud, "Malaria Detection using Deep Learning," 2020 4th International Conference on Trends in Electronics and Informatics (zCOEI)(48184), 2020, pp. 746-750, doi: 10.1109/ICOEI48184.2020.9143023.

[6] A. M. Joshi, A. K. Das and S. Dhal, "Deep Learning Based Approach For Malaria Detection in Blood Cell Images," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 241-246, doi: 10.1109/TENCON50793.2020.9293753.

[7] Reddy, A. Sai Bharadwaj and D. Sujitha Juliet. “Transfer Learning with ResNet-50 for Malaria Cell-Image Classification.” *2019 International Conference on Communication and Signal Processing (ICCSP)* (2019): 0945-0949.

[8] Rajaraman S, Antani SK, Poostchi M, et al. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. Peerj. 2018 ;6:e4568. DOI: 10.7717/peerj.4568. PMID: 29682411; PMCID: PMC5907772.

[9] Arunava, “Malaria Cell Images Dataset,” Kaggle, 05-Dec-2018. [Online]. Available: https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria.

[10] “*What Is* Transfer Learning?” [Examples & Newbie-Friendly Guide]. [Online]. Available: https://www.v7labs.com/blog/transfer-learningguide#:~:text=In%20other%20words%2C%20transfer%20learning,when%20modeling%20the%20second%20t.

[11] “What is transfer learning? exploring the popular deep learning approach.,” Built In. [Online]. Available: https://builtin.com/data-science/transfer-learning

[12] S. Mukherjee, “The annotated resnet-50,” *Medium*, 18-Aug-2022. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[13] “Deep Network designer,” *ResNet-50 convolutional neural network* - MATLAB. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/resnet50.html

[14] “ResNet-50: The basics and a quick tutorial,” *Datagen*, 13-Apr-2023. [Online]. Available: https://datagen.tech/guides/computer-vision/resnet-50/.

[15] “1.5. stochastic gradient descent,” *scikit.* [Online]. Available: https://scikit-learn.org/stable/modules/sgd.html#:~:text=Stochastic%20gradient%20descent%20is%20an,training%20example%20at%20a%20time.

Appendix

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 2**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Review the code done in FYP I
Getting more study materials for further research

2. WORK TO BE DONE

Note down where to take for further improvement.

3. PROBLEMS ENCOUNTERED

No problems encountered.

4. SELF EVALUATION OF THE PROGRESS

Not being productive enough

Sayed

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 4**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Finished researched on ResNet architecture.

2. WORK TO BE DONE

Get dataset and codes ready for testing purpose.

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

The fyp schedule need to be follow so the project can be completed in time.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 6**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Test run on training phase.

2. WORK TO BE DONE

Start training model.

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Could manage time well for incoming midterms meanwhile doing researching for the project.

Sayed

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 8**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Complete training

2. WORK TO BE DONE

Continue experiment on the model

3. PROBLEMS ENCOUNTERED

Limited GPU usages from Colab which delayed the progress.

4. SELF EVALUATION OF THE PROGRESS

Need to do faster

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 10**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Complete refining chapter 1-2

2. WORK TO BE DONE

Compare the results

3. PROBLEMS ENCOUNTERED

Limited GPU usages from Colab which delayed the progress.

4. SELF EVALUATION OF THE PROGRESS

Need to do faster.

Sayed

Supervisor's signature

Student's signature

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 2, Year 3	**Study week no.: 12**
Student Name & ID: Teow Yi Jia, 20ACB00323	
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee	
Project Title: Malaria Parasite Detection from human blood smear images using deep learning techniques	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Done testing.

