

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Machine learning can be viewed as an application or a subfield of artificial intelligence (AI) that provides machines and systems the ability to automatically learn and improve from experience without being explicitly programmed or guided. Machine learning directly focuses on the development of computer programs that can process data and use it learn for themselves. The process of learning starts with the analysis of data, such as examples, direct experience, or instructions, in order to look for patterns in data and make better and accurate decisions in the future based on the examples that were provided initially. The primary aim is to allow the computers learn automatically without human intervention and adjust actions accordingly. Machine learning enables analysis of large amounts of data. While it usually delivers faster and more accurate results in order to identify profitable opportunities or dangerous risks, it also needs additional time and resources to train it properly and effectively. Combining machine learning with AI and cognitive science like computer vision can make it even more effective in processing large volumes of information. Machine learning algorithms can be categorized as supervised or unsupervised (Marco *et al.*, 2016).

Supervised machine learning algorithms are able to apply what they have learned in the past to new data using labeled examples to predict future events. Starting from the observations gathered from an already known training dataset, the learning algorithm produces an inferred function to make accurate predictions about the output values. The system is then able to provide targets for any other new input after sufficient training. The learning algorithm also has the ability to compare its output with the correct, intended output and find errors or anomalies in order to modify the model accordingly. In contrast, unsupervised machine learning algorithms are used in scenarios where the information used to train is neither classified nor labeled. Unsupervised learning focuses on how systems can infer a function to describe a hidden structure from unlabeled datasets. The system does not seek out the right output, but it explores the entire data and can draw inferences from datasets to describe hidden structures from unlabeled data. It is best to employ supervised machine learning in malaria diagnosis since we are dealing with classification using extracted details from images as a learning tool. We are working based on a system in which both input and

desired output data are provided. Input and output data are labelled accordingly for classification to provide a learning basis for future data processing. Supervised machine learning systems provide the learning algorithms with known quantities or inferred functions to support future prediction. Chat robots, self-driving cars, facial recognition software, expert systems and bots are among the systems that employ either supervised or unsupervised learning. Supervised learning systems are largely associated with retrieval-based Artificial Intelligence but they are also capable of using a generative learning model (Rouse 2018). Training data for supervised learning algorithms consists of a set of labeled examples with paired input data and desired output (which is also referred to as the supervisory signal)

In supervised learning for image processing, for example, an AI system might be trained with labeled pictures of vehicles in categories such as cars and trucks. After a sufficient amount of observation, the system should be able to differentiate between and categorize unlabeled images, at that time then the training can be said to be complete. For this case, using supervised machine learning and image analysis which is implemented using computer vision we deal with how computers can be made to gain high-level understanding from digital images. Computer vision includes methods for acquiring, processing, analyzing and understanding features from digital images in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context entails the transformation of digital/visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit desired action. As a scientific discipline, computer vision is centered on the theory behind artificial systems that extract information from images. The image data can be in many forms, such as video feeds, views from multiple cameras, or multi-dimensional data from a camera. This project focuses on machine learning approaches to pattern recognition. Pattern recognition is categorized according to the type of learning algorithm employed to generate the output value. Supervised learning expects that a set of training data (the training dataset) has been provided, consisting of a set of instances that have been accurately labeled by hand with the correct output. Training patterns are the goals of the training process and not to be confused with the training set. A learning procedure then generates a model that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data (Ponce 2005).

1.2 Statement of the Problem

Malaria is a life threatening disease so rapid, accurate diagnosis is required to control the disease. The detection of Malaria parasites in blood samples is done by pathologists manually using microscopes. So, the chances of false detection due to human error still exists, which in turn can result into fatal conditions. The accurate and timely diagnosis of malaria infection is essential to detect, control and cure the disease.

1.3 Aim and Objectives of the Study

This major aim and objectives of this study include:

- Exploring the possibility of the computerized diagnosis of malaria.
- Implementing a convolutional neural network using supervised machine learning to reliably detect the presence of malaria parasites in images of a patient's blood sample.

1.4 Significance of the Study

If this program is implemented and deployed, it will be very useful in many areas such as:

- a. It will help to aid medical doctors in rural areas with limited or no access to pathologists.
- b. It can be used in battling malaria outbreaks where hundreds of blood samples may need to be tested quickly and accurately.
- c. It can support academic development in the study of medicine, pathology and microbiology
- d. It can be useful in many hospitals, both private and government.
- e. It can also be used in the laboratory for quick research work.

1.5 Scope of the Project

The scope of this work includes the following:

1. A set of training sample images used to indicate patterns of malaria infections.
2. A system for malaria diagnosis meant to serve medical doctors, pathologists and other medical personnel.
3. A parasite detection algorithm to detect parasites of the plasmodium species only.
4. Exploring the power of MATLAB in machine learning, classification and data handling.
5. To implement a dynamic system for image analysis with high speed and accuracy.

1.6 Limitations of the Study

The major constraint faced during the implementation of this work was data gathering and access to enough infected and non-infected malaria blood samples from local hospitals. Another limitation was distorted images and inaccurate labelling of the images. This is among other frustrations such as program failures during modular construction stages.

1.7 Definition of Terms

Accuracy	Accuracy is the measure of how good a classification model is. It is given by the number of correctly classified examples divided by the number of classified examples.
Artificial intelligence	This deals with efforts to make computers to think and do things intelligently. Artificial intelligence is a part of computer science that deals with designing intelligent computer systems, i.e., system that exhibit the characteristics we associate with intelligence in human behavior.
Class	A class is a group to which an example can belong. A labeled example consists of a feature vector and a reference to the class it belongs to.

Classification	Classification is a problem of assigning a label to an example.
Computer Vision	Computer Vision (CV) is a field of Artificial Intelligence concerned with providing tools for analysis and high-level understanding of image and video data.
Data	Any collection of information converted into a digital form.
Decision Boundary	In a classification problem with two or more classes, a decision boundary is a hypersurface that partitions the underlying vector space into two or more regions, one for each class.
Feature	A feature is an attribute of an example, usually a part of a feature vector. It can be numerical or categorical. If an example is a person, it can have the following features: height (numerical), weight (numerical), race (categorical), etc.
Image classification	Image classification is a CV task of teaching a model to recognize what is on a given image.
Image segmentation	Image segmentation is a CV task where one trains a model to annotate each pixel with a class from a predefined set to which a given pixel most probably belongs.
Machine Learning	Machine Learning refers to the techniques involved in dealing with vast data in the most intelligent fashion (by developing algorithms) to derive actionable insights
Model	A model, also known as a statistical model, is the result of a machine learning algorithm applied to the training data. Model is often a parametrized mathematical formula, where parameters are learned by the machine learning algorithm.

Neural Networks	Neural Networks is a very wide family of Machine Learning models. The main idea behind them is to mimic the behavior of a human brain when processing data.
Object detection	Object detection is a CV task of teaching the model to detect an instance of an object from a set of predefined categories by providing a bounding box around each instance of a given class.
Pathologist	A scientist who studies the causes and effects of diseases, especially one who examines laboratory samples of body tissue for diagnostic or forensic purposes.
Pattern Recognition	Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data. Classification is an example of pattern recognition wherein each input value is assigned one of a given set of classes.
Supervised learning	Supervised learning is a family of Machine Learning models that teach themselves by example. This means that data for a supervised ML task needs to be labeled (assigned the right, ground-truth class).
Test data	It is used once the final model is chosen to simulate the model's behavior on a completely unseen data, i.e. data points that weren't used in building models or even in deciding which model to choose.
Training data	Training data is used to train a model. It means that ML model sees that data and learns to detect patterns or determine which features are most important during prediction.
Unsupervised learning	Contrary to Supervised Learning, Unsupervised Learning models teach themselves by observation. The data provided to that kind of algorithms is unlabeled (there is no ground truth value given to the algorithm).
Validation data	Validation data is used for tuning model parameters and comparing different models in order to determine the best ones. The validation data should be different from the training data, and should not be used in the training phase.

CHAPTER TWO

LITERATURE REVIEW

2.1 History of Machine Learning

Right from the onset of evolution, humans have been using many types of tools to accomplish various tasks. The intelligence and creativity of the human brain led to the invention of many different machines. These machines helped make the life of humans easy by enabling people to meet various life needs, including traveling, industries, constructions, and computing. Despite the rapid and successful developments in the machine industry, intelligence has remained the fundamental difference between humans and machines in performing tasks. A human makes use of his or her five senses to gather information or stimuli from the surrounding atmosphere; the human brain works to analyze that information and takes suitable decisions accordingly. Machines, on the other hand, are not intelligent by nature. A machine lacks the ability to analyze data and take decisions. For example, a machine is not capable or expected to understand the story of Harry Potter or a James Bond movie, jump over a hole in the street, or interact with other machines through a common language. The era of intelligent machines began in the mid-twentieth century when Alan Turing thought whether it is possible for machines to think. Ever since then, the artificial intelligence (AI) branch of computer science has developed rapidly over the past few decades. Humans have had the dream and the zeal to create machines that have the same level of intelligence as humans. Many science fiction movies have expressed these dreams about Artificial Intelligence; The Matrix; The Terminator; I, Robot; and Star Wars.

According to (Tandara and Barwick, 2014), The history of AI started in the year 1943 when Warren McCulloch and Walter Pitts introduced the first neural network model. Alan Turing introduced the next noticeable work in the development of the AI in 1950 when he asked his famous question: can machines think? He introduced the B-type neural networks and also the concept of test of intelligence. In 1955, Oliver Selfridge proposed the use of computers for pattern recognition. In 1956, John McCarthy, Marvin Minsky, Nathan Rochester of IBM, and Claude Shannon organized the first summer AI conference at Dartmouth College, the United States. In the second Dartmouth conference, the term artificial intelligence was used for the first time. The term cognitive science originated in 1956, during a symposium in information science at the MIT, the United States (Bashier 2016).

Rosenblatt invented the first perceptron in 1957. Then in 1959, John McCarthy invented the LISP programming language. David Hubel and Torsten Wiesel proposed the use of neural networks for the computer vision in 1962. Joseph Weizenbaum developed the first expert system which was called Eliza that could diagnose a disease based on its symptoms. The National Research Council (NRC) of the United States founded the Automatic Language Processing Advisory Committee (ALPAC) in 1964 to advance the research in the natural language processing. But after many years, the two organizations terminated the research because of the high expenses and low progress. Marvin Minsky and Seymour Papert published their book Perceptrons in 1969, in which they demonstrated the limitations of neural networks. As a result, organizations stopped funding research on neural networks. The period from 1969 to 1979 witnessed a growth in the research of knowledge based systems. The programs Dendral and Mycin are examples of this research. In 1979, Paul Werbos proposed the first efficient neural network model with backpropagation. However, in 1986, David Rumelhart, Geoffrey Hinton, and Ronald Williams discovered a method that allowed a network to learn to discriminate between nonlinear separable classes, and they named it back propagation (Weizenbaum 2017).

In 1987, Terrence Sejnowski and Charles Rosenberg developed an artificial neural network NETTalk for speech recognition. In 1987, John H. Holland and Arthur W. Burks invented an adapted computing system that is capable of learning. In fact, the development of the theory and application of genetic algorithms was inspired by the book Adaptation in Neural and Artificial Systems, written by Holland in 1975. In 1989, Dean Pomerleau proposed ALVINN (autonomous land vehicle in a neural network), which was a three-layer neural network designed for the task of the road following (Gary 2011). In the year 1997, the Deep Blue chess machine, designed by IBM, defeated Garry Kasparov, the world chess champion. In 2011, Watson, a computer developed by IBM, defeated Brad Rutter and Ken Jennings, the champions of the television game show Jeopardy! The period from 1997 to the present witnessed rapid developments in reinforcement learning, natural language processing, emotional understanding, computer vision, and computer hearing. The current research in machine learning focuses on computer vision, hearing, natural languages processing, image processing and pattern recognition, cognitive computing, knowledge representation, and so on. These research trends aim to provide machines with the abilities of gathering data through senses similar to the human senses and then processing the gathered data by using the computational intelligence tools and machine learning methods to conduct predictions

and making decisions at the same level as humans. The term machine learning means to enable machines to learn without programming them explicitly.

2.2 The Study of Machine Learning

According to Durant (1929), Learning is a very personalized phenomenon for us. Will Durant in his famous book, The Pleasures of Philosophy, wondered in the chapter titled “Is Man a Machine?” when he wrote such classical lines:

“Here is a child; ... See it raising itself for the first time, fearfully and bravely, to a vertical dignity; why should it long so to stand and walk? Why should it tremble with perpetual curiosity, with perilous and insatiable ambition, touching and tasting, watching and listening, manipulating and experimenting, observing and pondering, growing—till it weighs the earth and charts and measures the stars?”

Nevertheless, learning is not limited to humans only. Even the simplest of species such as amoeba and paramecium exhibit this phenomenon. Plants also show intelligent behavior. Only nonliving things are the natural stuffs that are not involved in learning. Hence, it seems that living and learning go together. In nature-made nonliving things, there is hardly anything to learn. Can we introduce learning in human-made nonliving things that are called machines? Enabling a machine capable of learning like humans is a dream, the fulfilment of which can lead us to having deterministic machines with freedom (or illusion of freedom in a sense). During that time, we will be able to happily boast that our humanoids resemble the image and likeliness of humans in the guise of machines. Machines are by nature not intelligent. Initially, machines were designed to perform specific tasks, such as running on the railway, controlling the traffic, digging deep holes, traveling into the space, and shooting at moving objects. Machines do their tasks much faster with a higher level of precision compared to humans (Khan *et al.*, 2016). They have made our lives easy and smooth. The fundamental difference between humans and machines in performing their work is intelligence. The human brain receives data gathered by the five senses: vision, hearing, smell, taste, and tactility. These gathered data are sent to the human brain via the neural system for perception and taking action. In the perception process, the data is organized, recognized by comparing it to previous experiences that were stored in the memory, and interpreted. Accordingly,

the brain takes the decision and directs the body parts to react against that action. At the end of the experience, it might be stored in the memory for future benefits. A machine cannot deal with the gathered data in an intelligent way. It does not have the ability to analyze data for classification, benefit from previous experiences, and store the new experiences to the memory units; that is, machines do not learn from experience. Although machines are expected to do mechanical jobs much faster than humans, it is not expected from a machine to: understand the play Romeo and Juliet, jump over a hole in the street, form friendships, interact with other machines through a common language, recognize dangers and the ways to avoid them, decide about a disease from its symptoms and laboratory tests, recognize the face of the criminal, and so on. The challenge is to make dumb machines learn to cope correctly with such situations. Because machines have been originally created to help humans in their daily lives, it is necessary for the machines to think, understand to solve problems, and take suitable decisions akin to humans.

Machine learning is a branch of artificial intelligence that aims to equip machines with the ability to perform their jobs skillfully by using intelligent software. The statistical learning methods forms the backbone of intelligent software that is used to develop intelligent machines. Because machine learning algorithms need data to learn, the discipline must have connection with the discipline of database. Similarly, there are familiar terms such as Knowledge Discovery from Data (KDD), data mining, and pattern recognition (Asahd and Mohssen, 2018).

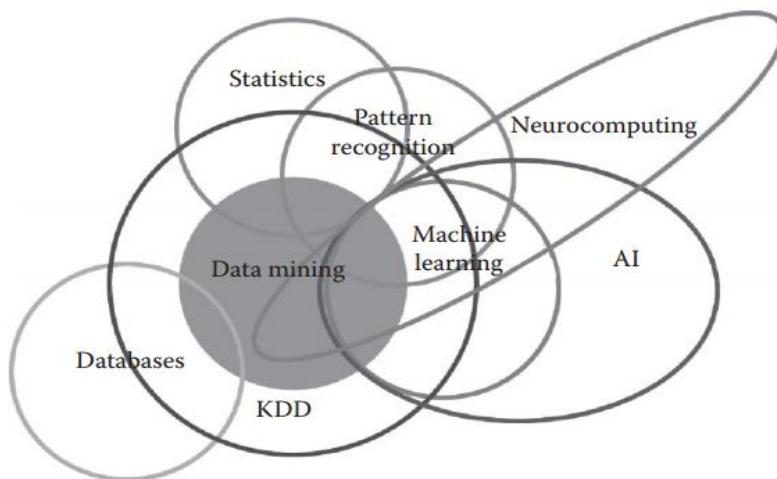


Fig 2.1 Machine Learning: Where Several Disciplines meet.

Source: <http://blogs.sas.com/content/subconsciousmusings/2014/08/22/looking-backwards-lookingforwards-sas-data-mining-and-machine-learning/2014>.

Machine Learning is an intersection of Computer Science and Statistics. We might say the outright question of Computer Science is ‘How can we build intelligent machines that can solve problems, and which problems are inherently amenable/non-amenable?’ The question that particularly defines Statistics is ‘What can be inferred from a given data plus a set of modeling assumptions and with what accuracy?’ Machine Learning integrates additional questions about what computational architectures and algorithms can be used to most effectively capture, store, index, retrieve and merge these data, how multiple learning subtasks can be orchestrated in a larger system, and questions of computational tractability (Junaid 2011).

There are some tasks that humans perform effortlessly or with some efforts, but we are unable to explain how we perform them. For example, we can recognize the speech of our friends without much difficulty. If we are asked how we recognize the voices, the answer is very difficult for us to explain. Because of the lack of understanding of such phenomenon (speech recognition in this case), we cannot craft algorithms for such scenarios. Machine learning algorithms are helpful in bridging this gap of understanding. The idea is very simple. We are not targeting to understand the underlying processes that help us learn.

We write computer programs that enables machines learn and enables them to perform tasks, such as prediction. The goal of learning is to construct a model that takes the input and produces the desired result. Sometimes, we can understand the model, whereas, at other times, it can also be like a black box for us, the working of which cannot be intuitively explained (Asahd and Mohssen, 2018). The model can be considered as an approximation of the process we want machines to mimic. In such a situation, it is possible that we obtain errors for some input, but most of the time, the model provides correct answers. Hence, another measure of performance (besides performance of metrics of speed and memory usage) of a machine learning algorithm will be the accuracy of results.

2.3 Machine Learning Techniques and Required Data

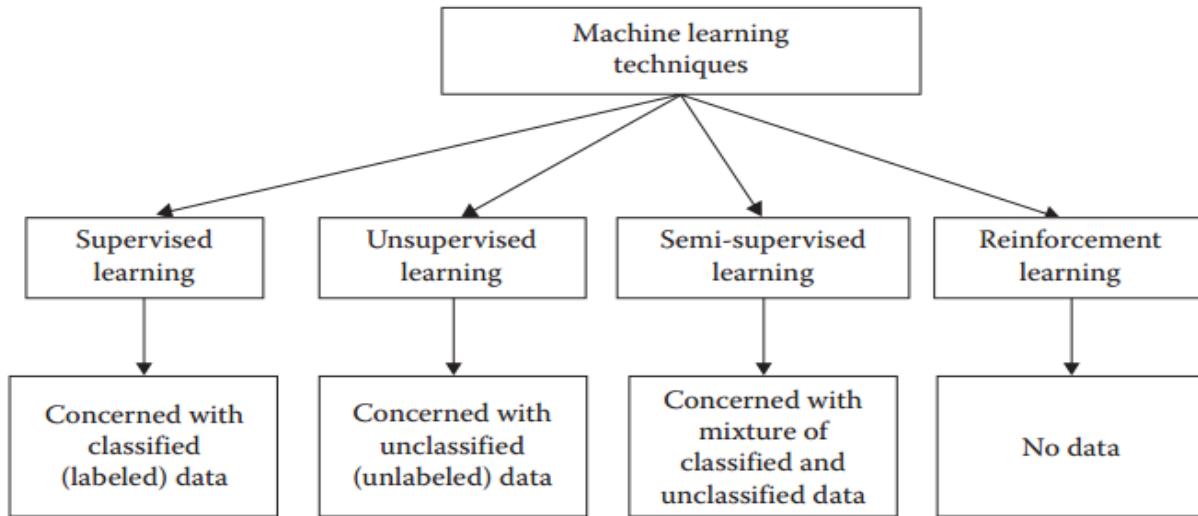


Fig 2.2 Different machine learning techniques and their required data.

Source: <https://www.aiworld/home/ml>

There are four general machine learning methods: supervised, unsupervised, semi-supervised, and reinforcement learning methods. The objectives of machine learning techniques are to enable machines to make predictions, perform clustering, extract association rules, or make decisions from a given dataset.

In supervised learning, the target is to infer a function or mapping from training data that is labeled. The training data consist of input vector X and output vector Y of labels or tags. A label or tag from vector Y is the explanation of its respective input example from input vector X. Together they form a training example. In other words, training data comprises training examples. If the labeling does not exist for input vector X, then X is unlabeled data. Why such learning is called supervised learning? The output vector Y consists of labels for each training example present in the training data. These labels for output vector are provided by the supervisor. Often, these supervisors are humans, but machines can also be used for such labeling. Human judgments are more expensive than machines, but the higher error rates in data labeled by machines suggest superiority of human judgment. The manually labeled data is a precious and reliable resource for supervised learning. However, in some cases, machines can be used for reliable labeling.

2.4 Supervised Machine Learning

Supervised Learning are the ones that involve direct supervision of the operation. In this case, the programmer labels sample data and sets strict parameters upon which the algorithm operates. It is a spoon-fed version of machine learning: you select what kind of information output (samples) to “feed” the algorithm and what kind of results it is desired (for example “yes/no” or “true/false”). From the machine’s point of view, this process becomes more or less like a game or a “connect the dots” routine. The primary purpose of supervised learning algorithms is to scale the scope of data and to make predictions of unavailable, future or unseen data based on labeled sample data (Bilyk,2012)

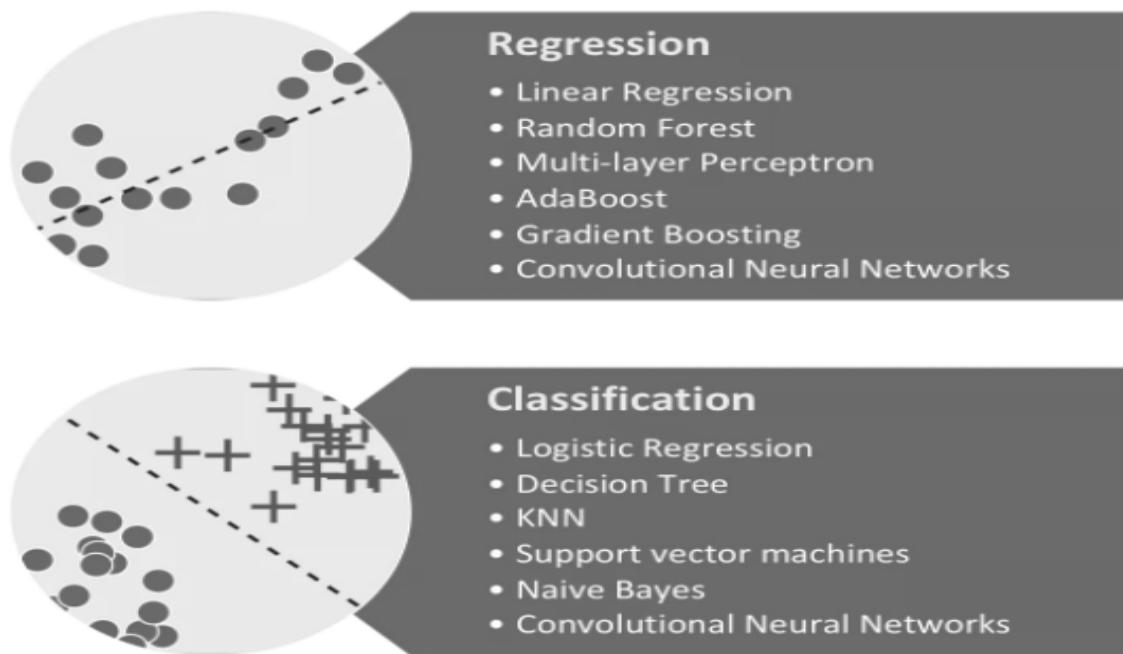


Fig 2.3 Machine Learning Algorithms
Source: <https://vinodsblog.com/2018/03/26/machine-learning-introduction-to-its-algorithms-mlalgos/>

- **Classification:** It predicts the category the data belongs to e.g.: Spam Detection, Churn Prediction, Sentiment Analysis, Dog Breed Detection, etc.
- **Regression:** It predicts a numerical value based on previous observed data. e.g.: House Price Prediction, Stock Price Prediction, Height-Weight Prediction, etc.

The process of the supervised machine learning algorithm learning from the training dataset can be thought of as a teacher overseeing/supervising the learning process. We already know the correct answers so the algorithm iteratively makes predictions on the training data it has been given and is corrected by the teacher. Learning can be stopped when the algorithm performs satisfactorily and achieves an acceptable level of performance. In Supervised Learning, the algorithms learn from already labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data. Supervised machine learning is a type of system in which both input and desired output data are provided. Input and output data are well labeled for classification to provide a learning basis for future data processing. Supervised Learning is divided into two categories i.e. Classification & Regression.

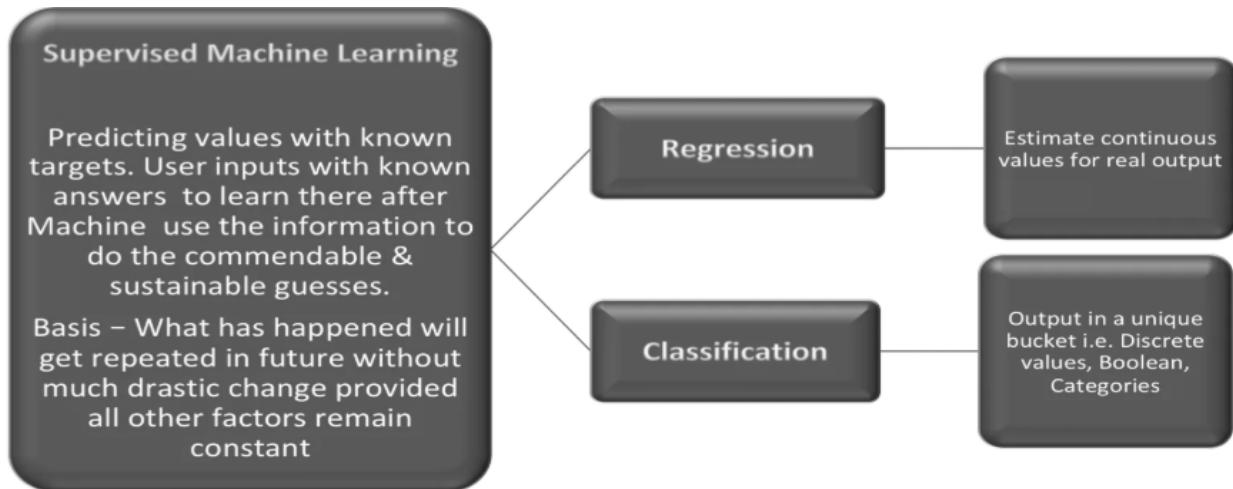


Fig 2.4 Supervised machine learning and its branches

Source: <https://vinodsblog.com/2018/04/02/supervised-machine-learning-insider-scoop-for-labeled-data/>

2.4.1 The Supervised Machine Learning Process

While there are numerous machine learning algorithms for supervised learning, most of them employ the same basic work flow for obtaining a predictor model. The accuracy of the supervised learning process is determined by the number of correct classification divided by the total number of test cases. This equation clearly shows accuracy will be more close to perfection when we have when the difference between “number of correct classification” and “number of test cases” is minimal. The process for Supervised Machine Learning is basically a two-way process as below.

- Learning – Learn a model using the training data or train model using training data.
- Testing – Test the model using unseen test data to assess the model accuracy

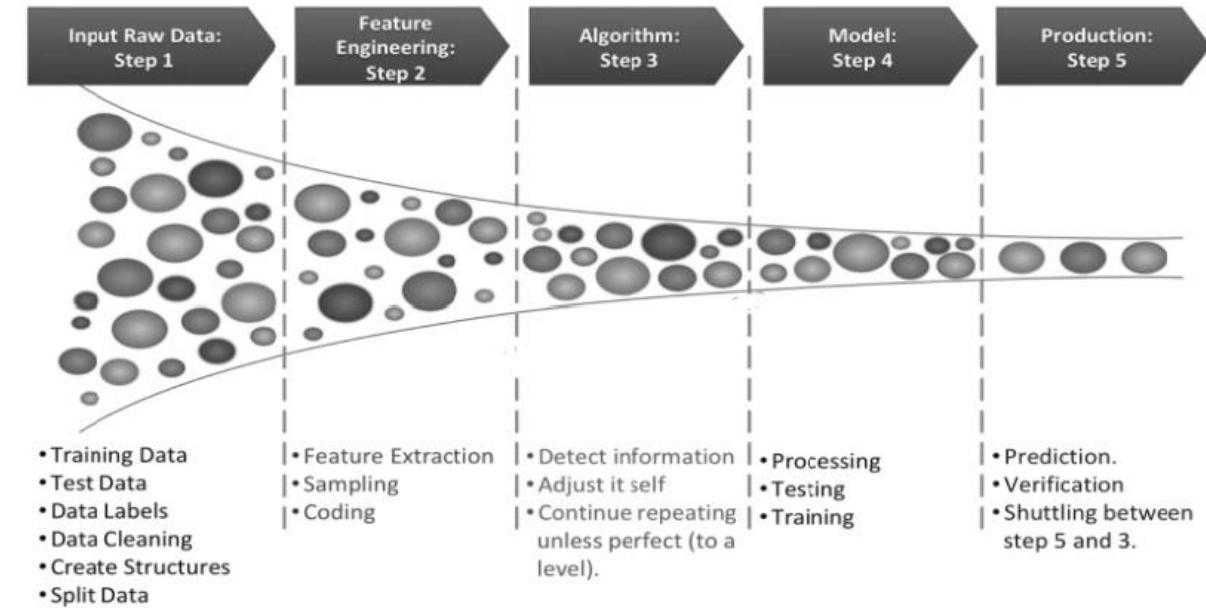


Fig 2.5 Supervised machine learning process

Source: <https://towardsdatascience.com/supervised-machine-learning-classification->

1. Prepare data
2. Choose an algorithm
3. Fit a model
4. Choose a validation method
5. Examine fit and update until satisfied
6. Use the fitted model for predictions

The detailed steps for supervised learning processes are included but not limited with pointers as above.

2.5 Convolutional Neural Networks in Image Analysis

Convolutional neural networks are a machine learning algorithm that is used primarily to classify images (by naming what they see), cluster them by resemblance and similarity (photo search), and perform detailed object recognition within scenes. They are algorithms that can spot and identify faces, objects, street signs, plants, tumors, platypuses and many other aspects of visual data. The efficacy of convolutional neural networks (CNNs) in image recognition is one of the main reasons why the world has woken up to the efficacy of deep learning. They are the

backbone of major breakthroughs in computer vision, which has applications in self-driving vehicles, robotics, drones, security cameras, medical diagnosis, and treatments for the visually impaired.

2.5.1 The Input Image

From the Latin *convolvere*, “to convolve” means to roll together. For mathematical purposes, a convolution is the integral measuring how much two functions overlap as one passes over the other. A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. Convolutional neural networks ingest and process images as tensors, and tensors are matrices of numbers with additional dimensions (. They can be hard to visualize, so let’s approach them by analogy. A scalar is just a number, such as 7; a vector is a list of numbers (e.g., [7,8,9]); and a matrix is a rectangular grid of numbers occupying several rows and columns like a spreadsheet (Rouse et al., 2015). Geometrically, if a scalar is a zero-dimensional point, then a vector is a one-dimensional line, a matrix is a two-dimensional plane, a stack of matrices is a three-dimensional cube, and when each element of those matrices has a stack of feature maps attached to it, you enter the fourth dimension. For reference, here’s a 2 x 2 matrix:

$$[1, 2]$$

$$[5, 8]$$

A tensor encompasses the dimensions beyond that 2-D plane. You can easily picture a three-dimensional tensor, with the array of numbers arranged in a cube. Here’s a 2 x 3 x 2 tensor presented flatly (picture the bottom element of each 2-element array extending along the z-axis to intuitively grasp why it’s called a 3-dimensional array)

$$\left(\begin{array}{c|c|c} \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 5 \end{pmatrix} & \begin{pmatrix} 4 \\ 7 \end{pmatrix} \\ \hline \begin{pmatrix} 3 \\ 4 \end{pmatrix} & \begin{pmatrix} 4 \\ 6 \end{pmatrix} & \begin{pmatrix} 5 \\ 8 \end{pmatrix} \end{array} \right)$$

In code, the tensor above would appear like this: [[[2,3],[3,5],[4,7]],[[3,4],[4,6],[5,8]]]. And here’s a visual:

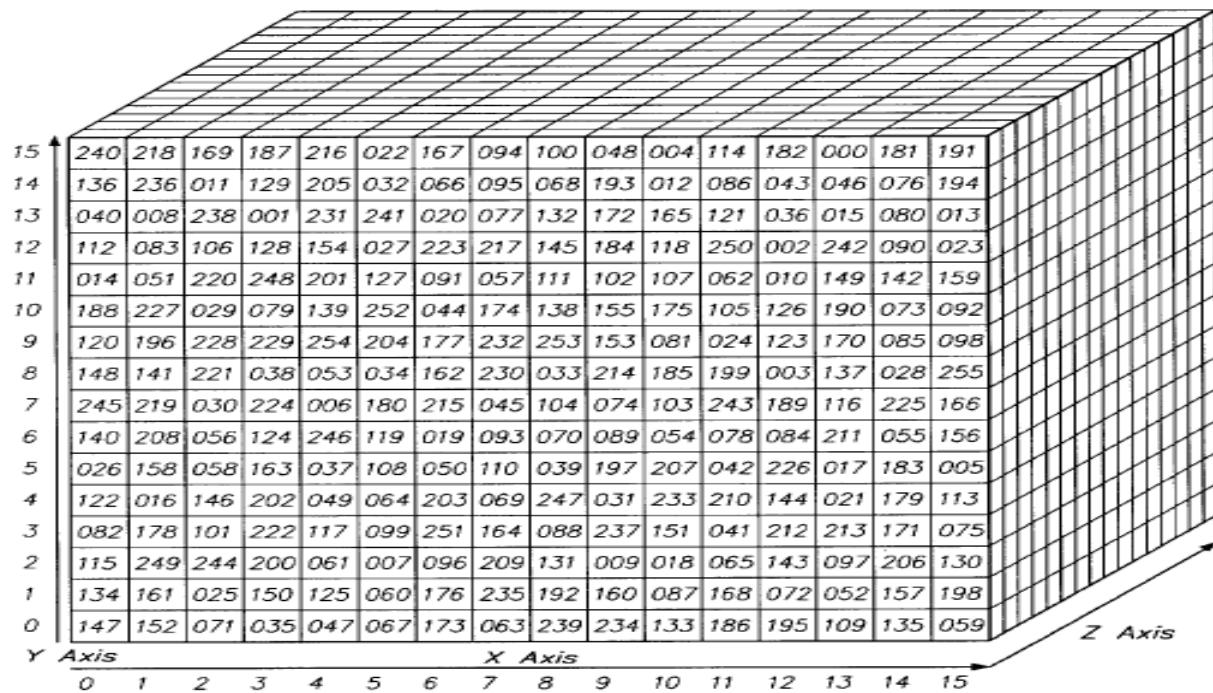


Fig 2.6 A scalar representation of an image

Source: <https://skymind.ai/wiki/convolutional-network>

In other words, tensors are formed by arrays nested within arrays, and that nesting can go on infinitely, accounting for an arbitrary number of dimensions far greater than what we can visualize spatially. A 4-D tensor would simply replace each of these scalars with an array nested one level deeper. Convolutional networks deal in 4-D tensors like the one below (notice the nested array).

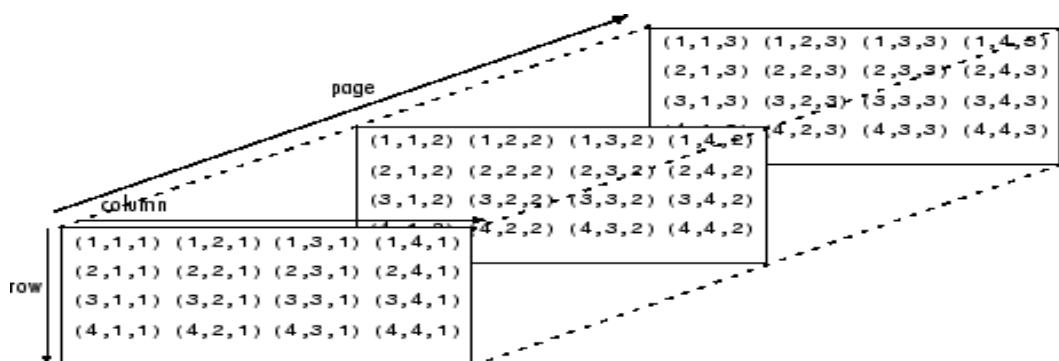


Fig 2.7 The three layers of an image (Red-Green-Blue)

Source: <https://skymind.ai/wiki/convolutional-network>

The first thing to know about convolutional networks is that they don't perceive images like humans do. Therefore, you are going to have to think in a different way about what an image means as it is fed to and processed by a convolutional network.

Convolutional networks perceive images as volumes; i.e. three-dimensional objects, rather than flat canvases to be measured only by width and height. That's because digital color images have a red-blue-green (RGB) encoding, mixing those three colors to produce the color spectrum humans perceive. A convolutional network ingests such images as three separate strata of color stacked one on top of the other. So a convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred to as channels. As images move through a convolutional network, we will describe them in terms of input and output volumes, expressing them mathematically as matrices of multiple dimensions in this form: 30x30x3. From layer to layer, their dimensions change for reasons that will be explained below. You will need to pay close attention to the precise measures of each dimension of the image volume, because they are the foundation of the linear algebra operations used to process images.

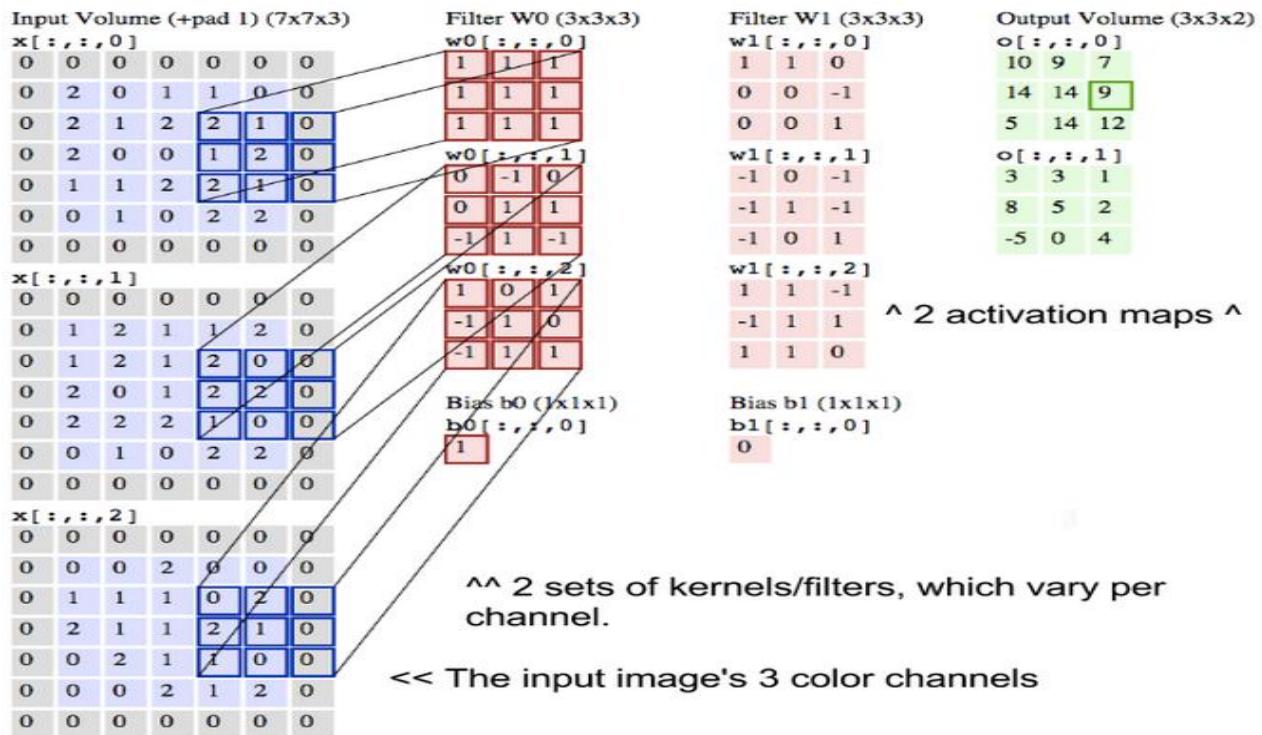


Fig 2.8 Convolutional operation on the three layers of an image(R-G-B)
Source: <https://skymind.ai/wiki/convolutional-network>

Now, for each pixel of an image, the intensity of R, G and B will be expressed by a number, and that number will be an element in one of the three, stacked two-dimensional matrices, which together form the image volume. Those numbers are the initial, raw, sensory features being fed into the convolutional network, and the CNN's purpose is to find which of those numbers are significant signals that actually help it classify images more accurately. Rather than focus on one pixel at a time, a convolutional net takes in square patches of pixels and passes them through a filter. That filter is also a square matrix smaller than the image itself, and equal in size to the patch. It is also called a kernel and the job of the filter is to find patterns in the pixels (Ruchibal 2014).