2. WORK TO BE DONE

Complete the report

3. PROBLEMS ENCOUNTERED

No problem

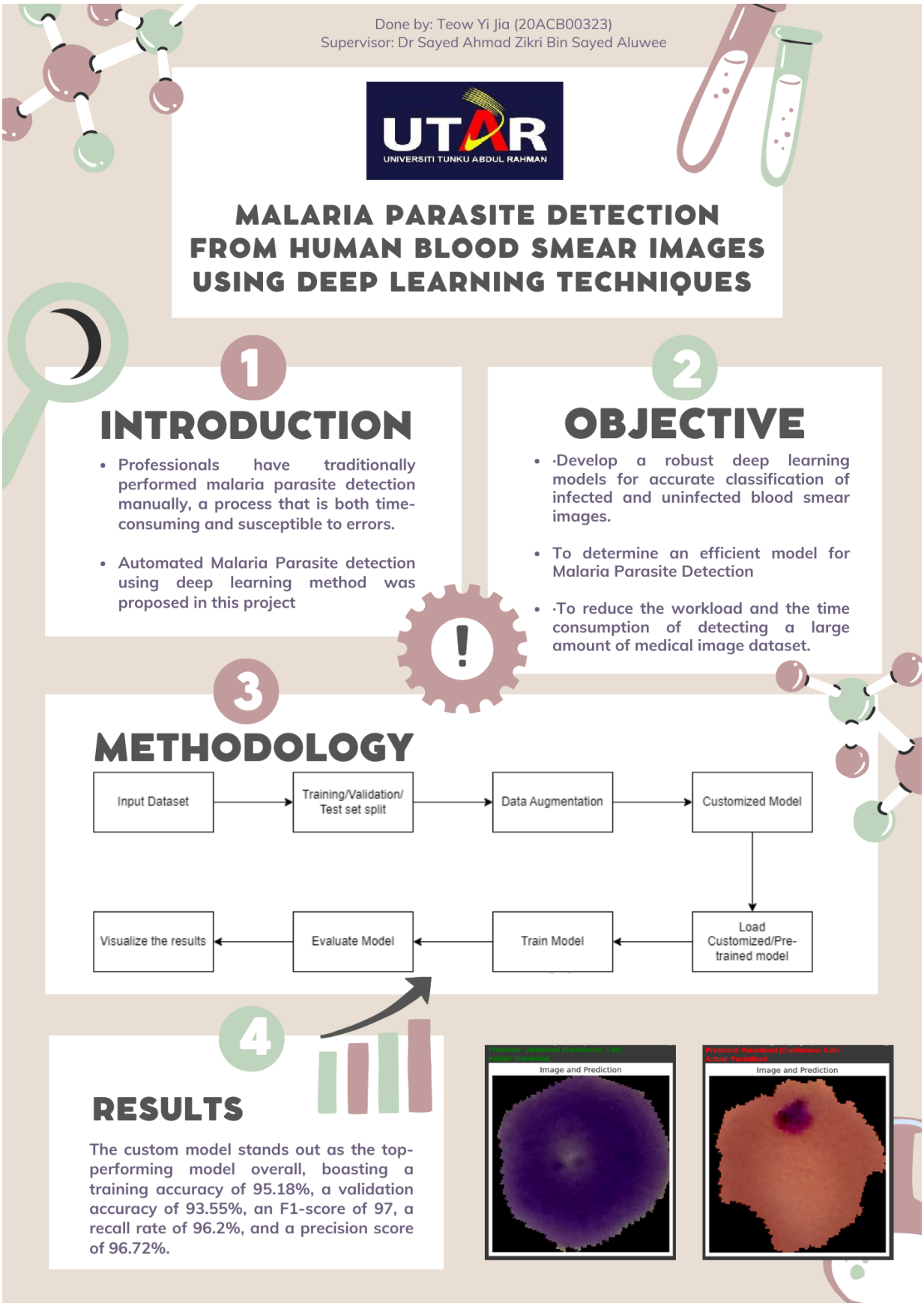
4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