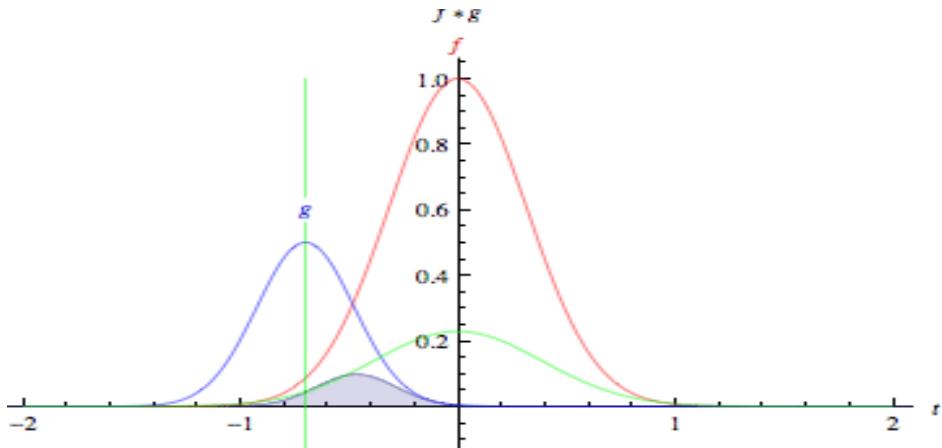


Fig 2.9 Graph showing convolution between the R-G-B images and how they overlap (1)

Source: <http://mathworld.wolfram.com/>

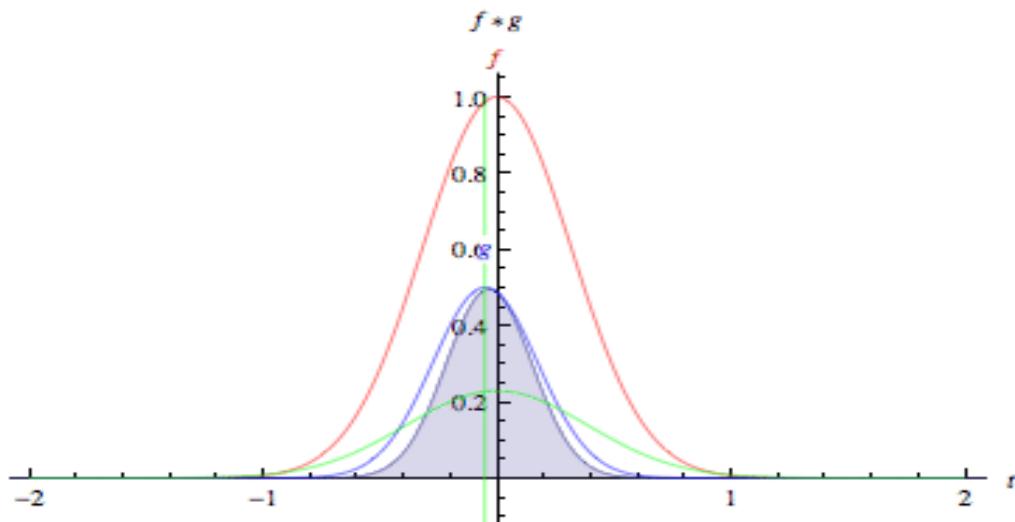


Fig 2.10 Graph showing convolution between the R-G-B images and how they overlap (2)

Source: <http://mathworld.wolfram.com/>

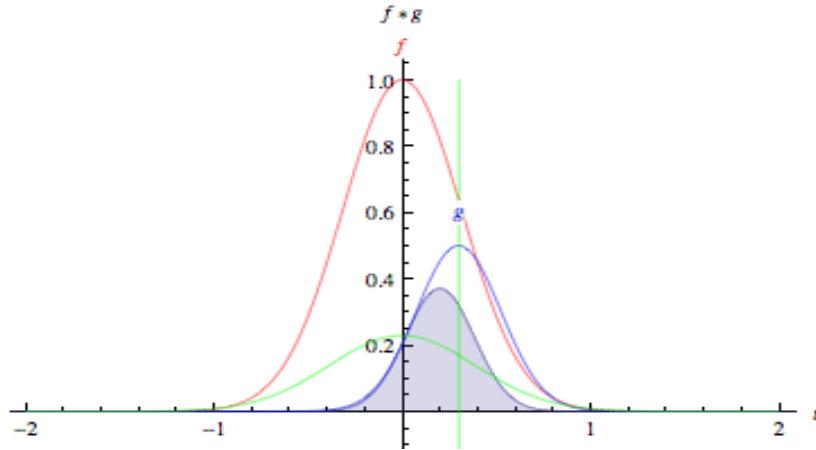


Fig 2.11 Graph showing convolution between the R-G-B images and how they overlap (3)

Source: <http://mathworld.wolfram.com/>

“The green curve shows the convolution of the blue and red curves as a function of t, the position indicated by the vertical green line. The gray region indicates the product $g(\tau)f(t-\tau)$ as a function of t, so its area as a function of t is precisely the convolution.”

Look at the tall, narrow bell curve standing in the middle of a graph. The integral is the area under that curve. Near it is a second bell curve that is shorter and wider, drifting slowly from the left side of the graph to the right. The product of those two functions’ overlap at each point along the x-axis is their convolution. So in a sense, the two functions are being “rolled together.”

What we just described is a convolution. You can think of Convolution as a kind of multiplication used in signal processing. Another way to think about the two matrices creating a dot product is as two functions. The image is the underlying function, and the filter is the function you roll over it.

2.6 Application Areas of Machine Learning in Healthcare

Machine learning is another field which is seeing gradual acceptance in the healthcare industry. Google recently developed a machine-learning algorithm to identify cancerous tumors in mammograms, and researchers in Stanford University are using deep learning to identify skin cancer. Machine Learning in healthcare helps to analyze and process thousands of different data and suggest outcomes, provide accurate and timely risk scores, precise resource allocation, and

has many other applications (Mohammad 2017). Here are some applications of machine learning in healthcare:



1. Identifying Diseases and Diagnosis:

One of the chief machine learning applications in healthcare is the identification and diagnosis of diseases and ailments which are otherwise considered hard-to-diagnose.

This can include anything from cancers which are tough to catch during the initial stages, to other genetic diseases.



2. Drug Discovery and Manufacturing:

One of the primary clinical applications of machine learning lies in early-stage drug discovery process. This also includes R&D technologies such as next-generation sequencing and precision medicine which can help in finding alternative paths for therapy of multifactorial diseases.



3. Medical Imaging Diagnosis:

Machine learning and deep learning are both responsible for the breakthrough technology called Computer Vision. This has found acceptance in the Inner Eye initiative developed by Microsoft which works on image diagnostic tools for image analysis. Another example of this is the detection of malaria parasites in blood smears.



4. Outbreak Prediction:

AI-based technologies and machine learning are today also being put to use in monitoring and predicting epidemics around the world. Today, scientists have access to a large amount of data collected from satellites, real-time social media updates, website information, etc. Artificial neural networks help to collate this information and predict everything from malaria outbreaks to severe chronic infectious diseases. Predicting these outbreaks is especially helpful in third-world countries as they lack in crucial medical infrastructure and educational systems.



5. Personalized Medicine:

Personalized treatments can not only be more effective by pairing individual health with predictive analytics but is also ripe for further research and better disease assessment. Currently, physicians are limited to choosing from a specific set of diagnoses or estimate the risk to the patient based on his symptomatic history and available genetic information.

2.7 Advantages and Disadvantages of Machine Learning:

Every coin has two faces and each face has its own property and features. It's time to uncover the faces of Machine Learning which is a very powerful tool that holds the potential to revolutionize the way things work.

Advantages of Machine learning include but are not limited to the following;

- 1) **Easily identifies trends and patterns:** Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans.
- 2) **No human intervention needed (automation):** With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own.
- 3) **Continuous Improvement:** As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions.
- 4) **Multi-dimensional and multi-variety data:** Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.
- 5) **Wide Applications:** You could be an e-tailer, scientist or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine learning include:

- 1) **Data Acquisition:** Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality.

- 2) **Time and Resources:** ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.
- 3) **Interpretation of Results:** Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.
- 4) **High error-susceptibility:** Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 The Existing System Design

Malaria is caused by protozoan parasites of the genus *Plasmodium*. There are four species of *Plasmodium* that infect man and result in four kinds of malarial fever: *P. falciparum*, *P. vivax*, *P. ovale*, and *P. malariae*. *P. vivax* shows the widest distribution and is characterized by reappearances of symptoms after a latent period of up to five years. With the similar characteristics, *P. ovale* appears mainly in tropical Africa. *P. falciparum* is most common in tropical and subtropical areas. It causes the most dangerous and malignant form of malaria without relapses and contributes to the majority of deaths associated with the disease. *P. malariae* is also widely distributed but much less than *P. vivax* or *P. falciparum* (Robertson 2009). There are three phases of development in the life cycle of most species of plasmodium:

- i. *exoerythrocytic stages* in the tissues, usually the liver;
- ii. *erythrocytic schizogony* (i.e. protozoan asexual reproduction) in the erythrocytes;
- iii. *the sexual process*, beginning with the development of gametocytes in the host and continuing with the development in the mosquito.

According to Morgan *et al.*, (2010) When an infected mosquito bites humans, several hundred *sporozoites* (the protozoan cells that develop in the mosquito's salivary gland and infect new hosts) may be injected directly into the blood stream, where they remain for about 30 min and then disappear. Many are destroyed by the immune system cells, but some enter the cells in the liver. Here they multiply rapidly by a process referred to as *exo-erythrocytic schizogony*. When schizogony is completed, the cells produced by asexual reproduction in the liver termed *merozoites* are released and invade the erythrocytes. In *Plasmodium vivax* and *P. ovale*, some injected sporozoites may differentiate into stages termed *hypnozoites* which may remain dormant in the liver cells for some time before undergoing schizogony causing relapse of the disease. When the released merozoites enter erythrocytes, the erythrocytic cycle begins. This process is referred to as *erythrocytic schizogony*. Within an erythrocyte, the parasite is first seen microscopically as a minute speck of chromatin surrounded by scanty protoplasm (Juri 2011). The plasmodium gradually becomes ring-shaped and is known as ring or immature *trophozoite*. It grows at the

expense of the erythrocyte and assumes a form differing widely with the species but usually exhibiting active pseudopodia (i.e. projections of the nuclei). Pigment granules appear early in the growth phase and the parasite is known as a mature trophozoite. As the nucleus begins to divide, the parasite is known as a *schizont*. Dividing nucleus tends to take up peripheral positions and a small portion of cytoplasm gathers around each. The infected erythrocyte ruptures and releases a number of merozoites which attack new corpuscles and the cycle of erythrocytic schizogony is repeated. The infection about this time enters the phase in which parasites can be detected in blood smears.

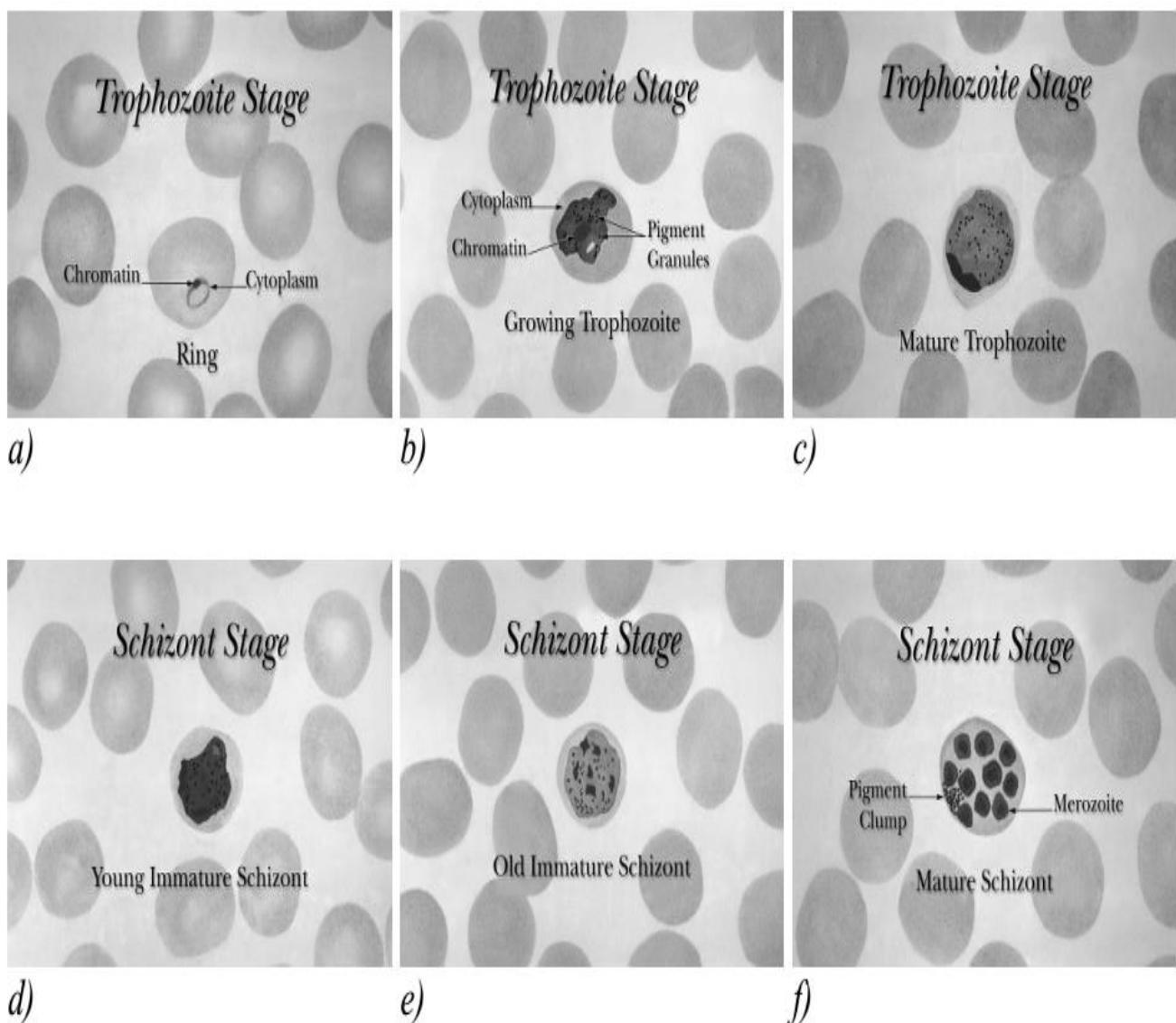


Fig 3.1 Development stages of the Plasmodium parasite
Source: <https://malariajournal.biomedcentral.com>

3.2 Input Analysis of the Existing System

Blood smears or blood films are microscopic slides prepared from a blood sample that allow the microscopic inspection of blood cells. Blood smears are typically used for investigation of hematological disorders and for detection of parasites, such as the Plasmodium. Two sorts of blood smears are traditionally used. Thin blood smears allow better species identification, because the appearance of the parasites is better preserved in this preparation. Thick blood smears allow screening of a larger volume of blood and, therefore, they can give more than a ten-fold increase in sensitivity over thin films. However, the appearance of the parasite is more distorted and, therefore, distinguishing between the different species can be more difficult.

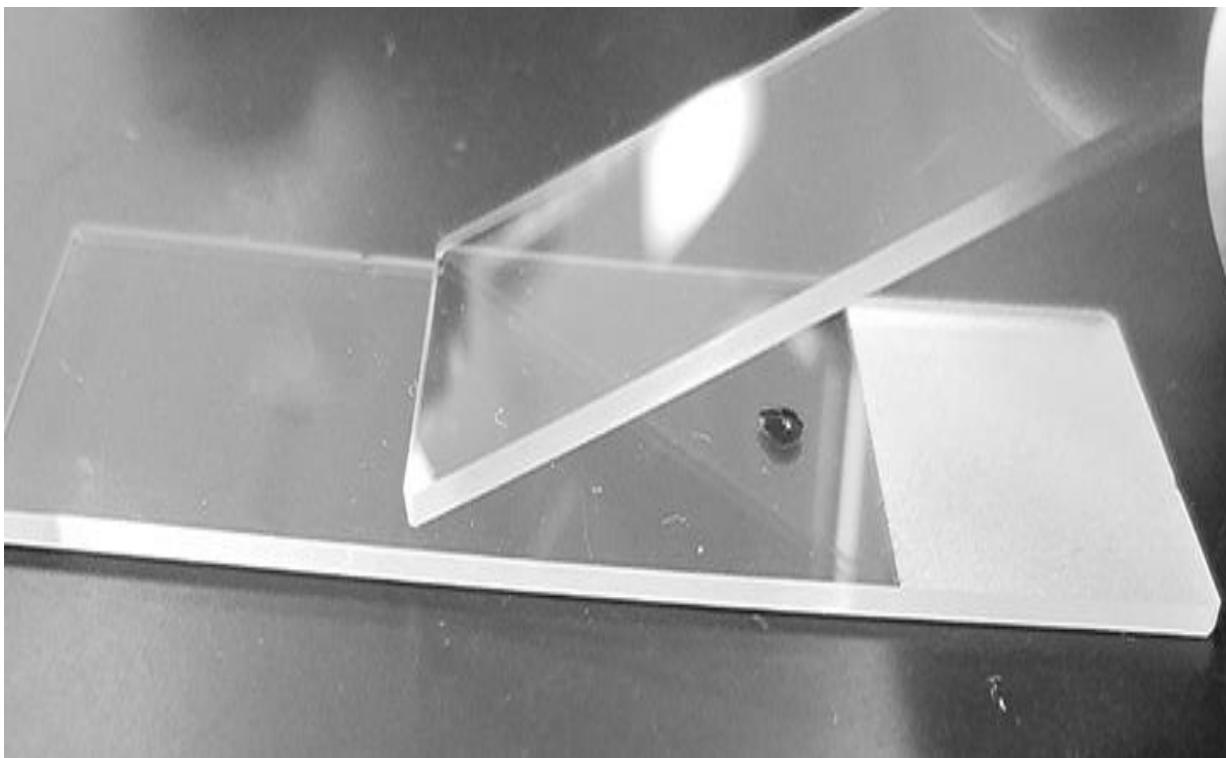


Fig 3.2 Blood smear preparation for malaria diagnosis
Source:www.cliniciansbrief.com-diagnostic-blood-smear-preparation

In principle, blood films are prepared by placing a drop of blood on one end or into the center of a slide and spread with the corner of another slide or a swab stick to cover an oval area along the slide. The aim is to get a region where the cells are sufficiently spread to be counted and differentiated.

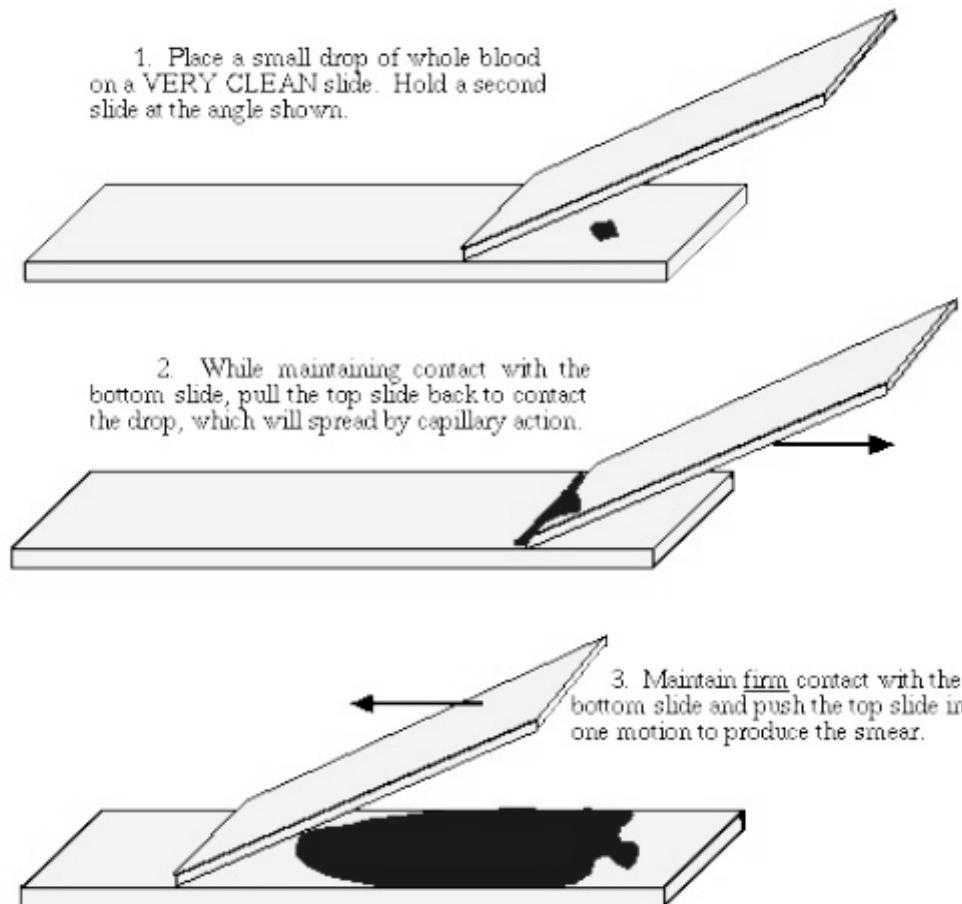


Fig 3.3 Illustration of blood smear preparation for malaria diagnosis
Source:www.cliniciansbrief.com-diagnostic-blood-smear-preparation

The well spread part of the blood smear, specifying the working area for microscopic analysis, is defined as a zone that starts on the body film side when red blood cells stop overlapping and finishes on the feather edge side when red blood cells start to lose their clear central zone. The smear is then thoroughly dried in an incubator at 37°C for around one hour. The dry film can be subsequently stained using Giemsa dilution.

3.3 Output Analysis of the Existing System

Diagnosis of malaria involves the identification of malaria parasite in the blood samples obtained from the patient. Although this seems simple, the effectiveness of the diagnosis is subject to many factors. The microscopic tests involve staining the blood sample with Giemsa and direct

visualization of the malaria parasite under the microscope. The direct microscopic visualization of the malaria parasite in the blood smears obtained from the patient has been the accepted method for the diagnosis of malaria in most settings, from the clinical laboratory to the field surveys. The careful examination of a well-prepared and well-stained blood film currently remains the gold standard for malaria diagnosis. Giemsa stain is used to differentiate nuclear and cytoplasmic morphology of platelets, red blood cells, white blood cells and parasites. Giemsa staining solution stains up nucleic acids and, therefore, parasites, white blood cells, and platelets, which contain DNA, are highlighted in a dark purple color. Red blood cells are usually colored in slight pink colors (Cabezos 2012).



Fig 3.4 Manual diagnosis of malaria in blood smear
Source:www.cliniciansbrief.com-diagnostic-blood-smear-preparation

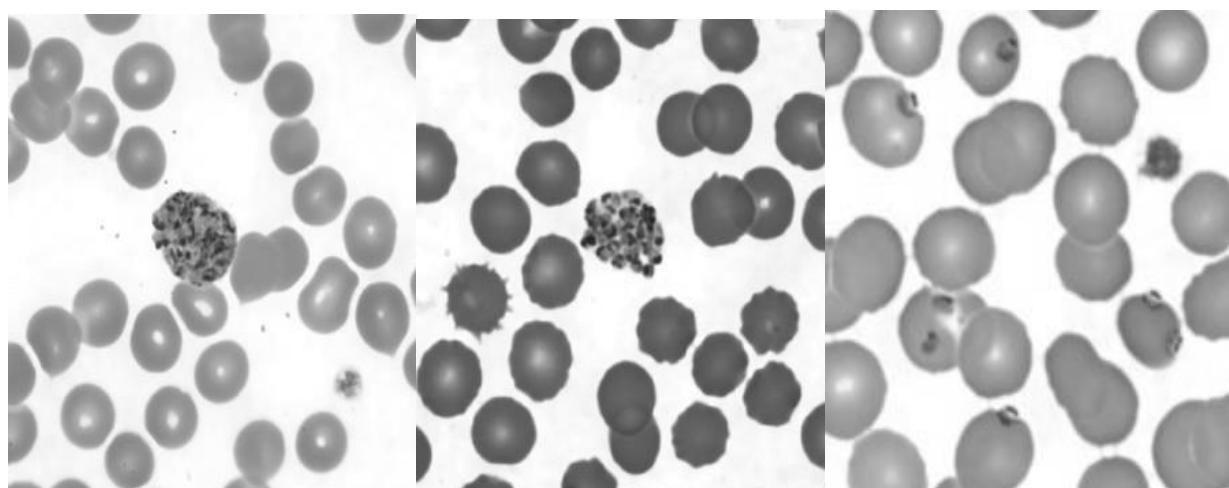


Fig 3.5 Samples of already stained blood smear images showing malaria parasites in the RBCs
Source: *Faith Mediplex, Airport Road, GRA, Benin City, Edo State.*

These images of Giemsa stained blood smears above (Fig 3.2) were gotten from the Faith Mediplex hospital laboratory and they have the following common characteristics:

- The Images are available in different magnifications and sizes. The images are available in TIFF and JPEG format with the resolution of 2 to 3 megapixels
- Digital images are obtained by scanning and, therefore, contain a part of the noise and artifact from the sample and from the microscope light also noise from the chemical development process or from the scanner.
- The Images exhibit high variability in color tone, intensity, contrast, and illumination.

The output of the existing system is usually a laboratory report on the malaria test. Clinical findings are confirmed and documented using a laboratory report. In addition to ordering the malaria specific diagnostic tests described below, the health-care provider should conduct an initial workup and request a complete red blood cell count and a routine chemistry panel. In the scenario that the person does have a positive malaria test, these additional tests will be helpful in determining whether the patient has uncomplicated or severe manifestations of the malaria infection. Specifically, these tests can detect severe anemia, hypoglycemia, renal failure, hyperbilirubinemia, and acid-base disturbances (Scribd 2010). The information contained in the lab report includes the following:

1. The name of the Laboratory
2. The name of the patient
3. The patient's age
4. The nature of test carried out
5. Red Blood Cells count
6. Presence of the Malaria parasite
7. Type of Malaria parasite present

8. Summary of the test result
9. Other observations made during the test
10. Etc.

Below is a sample of a Malaria diagnosis laboratory report:

SUBURBAN DIAGNOSTICS			
PRECISE TESTING • HEALTHIER LIVING			
X-Ray Sonography Mammography BMD (Dexa-Scan) ECG Stress Test / TMT 2D Echo OPG Pathology Carotid Doppler Eye Examination Dental Examination Diet Consultation Preventive Health Check-Up Audiometry Spirometry			
CID	: 1527903457	SID	: 177400416995 R
Name	: MR. KALPESH PATOLE	Registered	: 06-Oct-2015 / 22:20 E
Age / Gender	: 31 Years / Male	Collected	: 06-Oct-2015 / 22:22 P
Dr.	: SAGAR KAJBALE	Reported	: 07-Oct-2015 / 18:37 O
Reg. Location	: Thane Vartak Nagar	Printed	: 07-Oct-2015 / 21:10 R
COMPLETE BLOOD COUNT (CBC)			
PARAMETER	RESULTS	BIOLOGICAL REF RANGE	METHOD
Others	Normocytic, Normochromic	-	-
WBC MORPHOLOGY	-	-	-
PLATELET MORPHOLOGY	-	-	-
COMMENT	-	-	-
Specimen: EDTA Whole Blood			
MALARIAL PARASITE, EDTA WB Positive: Pl vivax			
* Sample processed at SUBURBAN DIAGNOSTICS Thane Vartak Nagar Lab *** End Of Report ***			

Fig 3.6 A malaria diagnosis laboratory report
Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5396426/>

3.4 Flowchart of the Existing System

The first suspicion of malaria is usually based on clinical criteria, especially fever or a recent history of fever; however, even in areas of high transmission, most cases of fever are usually not due to malaria. As the clinical manifestations of malaria are nonspecific, a diagnosis based on clinical symptoms alone results in a high number of false-positive results; often, other diseases are

overlooked or not treated in a timely manner, contributing to significant morbidity and mortality due to non-malaria illness. False-positive results also lead to misuse of antimalarial drugs, exposure of parasites to sub-therapeutic blood levels of the drugs and development of resistance, increased costs to the health services and patient dissatisfaction.

An accurate laboratory diagnosis is essential, as false-negative results can lead to untreated malaria and potentially severe consequences, including death. False-negative results can also significantly undermine both clinical confidence in laboratory results and the credibility of the health services within a community. The flowchart shows how information flows from one point to another during the diagnosis and subsequent treatment of malaria in an infected patient.

Below is the information flow diagram:

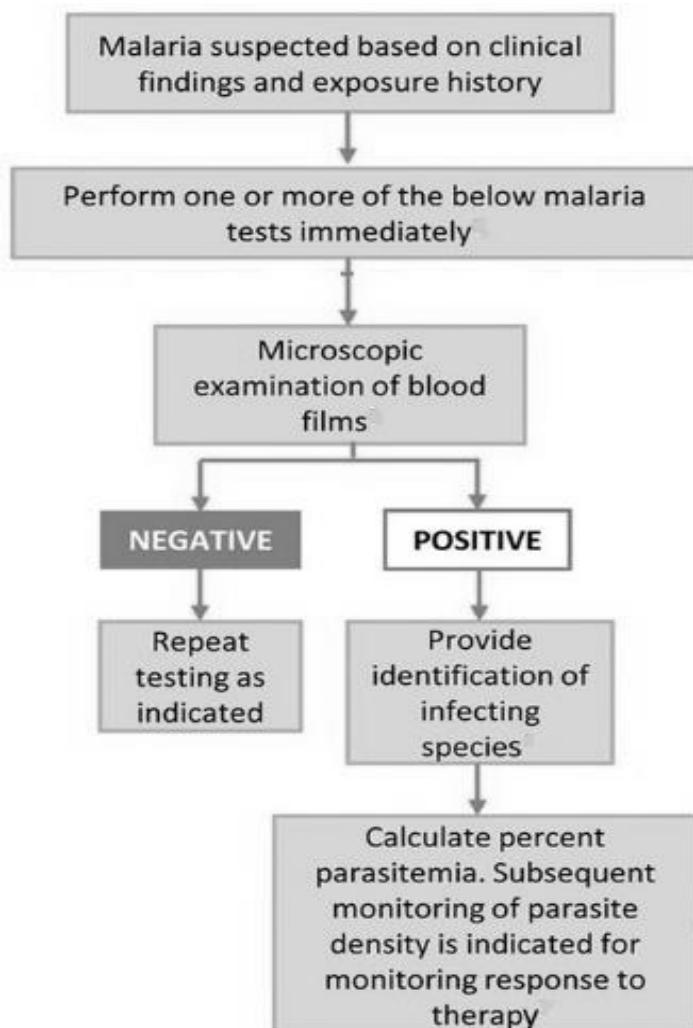


Fig 3.7 A typical Malaria diagnosis flowchart
Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5396426/>

3.5 Limitations of the Existing System

The accuracy of the diagnosis of malaria using the existing system depends on the availability of a competent pathologist using good-quality reagents for examining well-prepared slides under a well-maintained microscope with an adequate light source and with a low-to-moderate workload. It has therefore been difficult in some cases to maintain a good standard of malaria diagnosis, especially in rural areas, where over 60% of malaria cases occur. The factors that limit the availability and quality of the existing system include:

- Lack of resources to provide all laboratories with equipment and good-quality reagents for microscopy
- Difficulty in maintaining microscopy facilities in good order and lack of microscope maintenance capability
- Lack of electricity, water and suitable laboratory facilities in some rural areas
- Heavy workloads, which delay the provision of results to clinical staff

3.6 Objectives of The New System

The manual inspection and analysis of slides is, however, laborious, time-consuming and requires a skilled and well-trained operator. Moreover, the accuracy of the final diagnosis is subject to the skill and the experience of the technician and the time spent studying each slide. The main objective of the new system is to be able to determine the presence of the malaria parasites in Giemsa stained blood smears. Other objectives include the following;

- To reduce the chance of omission or false detection due to human error.
- To facilitate the accurate and timely diagnosis of malaria infection.
- To help in battling malaria outbreaks where hundreds of blood samples may need to be tested quickly and accurately.

In this context, the development of a mechanism that automates the process of evaluation, quantification and classification in blood samples becomes a high priority and the aim of this work was to contribute to improvement upon malaria microscopy diagnosis by removing the reliance on the performance of a human operator for diagnostic accuracy. A number of methods have been proposed for automatic parasite detection in Giemsa stained blood films based on different

approaches. These approaches include pixel-based parasite detection, detection based on morphological processing of segmented parasites, or detection by extracting image features using convolutional neural networks. In this work, detection of malaria parasites in blood smears is based on the last approach which is the use of convolutional neural networks.

3.7 Justification of the New System

The limitations and shortcomings of the existing system has already been described, related to sensitivity, specificity, accuracy, precision, time consumption, cost-effectiveness, labor intensiveness, the need for skilled pathologists and the problem of inexperienced technicians. In view of the problems present in the existing system, it is important to seek for an improvement. This improvement is computerization of the diagnosis system. This new automated system will have the following advantages:

1. Digitization of Output: The output is an analyzed image of the blood sample and the parasites found in the blood sample are detected by the computer within seconds.
2. Simpler and faster Diagnosis: It will be possible to check more blood samples faster, as protracted visual checks are replaced by fast computers.
3. Reliability: Contrary to a human eye, cameras and computers never get tired. The human factor is eliminated; you will not notice any fluctuations in reliability based on how your controllers slept that day or what day of the week it is.
4. Accuracy: The automated diagnosis has a very high accuracy due to the elimination of possible human error.
5. Reduction of costs: Automated diagnosis is relatively cheaper to implement than manual diagnosis.
6. Availability: The ability of running the automated diagnosis 24 hours a day and the machine is always there to render services.
7. Higher Quality: Machine vision technology is unique in its ability to resolve the trade-off between raising quality and cutting costs.

3.8 The New System Design

A number of new methods have been developed in recent years for the diagnosis of malaria. These include the use of fluorescent microscopy, rapid antigen detection methods and polymerase chain reaction (PCR) - based techniques that detect specific nucleic acid sequences. Despite these advances, malaria diagnosis by means of manual microscopy remains the most widely and commonly used method. Usually, these jobs are conducted by experienced pathologists manually. Microscopic diagnosis entails examining blood smears for the presence of Plasmodia. Unfortunately, there are also disadvantages to the method: substantial costs are incurred purchasing and maintaining microscopes and training technicians, the technique is labor intensive and time-consuming and the accuracy of the final diagnosis relies on the skill and experience of the technician and the time spent studying each slide. Variable smear quality and slide degeneration with time are also problematic. In this research, we proposed an automated system based on supervised machine learning to detect malaria plasmodium which is able to eliminate the most important limits of microscopic method, that is:

- i. Time-consuming and tiring job
- ii. Low accuracy even in experts.

Here in this research we mainly focus on the task of determining the presence of the parasites and highlighting them for ease of identification, because it is the most essential and time-consuming step in the diagnosis of malaria. Also, we propose a motorized microscope which is fully matched with image processing procedure to make the whole diagnosis process automatic. Furthermore, it means that the physician just puts the blood smear under the lens of microscope and runs the system; after a few minutes the report, which includes the number of RBCs and parasites, is issued.

3.9 Flowchart of the New System

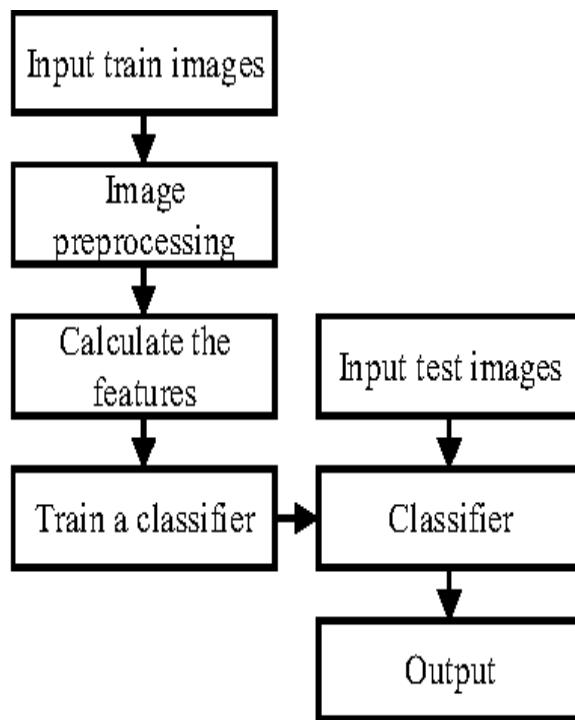


Fig 3.8 Flowchart of the new system
Source: My computer/Photoshop Express

An automated diagnostic method can be developed by understanding the diagnostic process and implementing it using a machine learning algorithm. The machine learning algorithm should perform diagnosis more or less imitating the manual microscopy. The developed algorithm should be capable of functioning in an unsupervised environment and should be robust with minimal false negatives (leading to high sensitivity). The unsupervised nature of the proposed machine learning diagnostic method should reduce human intervention, and in so doing speed up the diagnosis process. The machine learning algorithm must also be sensitive enough to detect malaria parasites at all stages particular at the early stages of their life cycle and must be capable of doing this without missing any parasite irrespective of image variations. In order to perform diagnosis, the method must be capable of differentiating between malaria parasites and artefacts.

3.10 Requirements for the New System

The first requirement in this research is blood smears which are Giemsa stained microscopic slides prepared from blood samples that allow microscopically examinations of blood cells. Thin blood

smears allow better species identification, because the appearance of the parasites is better preserved in this preparation. Thick blood smears allow screening of a larger volume of blood, therefore, they can give more than a ten-fold increase in sensitivity over thin films. In this research, morphological properties are important for us, hence we used thin films. For effective malaria diagnosis, blood films should be prepared as fast as possible after blood samples are taken. Such films adhere better to the slides; leave a clearer background after drying, thus, parasite and red cell changes are minimal. After preparing blood films they should be examined by a microscope.

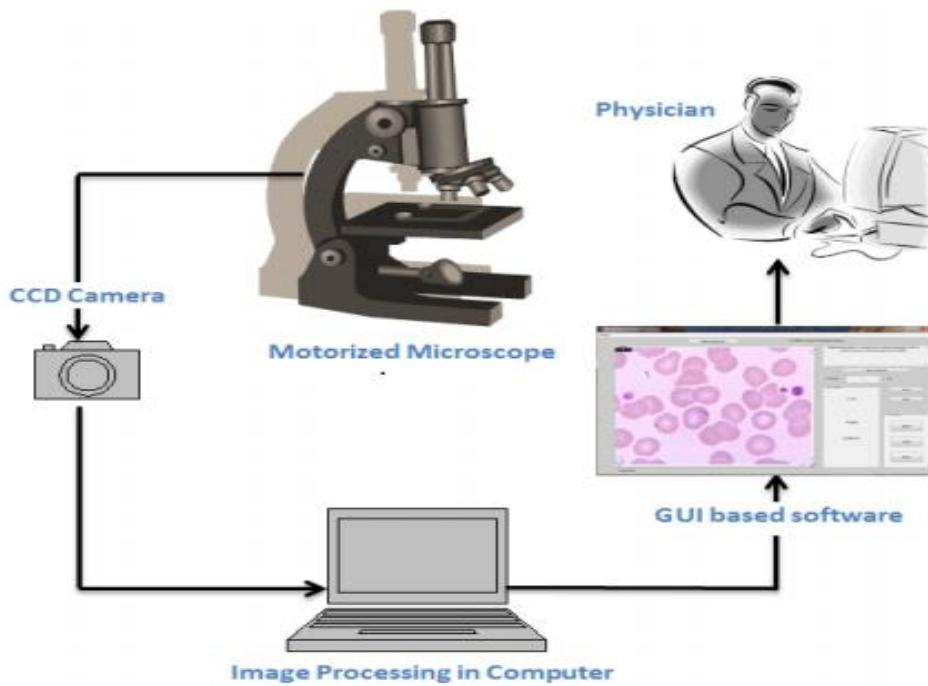


Fig 3.9 Overall scheme of proposed Malaria diagnosis system
 Source: <https://www.vinodsblog.com>

A microscope is equipped with two stepper motors which move the blood samples under lens quite smoothly. The amount of movement in each direction is calculated by a microcontroller installed on the microscope board to avoid taking overlapped photos. This also helps avoid calculating each RBC more than once. The photos taken by CCD are transmitted to the image processing program running on a computer. RBCs and infection are detected simultaneously and the final report is issued.



Fig 3.10 Camera attached to Microscope for acquiring images of blood samples
Source: *Faith Mediplex Hospital Laboratory*

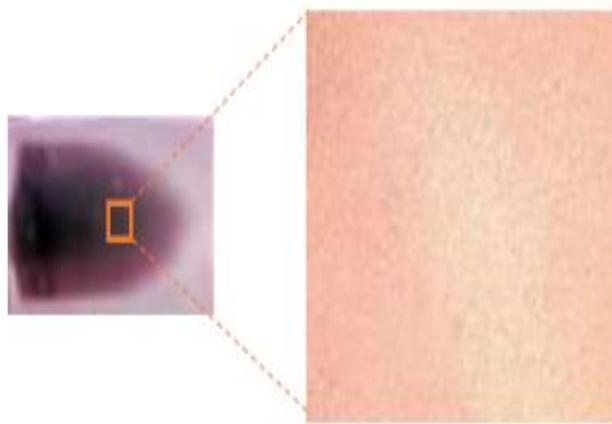


Fig 3.11 Thin blood smear slide and acquired digital microscope image at 40X magnification.
Source: www.semanticscholars.com

The images recorded with the magnification of 1000X are indicates parasite clearly. The Olympus DP25 digital camera of 5 MP attached to the light microscope Olympus BX51, which is connected with the computer, along with the user interface software (DP2 BSW) are shown in Figure 3.10. The acquired the blood image from the focused slide area are collected. The typical malarial thin blood smear image acquired at 40X magnification is shown in the Figure 3.11. Blood images acquired with the various magnifications such as 100X, 200X, 400X and 1000X are shown respectively in the Figures 3.12 A-D. A total of 1,160 images were considered for classification of malarial and non-malarial classes. The acquired thin blood smear image has red blood corpuscles (RBC), malarial parasites, Platelets and other objects. But the proposed technique focus on diagnosis of malaria is based on examination of RBCs, since the malarial parasite infects the RBC.