POSTER


Done by: Teow Yi Jia (20ACB00323)
Supervisor: Dr Sayed Ahmad Zikri Bin Sayed Aluwee
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
MALARIA PARASITE DETECTION FROM HUMAN BLOOD SMEAR IMAGES USING DEEP LEARNING TECHNIQUES
1
INTRODUCTION
Professionals have traditionally performed malaria parasite detection manually, a process that is both time-consuming and susceptible to errors.
Automated Malaria Parasite detection using deep learning method was proposed in this project
2
OBJECTIVE
·Develop a robust deep learning models for accurate classification of infected and uninfected blood smear images.
To determine an efficient model for Malaria Parasite Detection
·To reduce the workload and the time consumption of detecting a large amount of medical image dataset.
3
METHODOLOGY
Input Dataset
Training/Validation/ Test set split
Data Augmentation
Customized Model
Load Customized/Pre-trained model
Train Model
Evaluate Model
Visualize the results
4
RESULTS
The custom model stands out as the top-performing model overall, boasting a training accuracy of 95.18%, a validation accuracy of 93.55%, an F1-score of 97, a recall rate of 96.2%, and a precision score of 96.72%.
Image and Prediction
Image and Prediction

PLAGIARISM CHECK RESULT

fyp2

ORIGINALITY REPORT

15%	8%	9%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Amogh Manoj Joshi, Ananta Kumar Das, Subhasish Dhal. "Deep Learning Based Approach For Malaria Detection in Blood Cell Images", 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020 Publication	1%
2	Submitted to Liverpool John Moores University Student Paper	1%
3	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
4	Chiang Kang Tan, Chuan Meng Goh, Sayed A. Zikri Bin Sayed Aluwee, Siak Wang Khor, Meei Tyng Chai. "Malaria Parasite Detection using Residual Attention U-Net", 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2021 Publication	1%
5	www.mdpi.com Internet Source	1%

6	A. Sai Bharadwaj Reddy, D. Sujitha Juliet. "Transfer Learning with ResNet-50 for Malaria Cell-Image Classification", 2019 International Conference on Communication and Signal Processing (ICCSP), 2019 Publication	1 %
7	Submitted to National College of Ireland Student Paper	1 %
8	link.springer.com Internet Source	<1 %
9	www.ijraset.com Internet Source	<1 %
10	section.iaesonline.com Internet Source	<1 %
11	munin.uit.no Internet Source	<1 %
12	"Machine Learning in Medical Imaging", Springer Science and Business Media LLC, 2019 Publication	<1 %
13	"Machine Vision and Augmented Intelligence—Theory and Applications", Springer Science and Business Media LLC, 2021 Publication	<1 %
14	P.O. Abas Sunarya, Rina Refianti, Achmad Benny, Wiranti Octaviani. "Comparison of	<1 %

	Accuracy between Convolutional Neural Networks and Naïve Bayes Classifiers in Sentiment Analysis on Twitter", International Journal of Advanced Computer Science and Applications, 2019 Publication	
15	Submitted to University of Surrey Student Paper	<1 %
16	Submitted to Brunel University Student Paper	<1 %
17	Mandar Diwakar, Brijendra Gupta. "The robust feature extraction of audio signal by using VGGish model", Research Square Platform LLC, 2023 Publication	<1 %
18	Pengli Lu, Yue Luo, Teng Zhang. "A critical node identification approach for complex networks combining self-attention and ResNet", International Journal of Modern Physics C, 2023 Publication	<1 %
19	peerj.com Internet Source	<1 %
20	Submitted to Istanbul Aydin University Student Paper	<1 %
21	eprints.utar.edu.my Internet Source	<1 %

Plagiarism Checking Result

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Teow Yi Jia
ID Number(s)	20ACB00323
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	Malaria parasite detection from human blood smear images using deep learning techniques

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 15 % **Similarity by source** Internet Sources: 8 % Publications: 9 % Student Papers: 7 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:
(i) Overall similarity index is 20% and below, and
(ii) Matching of individual sources listed must be less than 3% each, and
(iii) Matching texts in continuous block must not exceed 8 words
Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr Sayed Ahmad Zikri Bin Sayed Aluwee

Date: 14/9/2023

Signature of Co-Supervisor

Name: Dr Goh Chuan Meng

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB00323
Student Name	Teow Yi Jia
Supervisor Name	Dr Sayed Ahmad Zikri Bin Sayed Aluwee

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

_____

(Signature of Student)
Date: 15/9/2023