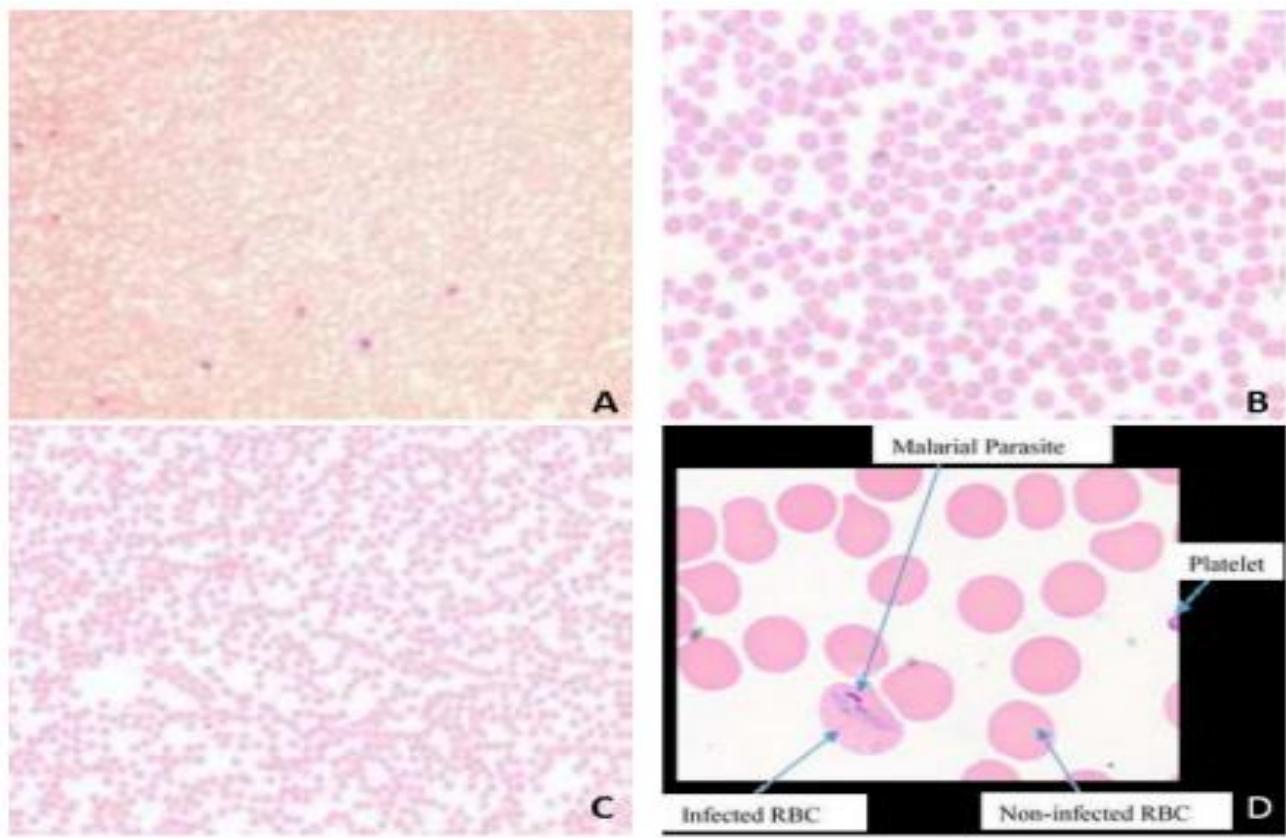


Fig 3.12 The magnified microscopic blood images of the blood sample: (a) at 100X; (b) 200X; (c) 400X; (d) 1000X

Source: <https://www..machinecurve.com/malaria-diagnosis-and-computer-vision>

CHAPTER FOUR

DESIGN, IMPLEMENTATION AND DOCUMENTATION

4.1 The Training Data

Images of Giemsa stained malaria-infected blood smears were obtained from the Faith Mediplex Hospital Laboratory and they have the following common characteristics as seen in Fig 4.1:

- Images are available in different magnifications and sizes. The images are available in JPEG format with the resolution of 2 to 3 megapixels.
- Digital images are obtained by scanning and, therefore, contain a part of the noise and artifact from the sample and from the microscope light also noise from the chemical development process or from the scanner.
- Images exhibit high variability in color tone, intensity, contrast, and illumination. The overall color tone varies significantly from grayish, blue, purple, and pink to yellowish and it may even change from the center of the image to its borders. Some images have very low contrast while some images exhibit high contrast between infected and non-infected cells. Many images suffer from irregular illumination.
- The overall shape and appearance of the cells may also vary substantially among the slides. Some cells lack their clear central parts and, in some images, cells may assume shapes that differ from the usual circular shape. Moreover, red blood cells are often overlapping and may form big clusters. Occasionally, blurring and various artifacts may also appear.

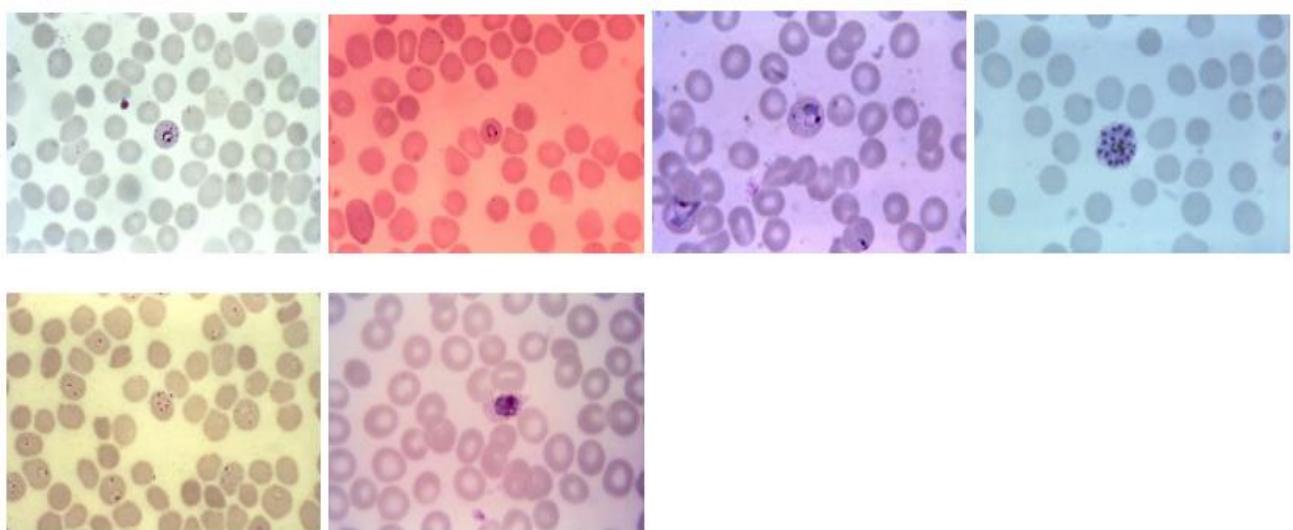


Fig 4.1 Samples of available stained blood smear images showing differences in color tone and illumination.
Source: Faith Mediplex Hospital Laboratory

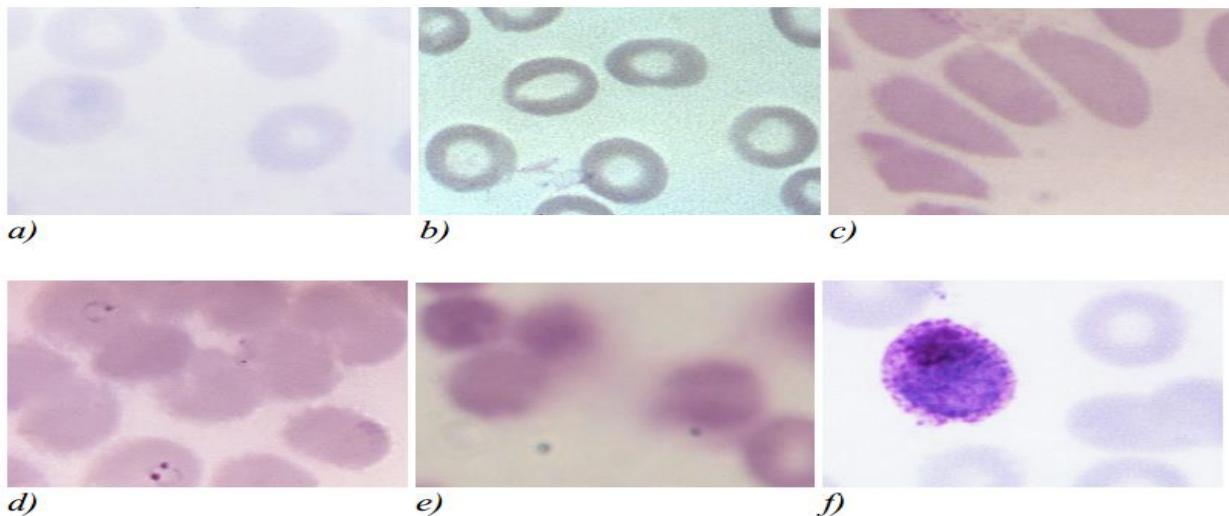


Fig 4.2 Cropped samples of available blood smear images showing different qualitative characteristics of the input images.

Source: Faith Mediplex Hospital Laboratory

The selection of the features for the further evaluation was based on the visual differences between infected and non-infected red blood cells, the measures of infected red blood cells that are commonly used by technicians for manual microscopic diagnosis, and the feature selection used by other cytological studies. The chosen features can be grouped into three categories: shape features, intensity features, and texture features.

4.1.1 Shape Features

These features express the overall size and shape of the cell without taking the density of the cell into account. Although the applicability of the features based only on the shape of the cell is necessarily limited, they can be useful in distinguishing development stages of certain species of plasmodium which are characterized by a specific shape of the infected cell and they may also be useful in distinguishing between red blood cells and other objects, such as white blood cells, platelets or artifacts.

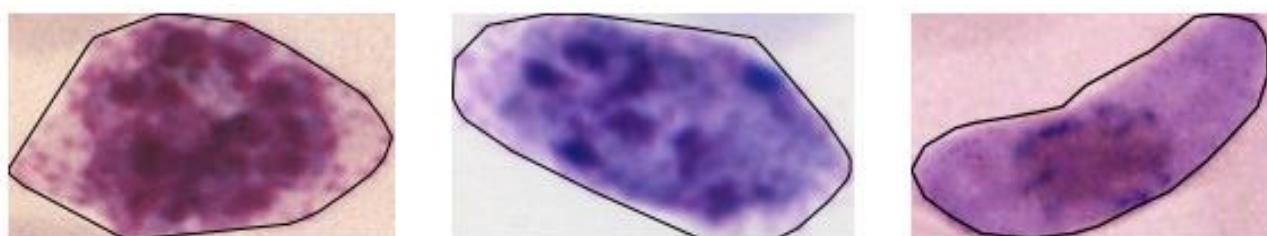


Fig 4.3 Infected red blood cells with distinct shapes

Source: MATLAB

Although the number of classes had to be reduced to only two classes – an infected and a non-infected cell – due to the insufficient number of samples, several of these features were evaluated both as guidance for possible future studies, which would include also classification distinguishing between infected and non-infected red blood cells and other objects and classification of parasites according to the development stage and species of the plasmodium, and also to evaluate whether these features could possibly improve the overall discrimination power when combined with other features.

4.1.2 Intensity Features

Intensity and color are the most palpable visual differences between red blood cells and parasites. This difference is a result of the staining procedure during which all the object containing DNA, and thus also the plasmodium parasites, are stained in saturated purple color. The intensity features may be advantageous especially when the texture of such a cell is indistinct.

4.1.3 Textural Features

Depending on the development stage and the species of the parasite with which a red blood cell is infected, different types of texture can be observed. Parasites of early development stages form rings with distinct speckles of chromatin. Since the rest of the area of the red blood cell is usually quite intact, the texture is given mainly by the chromatin speckles and possibly by the ring lines which are not always visible (Fig 4.4).

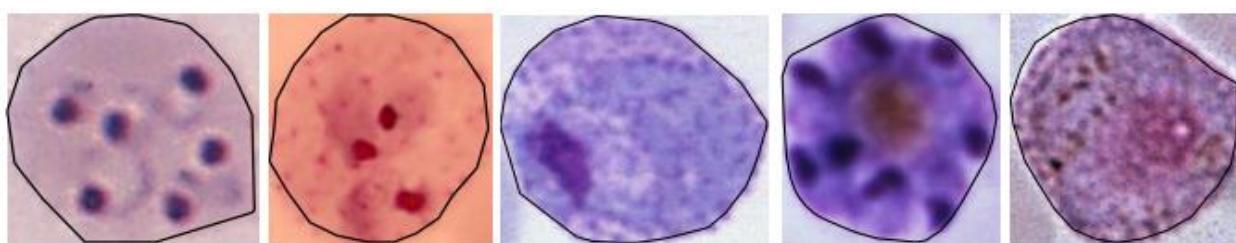


Fig 4.4 Different textural properties of infected red blood cells
Source: MATLAB

More distinct texture can be observed in red blood cells infected by parasites in later stages of development. Pigment granules appear early in the growth phase of the parasite as the immature ring-shaped trophozoite becomes mature trophozoite. Depending on the species, the texture in this phase consists mainly of large amoeboid cytoplasm with large chromatin and fine pigment dots called Schüffner's dots.

4.2 Choice of Programming Language

The entire project, including all functions, segmentation, image database files, and feature evaluation scripts, have been implemented using MATLAB programming environment version 2019. Although we cannot guarantee that all functions will work properly with any older version of MATLAB, most of the functions, and especially the main script, are simplified where necessary to ensure backward compatibility. MATLAB is a high-performance language for technical computing, which integrates computation, visualization, and programming in an easy-to-use environment. It is an interactive system using an array that does not require dimensioning as the basic data element. Typical uses include machine learning, math and computation, algorithm development, computer vision, data acquisition, analysis and visualization, modeling and simulation, scientific and engineering graphics, and application development including graphical user interface building. It removes the need of programming many routine tasks for numerical computing and allows easy and quick displaying of results both in numerical form as well as in the form of 2D or 3D graphs. The open-architecture of MATLAB allows programmers to incorporate their own area specific set of functions implemented in separate m-files into MATLAB, so that they can be easily used by other functions and scripts written in MATLAB. The following toolboxes are used and required in order to run all functions and scripts in this project:

- Image Processing Toolbox
- Deep Network Designer Toolbox

The Image Processing toolbox is essential as it is used by most of the functions and scripts and it is the only toolbox required for running the segmentation method and the GUI. There are many reasons why it is easier to learn and use MATLAB than other general purpose languages and here are a few:

1. For one, basic MATLAB usage abstracts away almost all of the tricky/annoying/fun details commonly associated with general purpose programming. No memory management, no complicated integration of external libraries, no need to learn about type systems, very simple syntax, etc. Structure is as simple as can be. Calling functions from other files basic and simple.
2. For two, MATLAB is self-contained. It is the language, the runtime, the REPL, the platform, the IDE, and the debugger. Once MATLAB is installed, all you need to do is launch it and everything you need is right there, ready to go. Additional libraries, or “toolboxes”, are

extensive (if not expensive) and easy to work with. It also has very detailed help documents for each function.

The default data type in MATLAB is a matrix, so there are umpteen different ways to slice and dice them. This is very handy for computer vision and data processing, which are two very common uses for MATLAB. The same tasks are not always very easy in other languages.

4.3 The Convolutional Neural Network

Convolutional neural networks are deep artificial neural networks that are used primarily to classify images (e.g. name what they see), cluster them by similarity (photo search), and perform object recognition within scenes (Junhao 2016). They are algorithms that can identify many aspects of visual data. These make the networks more efficient to implement and vastly reduce the amount of parameters to tune which is why it is ideal for this project. Most recent well performing ConvNets are mainly built from three types of (hidden) layers, convolution layers, pooling layers and fully connected layers although they employ other layers to improve overall accuracy. Convolutional neural networks employ deep learning. Deep learning can be seen as an extension of the well-known multilayer neural network classifiers trained with back-propagation, except that many more layers are used. There are also different kind of layers that are used in typical successions. Deep learning typically requires large training sets. This is the reason why medical applications have been among the last applications to adopt deep learning, as annotated training images are significantly harder to obtain because of expert knowledge requirements and privacy concerns. In this project, we intend to use a convolutional neural network to detect the presence of malaria parasites by differentiating between infected and uninfected cells in thin blood smears.

4.4 Architecture of the Convolutional Neural Network

ConvNet is not just a deep neural network that has many hidden layers. It is a deep network that imitates how the visual cortex of the brain processes and recognizes images. That is how much ConvNet differs in concept and operation from the other neural networks. This section briefly introduces the fundamental architecture of ConvNet. ConvNet includes the feature extractor in the training process rather than designing it manually.

The feature extractor of ConvNet is composed of special kinds of neural networks, of which the weights are determined via the training process. The fact that ConvNet turned the manual feature extraction design into the automated process is its primary feature and advantage. ConvNet yields better image recognition when its feature extraction neural network is deeper (contains more layers), at the cost of difficulties in the training process. It consists of a neural network that extracts features of the input image and another neural network that classifies the feature image.

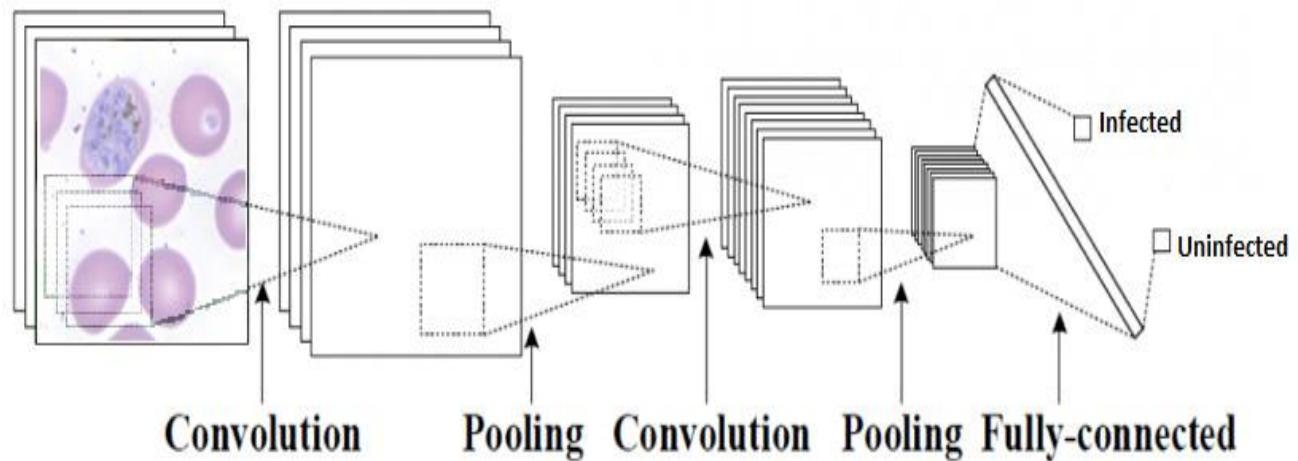


Fig 4.5 Architecture of the convolutional neural network

Source: My computer/Photoshop Express

The feature extraction neural network consists of piles of the convolutional layer and pooling layer pairs. The convolution layer, as its name implies, converts the image using the convolution operation. It can be thought of as a collection of digital filters. The pooling layer combines the neighboring pixels into a single pixel. Therefore, the pooling layer reduces the dimension of the image. As the primary concern of ConvNet is the image; the operations of the convolution and pooling layers are conceptually in a two-dimensional plane. This is one of the differences between ConvNet and other neural networks. In summary, ConvNet consists of the serial connection of the feature extraction network and the classification network. Through the training process, the weights of both layers are determined. The feature extraction layer has piled pairs of the convolution and pooling layers. The convolution layer converts the images via the convolution operation, and the pooling layer reduces the dimension of the image. The classification network usually employs the ordinary multiclass classification neural network.

4.4.1 Image Input Layer

The input layer is the first layer of our convolutional neural network. The Input layer in the CNN should contain image data. Image data is represented by three dimensional matrices as we saw earlier. You need to reshape all the images it into a single dimension. Suppose you have some images of dimension 150 x 150 and other images of varying dimensions like 200 x 250, 300 x 300, 150 x 200, etc., we need to convert them all into a single uniform dimension before feeding them into the input layer. For this project, I have chosen the dimension of 200 x 200 for all the input images and resized them accordingly. We then specify the color bit after the image dimension where black and white images = 2 bits (B-W) and colored images = 3 bits (R-G-B).

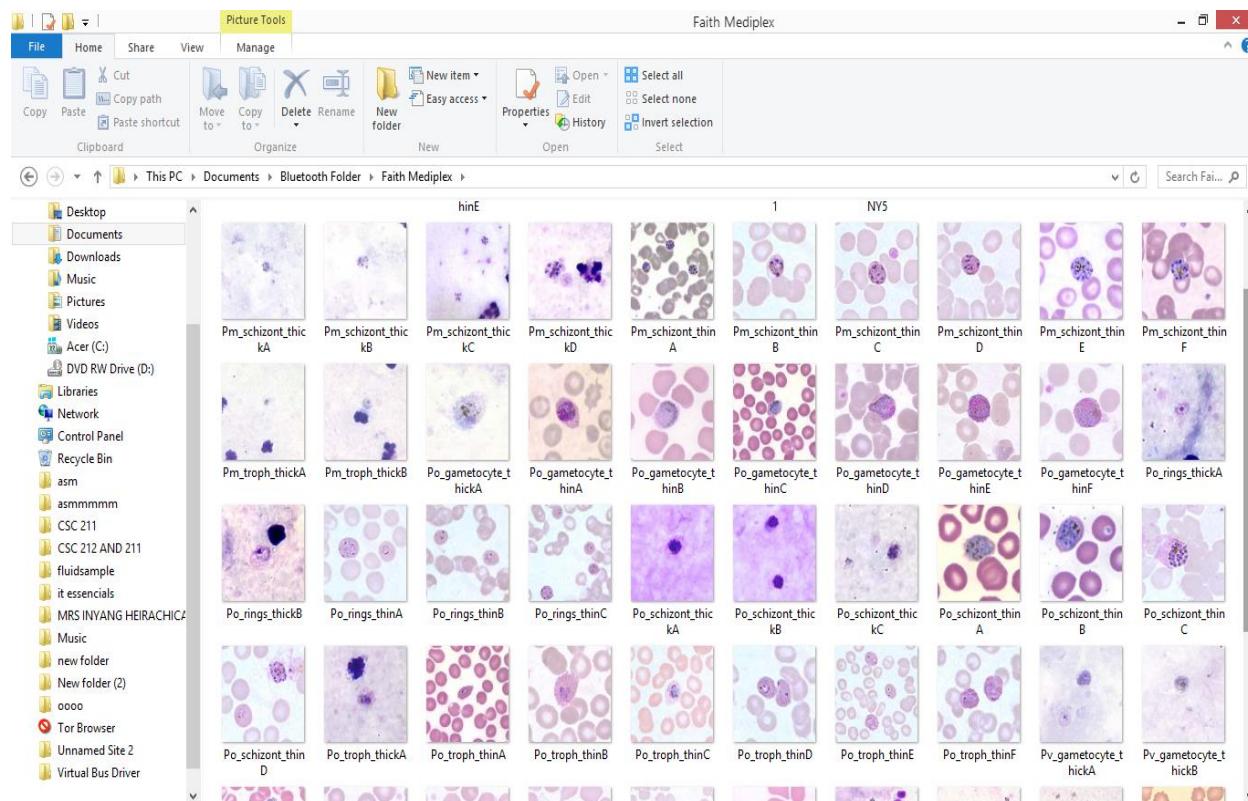


Fig 4.6 Snapshot of some of our training images for the infected class

Source: My computer

One of the distinct characteristics of the input layer is that artificial neurons in the input layer have a different role to play – the input layer being constituted of “passive” neurons that do not take in information from previous layers because they are the very first layer of the network.

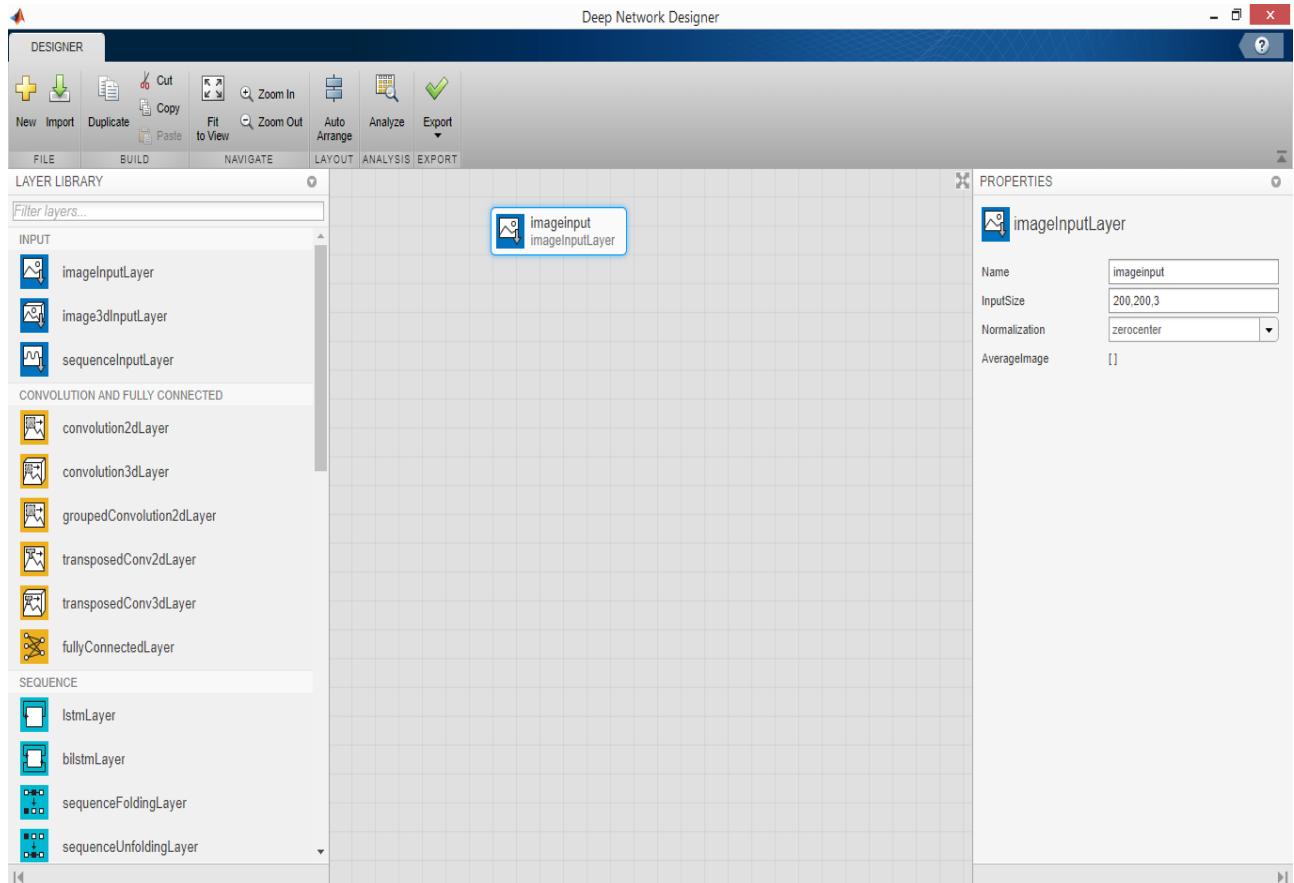


Fig 4.7 The Input Layer
Source: MATLAB

4.4.2 Convolution Layers

This section explains how the convolution layer, which is one side of the feature extraction neural network, works. The convolution layer generates new images called feature maps. The feature map accentuates the unique features of the original image. The convolution layer operates in a very different way compared to the other neural network layers. This layer does not employ connection weights and a weighted sum instead, it contains filters that convert images. We will call these filters convolution filters. The process of inputting the image through the convolution filters yields the feature map.

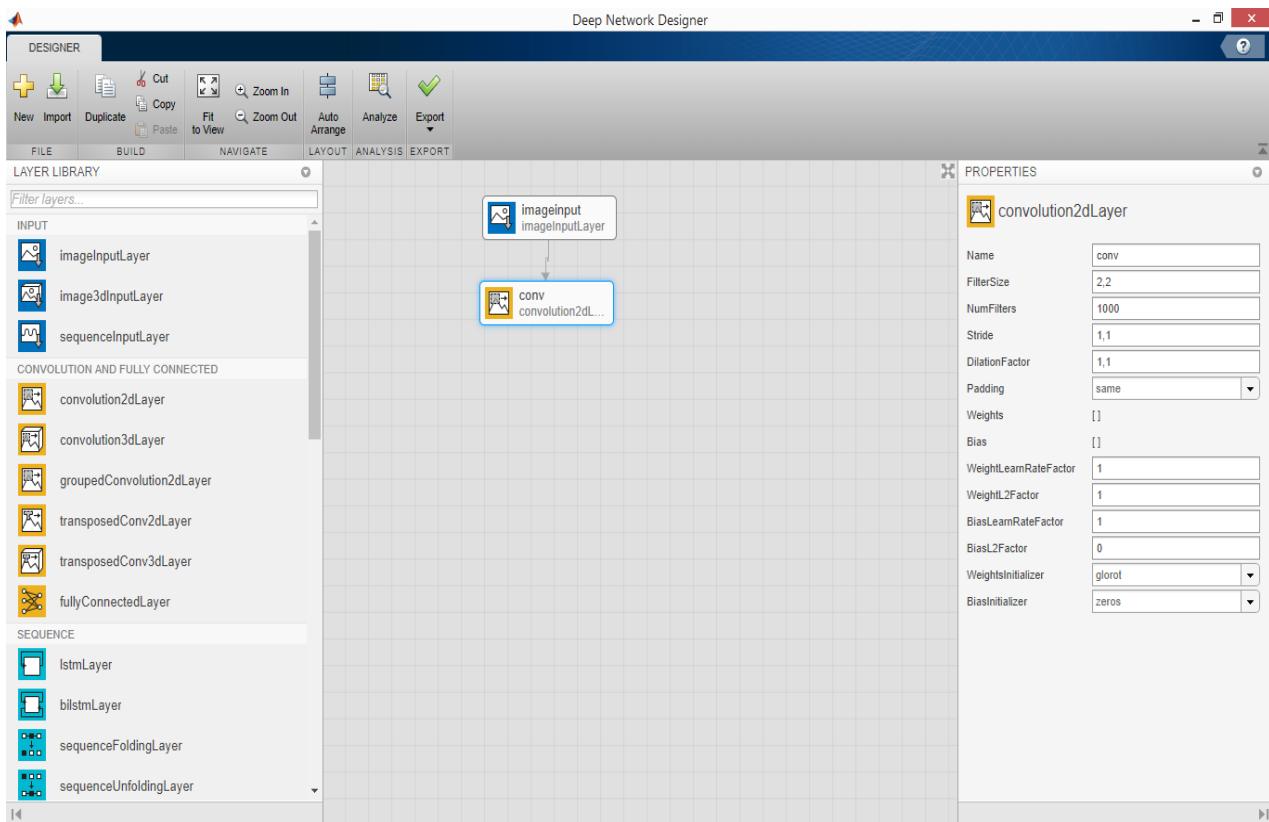


Fig 4.8 The Convolutional Layer

Source: MATLAB

We then specify the number of filters for our network which is the number of input images (The training data). We have 1000 images each for the infected samples and uninfected samples which we will feed into the network. We leave the other parameters in the convolution layer as default. (Fig 4.9) below shows the process of the convolution layer, where the circled * mark denotes the convolution operation, and the φ mark is the activation function. The square grayscale icons between these operators indicate the convolution filters. The convolution layer generates the same number of feature maps as the convolution filters. Therefore, for instance, if the convolution layer contains four filters, it will generate four feature maps.

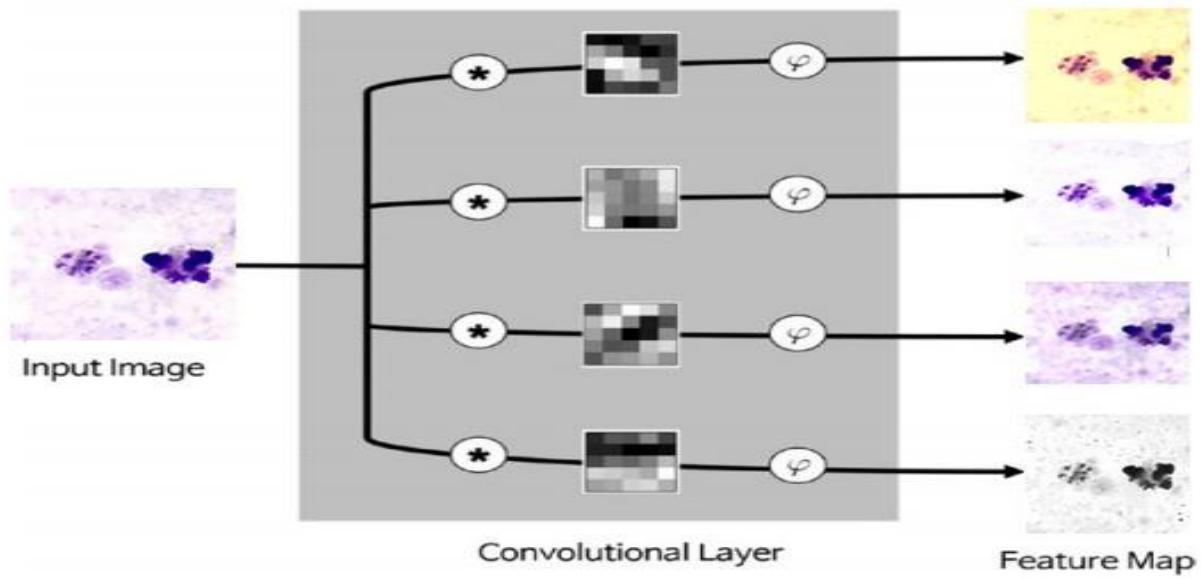


Fig 4.9 In-depth illustration of the convolutional layer
Source: My computer/Photoshop Express

4.4.3 Rectified Linear Unit (ReLU Function)

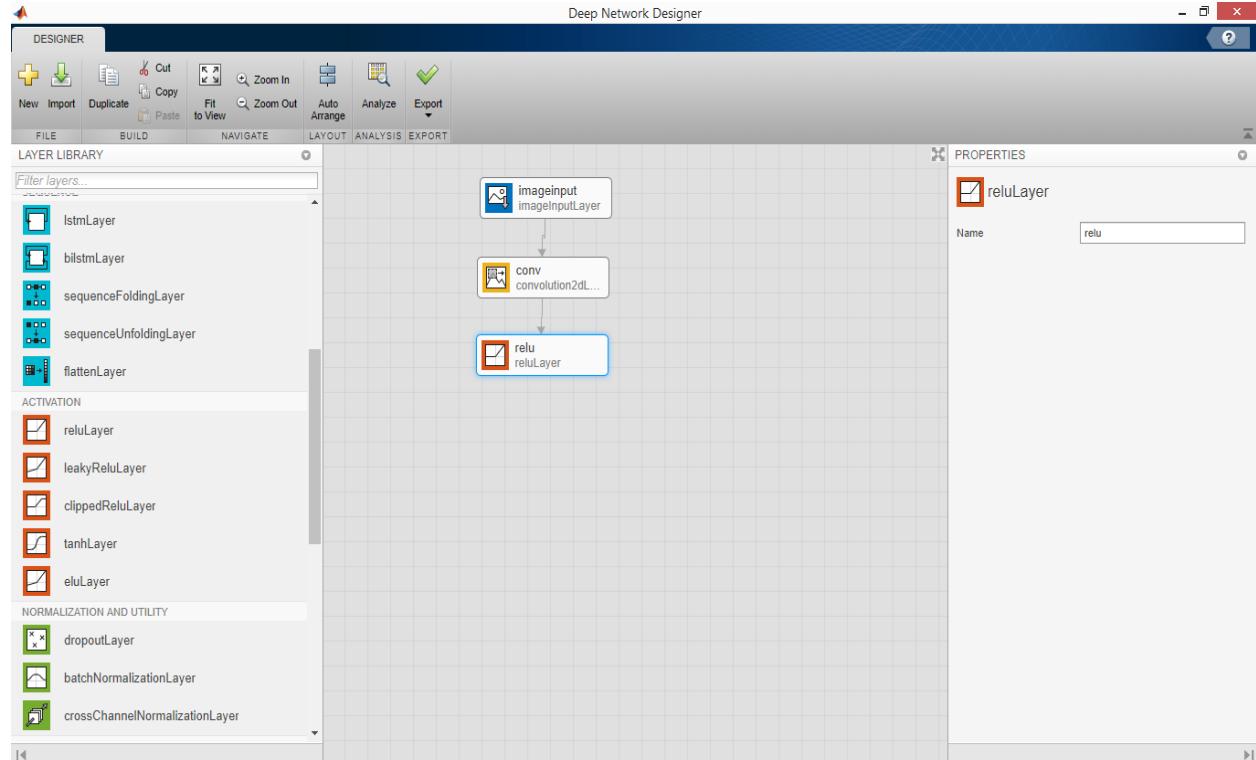


Fig 4.10 The ReLU layer
Source: MATLAB

The Rectified Linear Unit, or ReLU, is not a separate component of the convolutional neural networks' process. It's a supplementary step to the convolution layer that comes ahead of it.

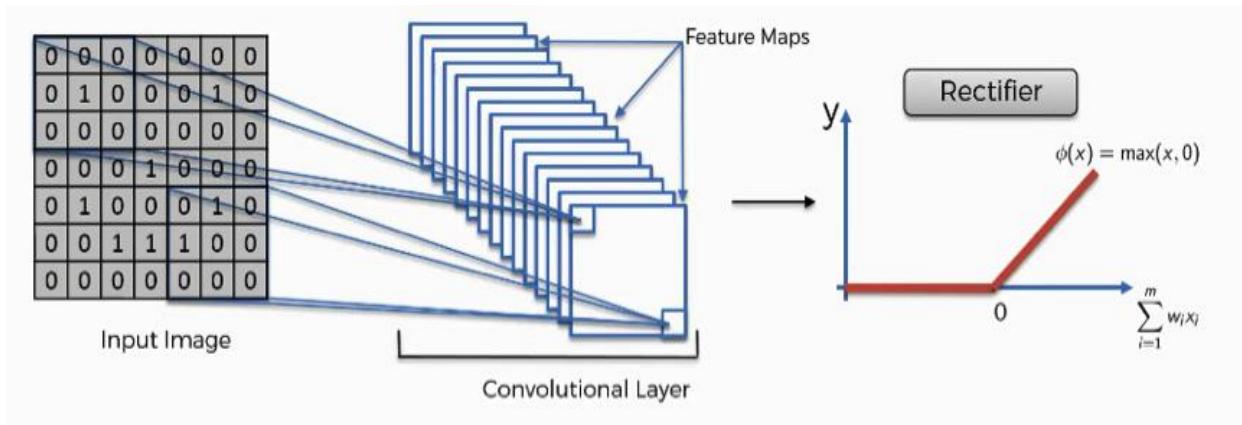


Fig 4.11 In-depth illustration of the ReLU layer

Source: My computer/Photoshop Express

The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do that is that images are naturally non-linear. When you look at any image, you'll find it contains a lot of non-linear features (e.g. the transition between pixels, the borders, the colors, etc.). The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose on an image when we put it through the convolution operation.

The ReLU function also fixes the problem of vanishing gradient. The representative solution to the vanishing gradient is the use of the Rectified Linear Unit (ReLU) function as the activation function. It is known to better transmit the error than the sigmoid function. Fig 4.12 depicts the ReLU function. It produces zero for negative inputs and conveys the input for positive inputs. Its implementation is extremely easy as well.

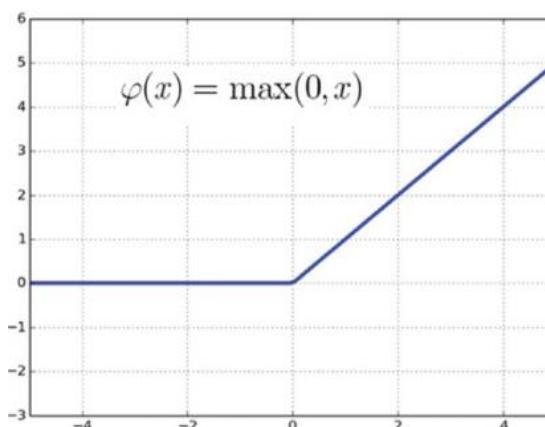


Fig 4.12 The ReLU function

Source: www.towardsdatascience.com/ReLU_activation/note.html

4.4.4 Batch Normalization

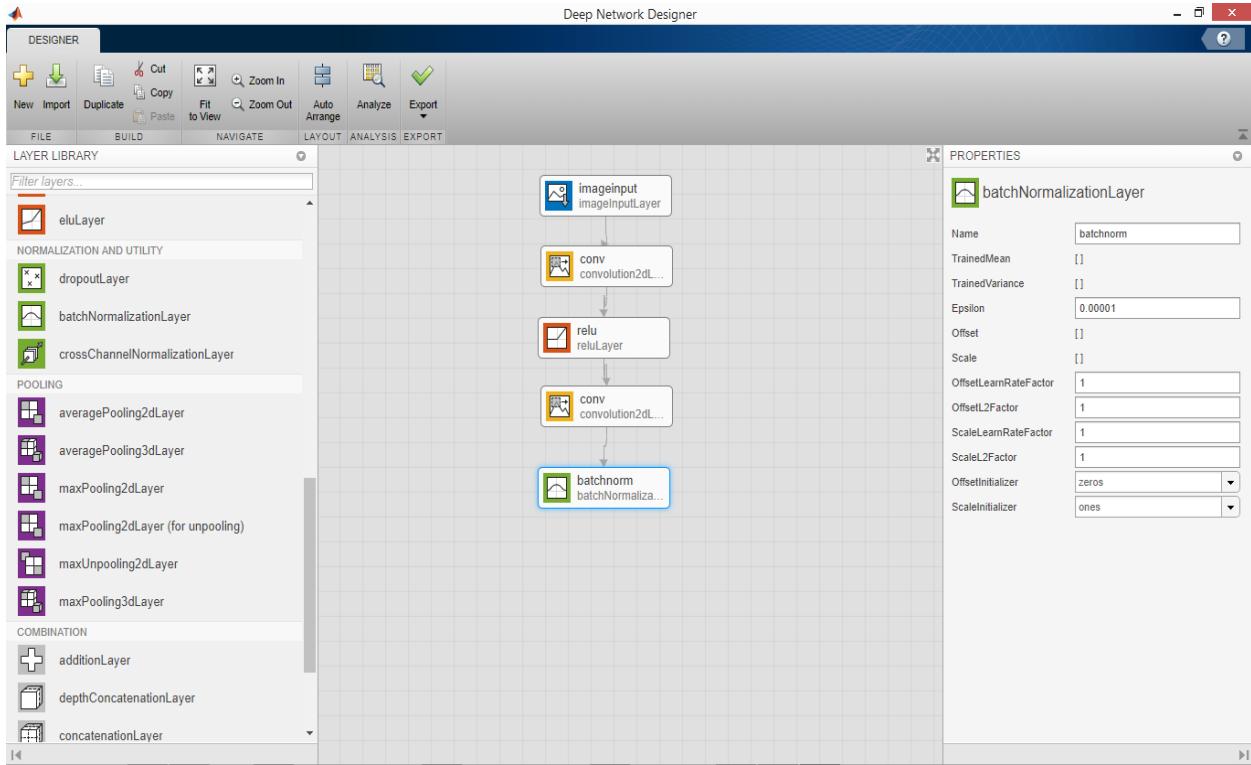


Fig 4.13 Batch Normalization Layer

Source: MATLAB

We include another convolution layer and then a batch normalization layer. Batch normalization is a technique for improving the speed, performance, and stability of artificial neural networks. Batch normalization was introduced in the year 2015. It is used to normalize the input layer by adjusting and scaling the activations. Batch normalization was initially proposed to solve internal covariate shift. During the training stage of networks, as the parameters of the preceding layers change, the distribution of inputs to the current layer changes accordingly, such that the current layer needs to constantly readjust to new distributions. This problem is especially severe for deep networks, because small changes in shallower hidden layers will be amplified as they propagate within the network, resulting in significant shift in deeper hidden layers.

Therefore, the method of batch normalization is proposed to reduce these unwanted shifts to speed up training and to produce more reliable models. Besides reducing internal covariate shift, batch normalization introduces many other benefits. With this additional layer, the network can use higher learning rate without vanishing or exploding gradients. Furthermore, batch normalization

regularizes the network such that it is easier to generalize, and it is thus unnecessary to use dropout to mitigate overfitting. The network also becomes more robust to different initialization schemes and learning rates.

4.4.5 Pooling Layer

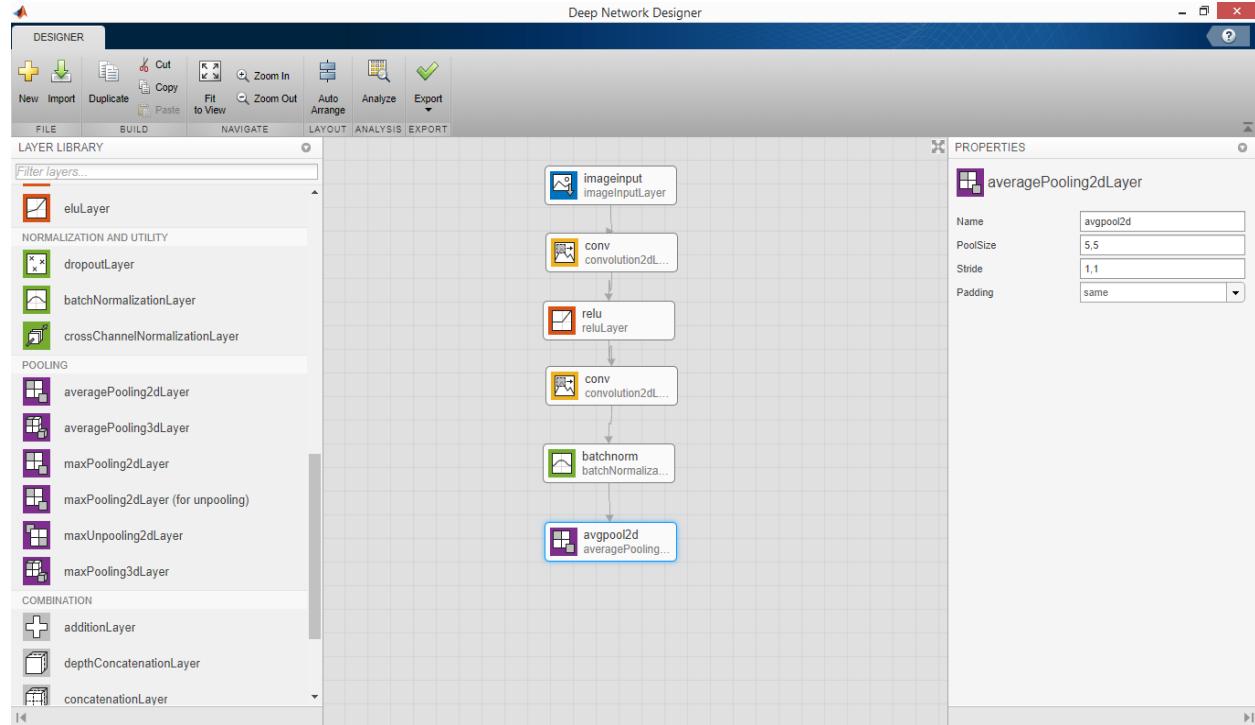


Fig 4.14 The Pooling Layer

Source: MATLAB

The pooling layer reduces the size of the image, as it combines neighboring pixels of a certain area of the image into a single representative value. Pooling is a typical technique that many other image processing schemes have already been employing. In order to conduct the operations in the pooling layer, we should determine how to select the pooling pixels from the image and how to set the representative value. The neighboring pixels are usually selected from the square matrix, and the number of pixels that are combined differs from problem to problem. The representative value is usually set as the mean or maximum of the selected pixels. Pooling layer is used to reduce the spatial volume of the input image after convolution. It is used between two convolution layers. If we apply the fully connected layer after the Convolution layer without applying pooling, then it will be computationally expensive and take a longer amount of time to train.

There are several ways to condense the information, but a usual one, which I will be using for this project, is known as average-pooling. In average-pooling, each group of entry points is transformed into the average value of the group of points instead of its maximum value. The operation of the pooling layer is surprisingly simple. As it is a two-dimensional operation, and an explanation in text may lead to more confusion, let's go through an example.

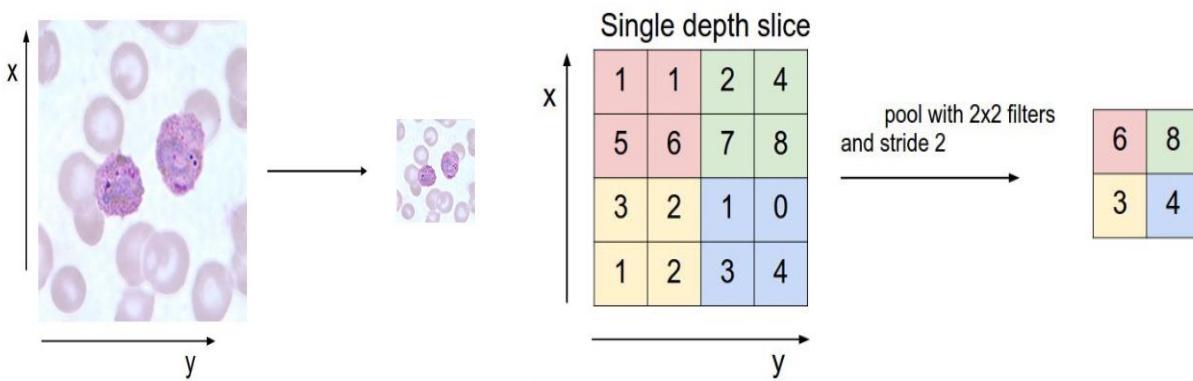


Fig 4.15 In-depth illustration of the pooling layer
Source: My computer/Photoshop Express

We combine the pixels of the input image into a 2×2 matrix without overlapping the elements. Once the input image passes through the pooling layer, it shrinks into a 2×2 -pixel image. The pooling layer compensates for eccentric and tilted objects to some extent. For example, the pooling layer can improve the recognition of a malaria parasite, which may be distorted or off-center in the input image. In addition, as the pooling process reduces the image size, it is highly beneficial for relieving the computational load and preventing overfitting.

4.4.6 The Fully-connected Layer

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training. It comes before the Softmax layer which is used for multi-classification and the Output layer which contains the labels. The Output layer is at the end of the network. The fully connected (FC) layer in the CNN represents the feature vector for the input. This feature vector/tensor/layer holds information that is vital to the input. When the network gets trained, this feature vector is then

further use for the classification. During training, this feature vector is being used to determine the loss, and helps the network to train.

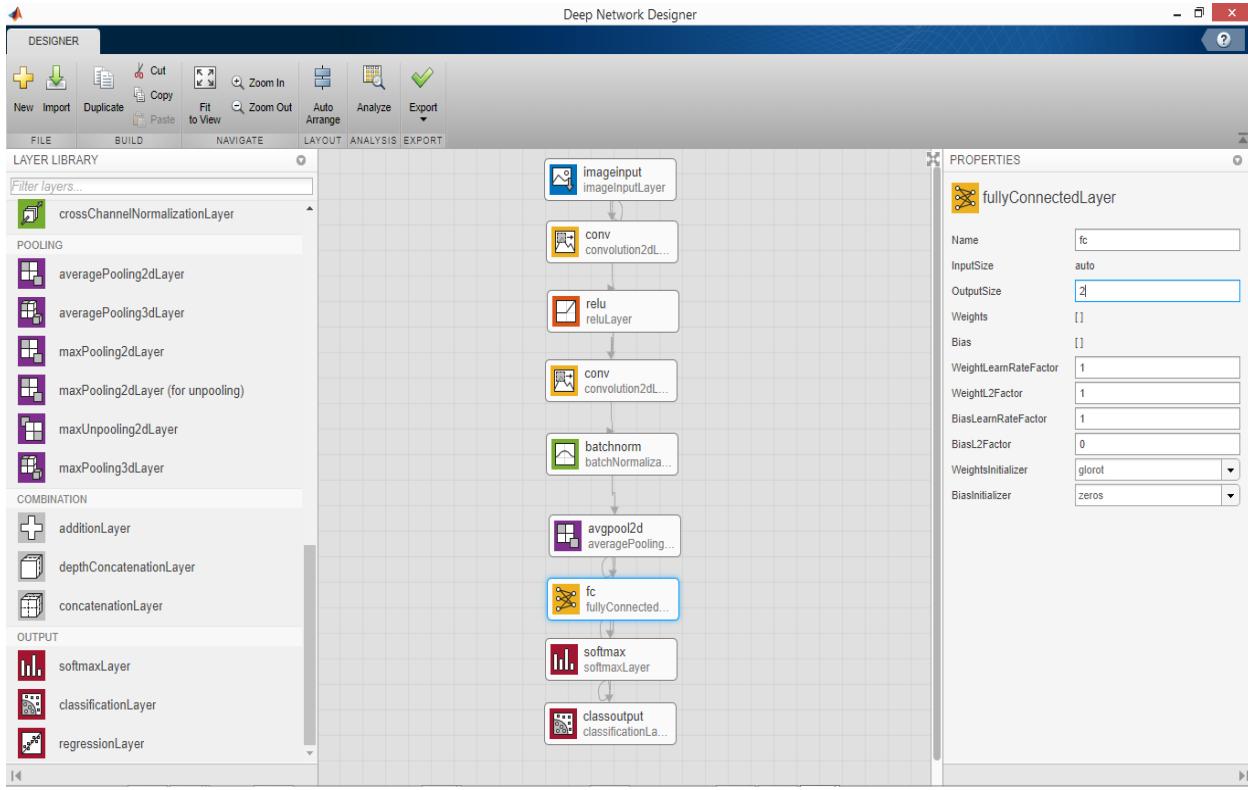


Fig 4.16 The Fully-connected Layer
Source: MATLAB

The convolution layers before the fully-connected layer hold information regarding local features in the input image such as edges, blobs, shapes, etc. Each convolution layer holds several filters that represent one of the local features. The fully-connected layer holds composite and aggregated information from all the convolution layers that matters the most.

4.5 The Training Phase

We begin the training phase by analyzing the convolutional neural network for errors and then exporting it into the MATLAB workspace for training. Using the network analysis tool, we get a full, detailed and concise view of the neural network and the component layers as well any alerts for warnings and errors.

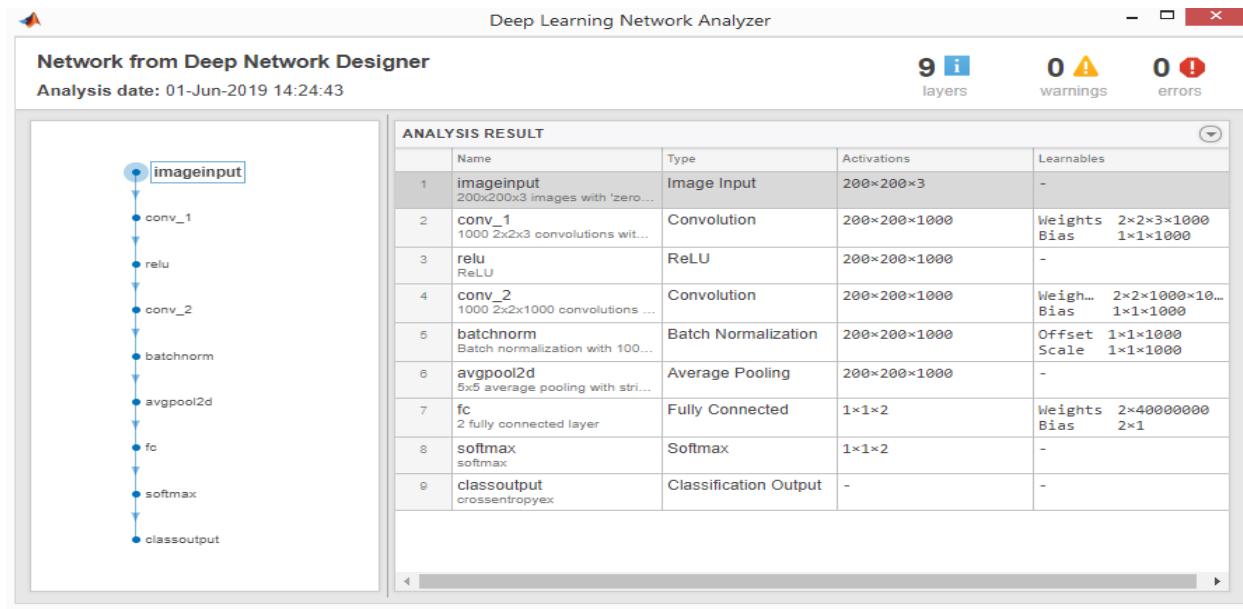


Fig 4.17 The Deep Learning Network Analyzer

Source: MATLAB

We can then export the network and proceed with the training phase using the MATLAB code shown in the figure below:

```

%Load the input images for training from the datastore
imagepath = fullfile('myImages');
imds = imageDatastore(imagepath, 'IncludeSubfolders',true,'LabelSource','FolderNames');

%Split inputs into training and validation sets (70% and 30% respectively)
[trainDS,validDS] = splitEachLabel(imds,0.7,0.3, 'randomized');

%Specify the training parameters like learnrate, batchsize, e.t.c.
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, ...
    'ValidationData', validDS, ...
    'Plots','training-progress',...
    'MiniBatchSize', 24, ...
    'ValidationPatience', 3, 'ExecutionEnvironment','cpu');

%Train the network
nnet = trainNetwork(trainDS, cnnmalaria, opts);

```

Fig 4.18 The MATLAB script for training the convolutional neural network

Source: MATLAB

The code on Line 11 in Fig 4.18 ('`Plots`', '`'training-progress'`, ...) calls the figure plot table to show the training process which is shown below:

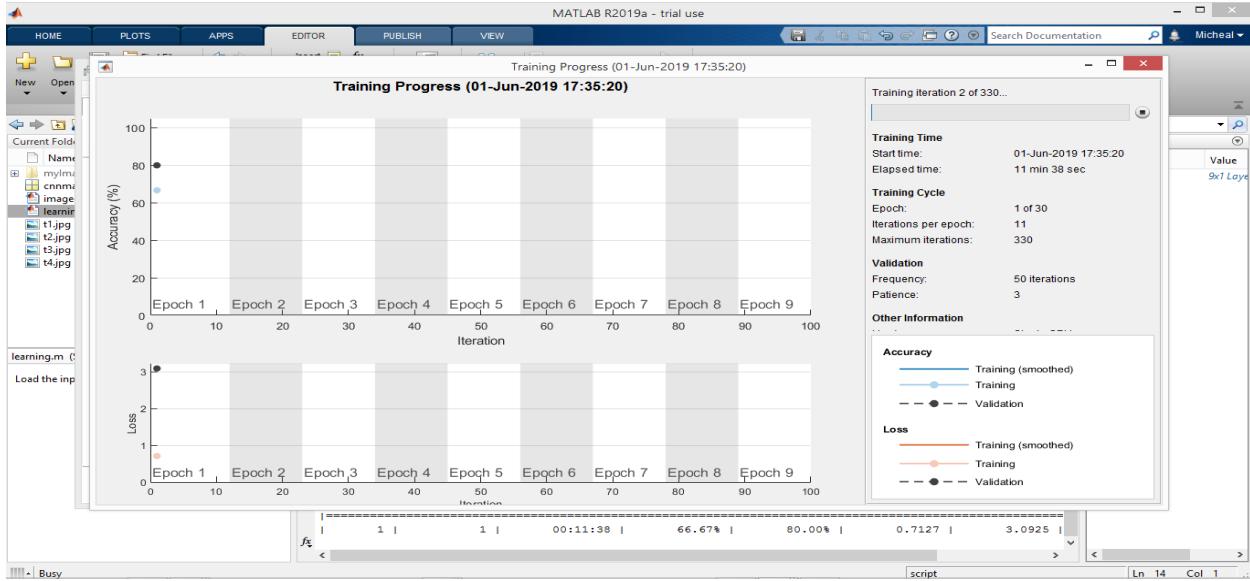


Fig 4.19 The Training Progress at Epoch 1
Source: MATLAB

As seen in Fig 4.19 above, the training and validation start at 66.67% and 80.00% respectively for the first epoch. In this network, there are 11 iterations per epoch. The training process continuously progresses and improves as shown in Fig 4.20 below. As the training continues in Fig 4.20, the loss increases temporarily for epoch 1.

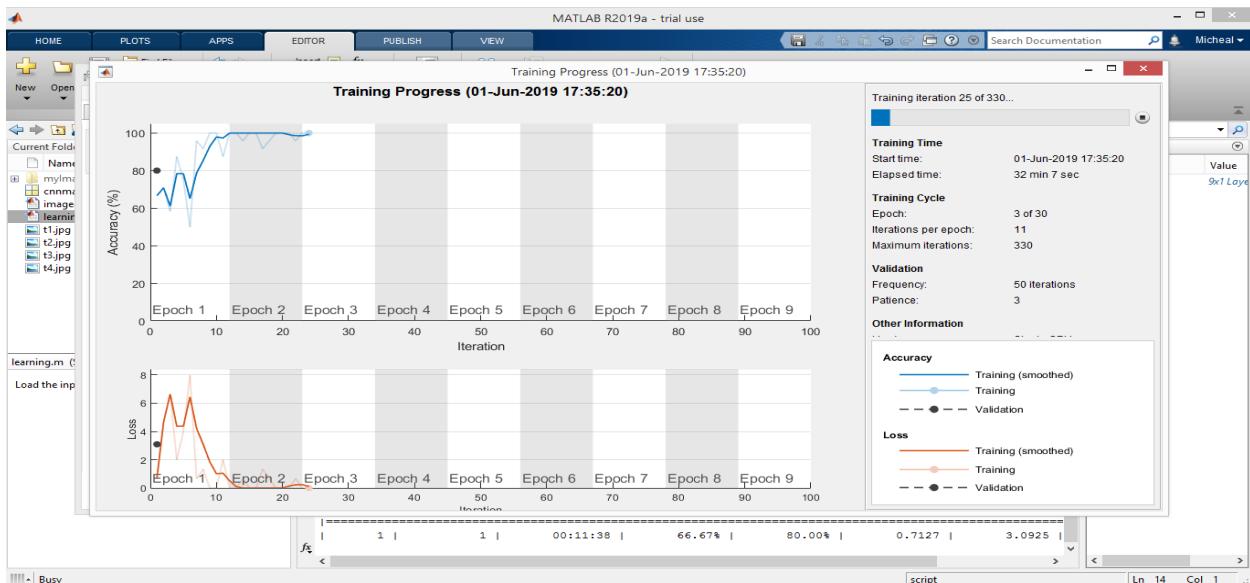


Fig 4.20 The Training Progress at Epoch 3
Source: MATLAB

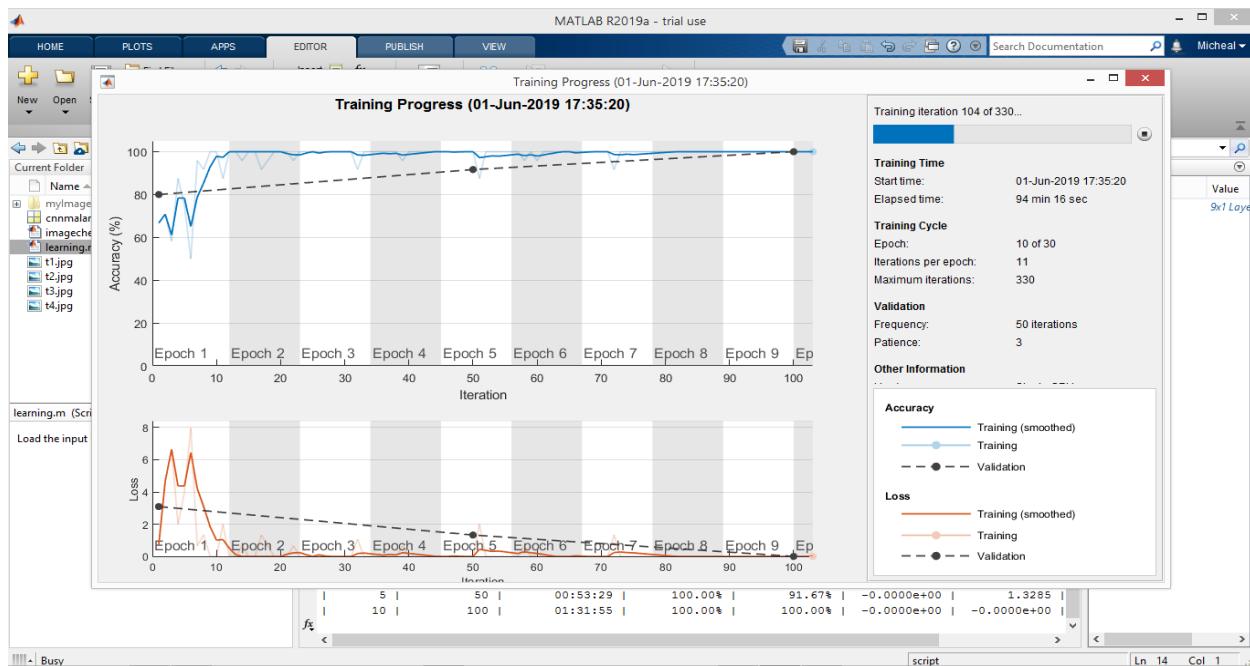


Fig 4.21 The Training Progress at Epoch 10

Source: MATLAB

In Fig 4.21 above, the network has completed 10 training epochs out of 30 with a training and validation accuracy of 100.00% for both. And as we can see in Fig 4.21 and Fig 4.22, the loss is gradually reducing. Loss is the penalty for a bad prediction i.e. loss is a number indicating how bad the model's prediction was on a single example.

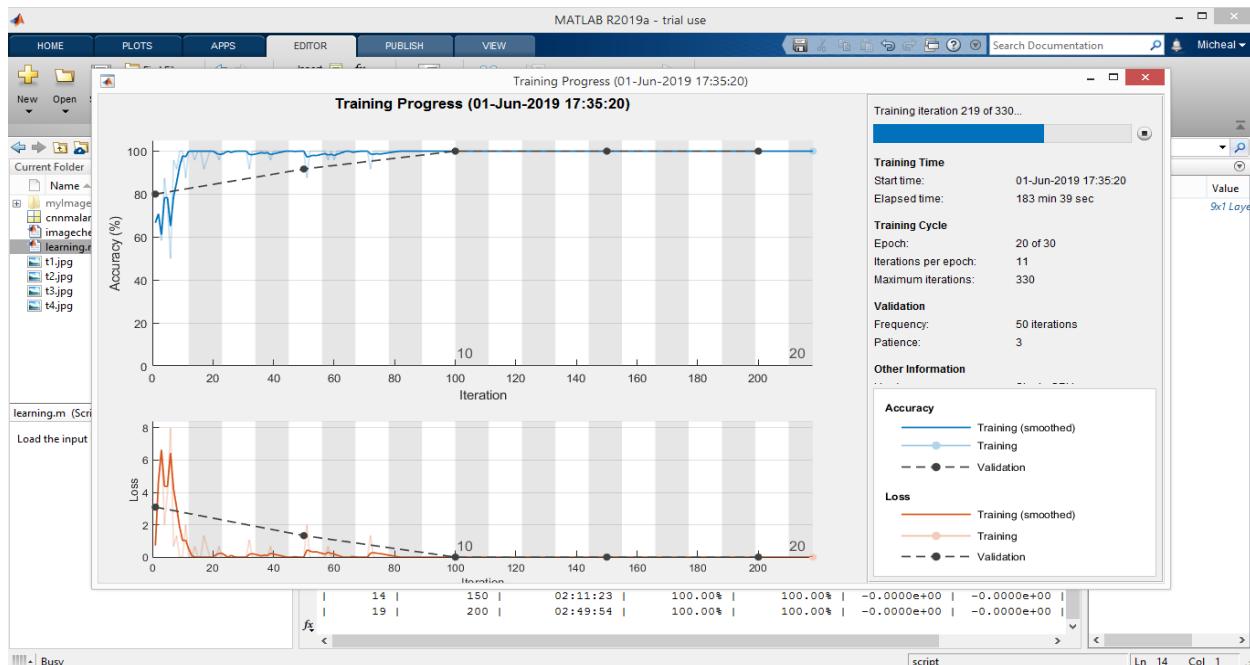


Fig 4.22 The Training Progress at Epoch 20

Source: MATLAB

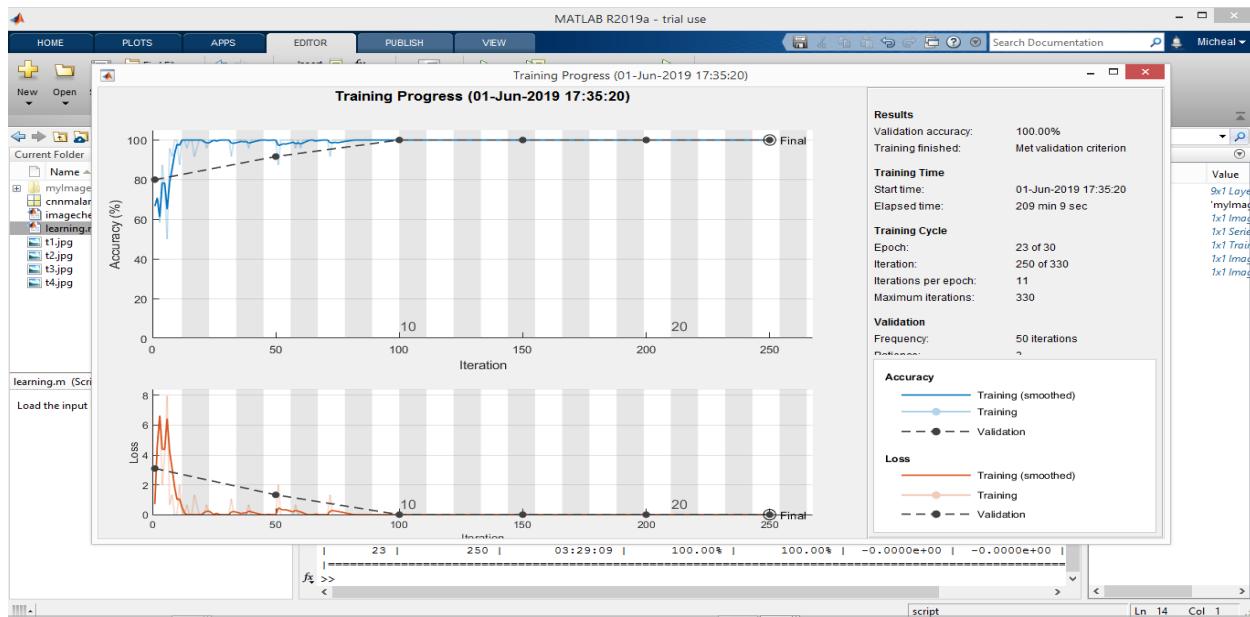


Fig 4.23 The Training Progress at the Final Epoch

Source: MATLAB

In fig 4.23 above, training has been completed and reached the final state with an overall validation accuracy of 100.00% in 30 epochs. As we can see in Fig 4.23, the training loss is zero. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples. In fig 4.24 below, the total result of the completed training process is shown in the command window.

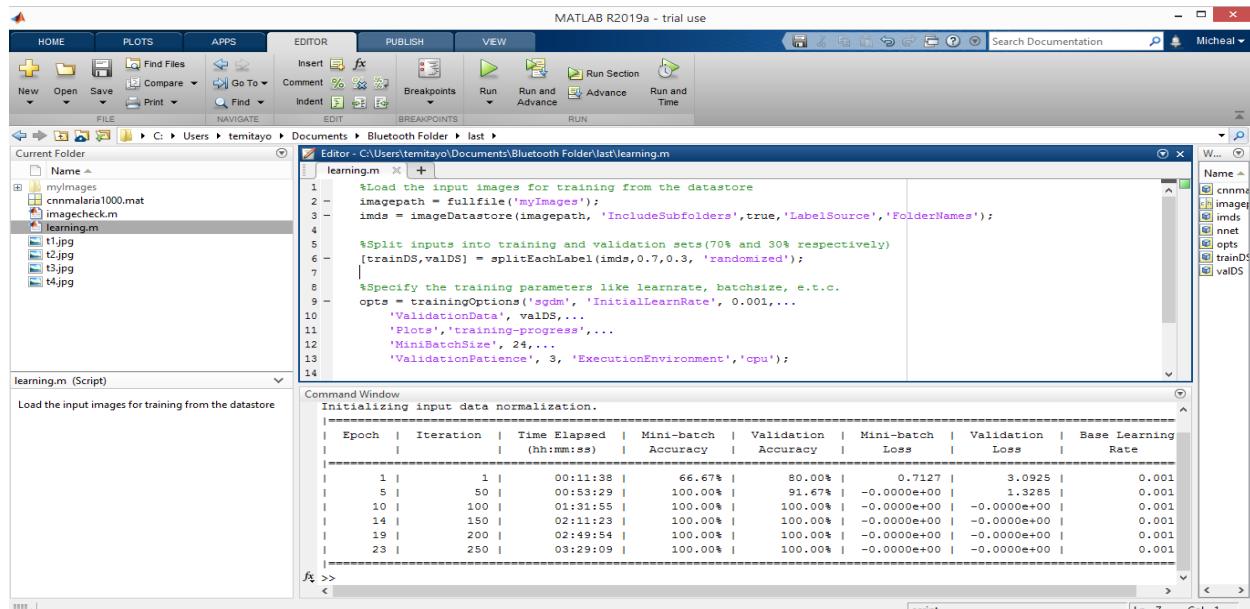


Fig 4.24 The Total Result of the Completed Training Process

Source: MATLAB

We can take this simplified illustration below to explain and explore how the training is performed, as well as the validation and how the training data is processed from the moment it is inserted into the convolutional neural network until it develops its classes (**infected**, **uninfected**).

First we have our untrained network which we just designed and specified the outputs as two classes:

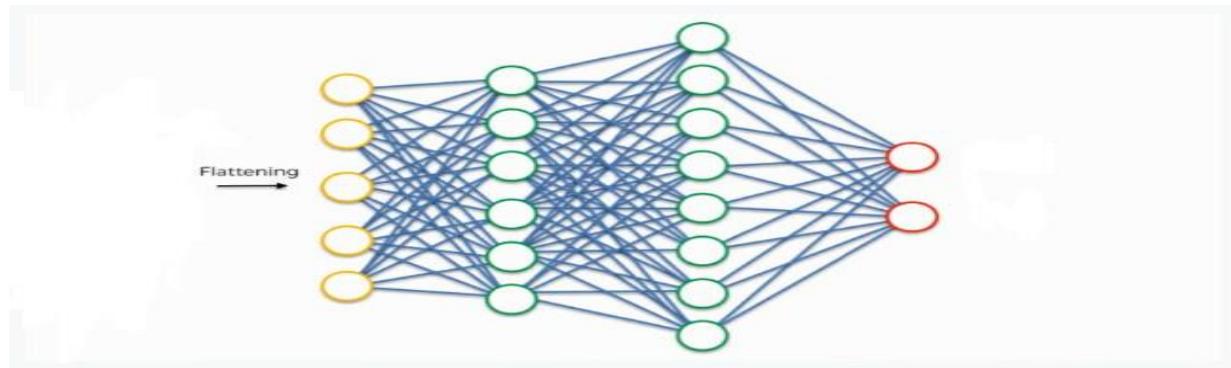


Fig 4.25 Simplified Illustration of the Training Process (1)
Source: My computer/Photoshop Express

At the very beginning, we have an input image which we convolve, pool, flatten, and then pass through the convolutional neural network. By the end of this channel, the neural network issues its predictions. Say, for instance, the network predicts the image to be infected by a probability of 80%, yet the image actually turns out to be uninfected. An error has to be calculated in this case with convolutional neural networks, it is more commonly referred to as a “loss function.” We use the cross-entropy function in order to achieve that. The loss function informs us of how accurate our network is which is shown on the figure table during the training process, which we then use in optimizing our network in order to increase its effectiveness. That requires certain things to be altered in our network. These include the weights (the blue lines connecting the neurons, which are basically the synapses), and the feature detector since the network often turns out to be looking for the wrong features and has to be reviewed multiple times for the sake of optimization. As we work to optimize the network, the information keeps flowing back and forth over and over (back propagation) until the network reaches the desired state.

During the training process, the fully-connected layer practically works as follows:

- The neuron in the fully-connected layer detects a certain feature; a purple stain (malaria parasite).

- It preserves its value.
- It communicates this value to both the “infected” and the “uninfected” classes.
- Both classes check out the feature and decide whether it's relevant to them.

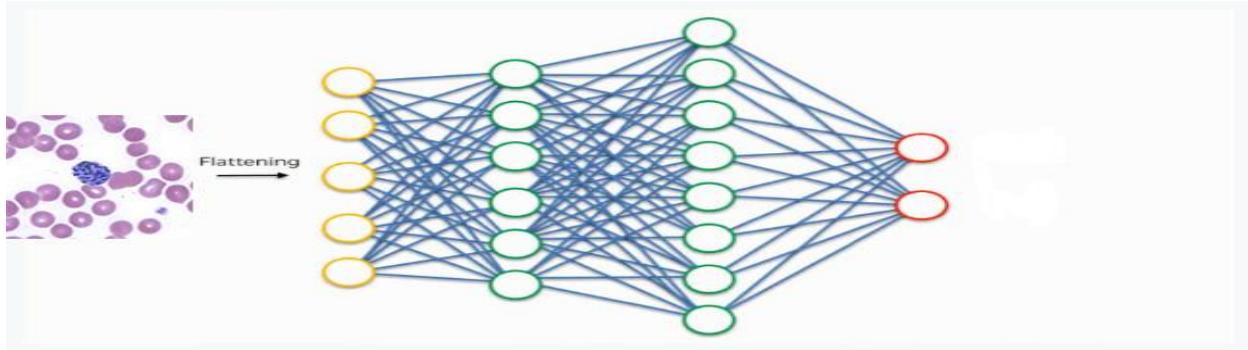


Fig 4.26 Simplified Illustration of the Training Process (2)
Source: My computer/Photoshop Express

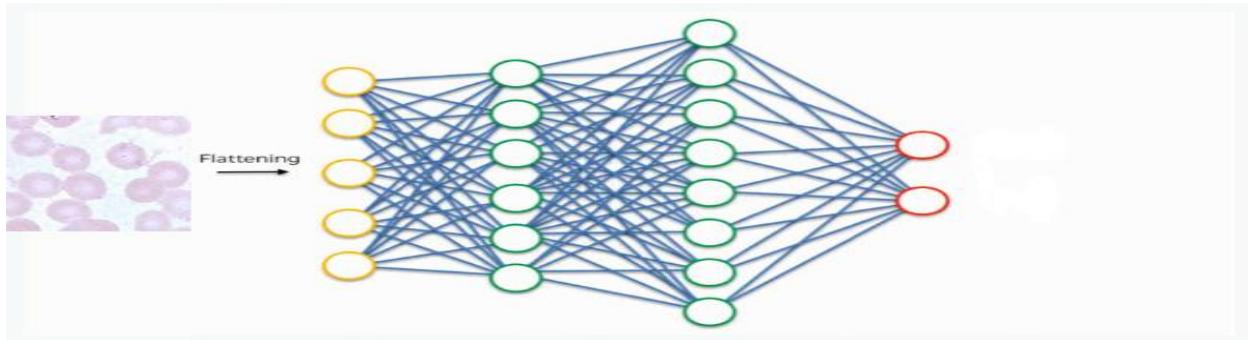


Fig 4.27 Simplified Illustration of the Training Process (3)
Source: My computer/Photoshop Express

In this network, the priority placed on the purple stain synapse is high because it's a common feature for infected images, which means that the network is confident that it is an infected image. Since the information is constantly flowing in both directions, the “uninfected” class takes note of this and understands that the purple stain is a feature of an infected image, then it simply can't belong to the uninfected class. Even if at first it would have considered the images with that feature, now it dismisses them. This explains the reason why the validation and training accuracy is very poor in the first few epochs of the training process as the network has is still considering the feature of the two classes (infected, uninfected). This happens gradually as it receives the same reading multiple times. The infected class on its part will start focusing more on the features carrying the highest priorities (the three thick purple lines in the figure below), and it will ignore the rest. The

same process simultaneously occurs with the uninfected class, enabling it to pick out its own priority features. What we end up with is what you see in the image below. As this process goes on repeat for hundreds or thousands of times, you find yourself with a well-trained and optimized convolutional neural network.

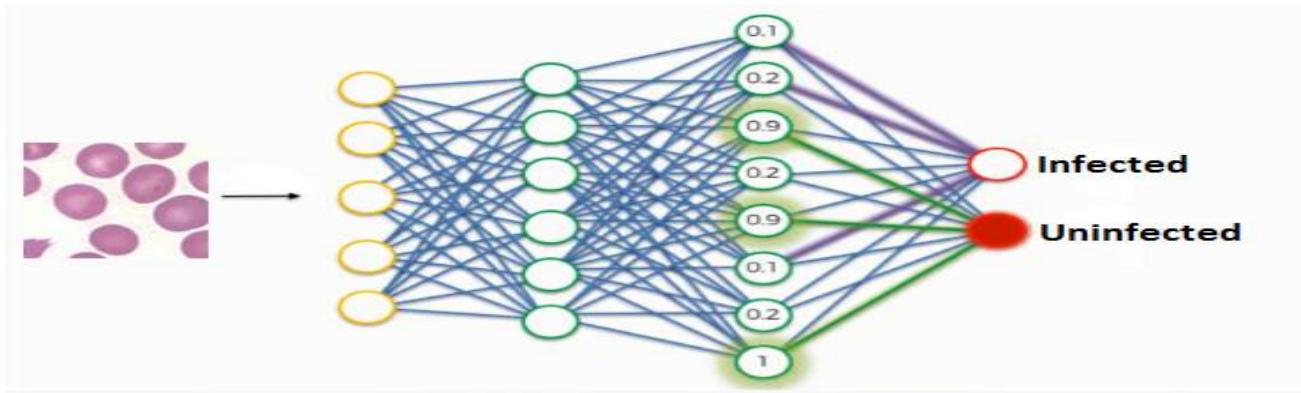


Fig 4.28 Simplified Illustration of the Training Process (4)
Source: My computer/Photoshop Express

4.6 The Implementation of the New System

The convolutional neural network was applied to four test images (Test Image 1, 2, 3 and 4). These images were chosen due to their lack of visual noise (red blood cells are evenly colored, no background artifacts) in order to carry out a baseline performance evaluation and identify strengths and weaknesses in the neural network. The MATLAB code to test the convolutional neural network on new images is given below:

```
% % %
% Create the figure window. First, resize the window to have twice the
% width, and create two subplots.
h = figure;
h.Position(3) = 2*h.Position(3);
ax1 = subplot(1,2,1);
ax2 = subplot(1,2,2);
```

```

%% %

% In the left subplot, display the image and classification together.

load cnnmalaria1000;

im = imread('t1.jpg'); %Read the image

imshow(im);

image(ax1,im)

im = imresize(im,[200,200]);

[label,score] = classify(nnet,im);

title(ax1,{['CNN prediction = ' char(label)]});

%% %

% Select the top two predictions by selecting the classes with the highest
% scores.

[~,idx] = sort(score,'descend');

idx = idx(2:-1:1);

classes = nnet.Layers(end).Classes;

classNamesTop = string(classes(idx));

scoreTop = score(idx);

%% %

% Display the top two predictions as a histogram.

bar3(ax2,scoreTop)

xlim(ax2,[0 1])

title(ax2,'Prediction Probability')

xlabel(ax2,'Probability for the two classes')

yticklabels(ax2,classNamesTop)

ax2.YAxisLocation = 'right';

```



Fig 4.29 Test Image 1
Source: Faith Mediplex Hospital Laboratory

The convolutional neural network exhibits a good performance on test image 1 in Fig 4.29. It demonstrates the identification of the infected cell in the image of the blood sample by detecting the malaria parasite which is clearly visible right in the middle of image and the convolutional neural network classifies it accurately as infected and plots a histogram showing the prediction probability for the infected class as seen in Fig 4.30.

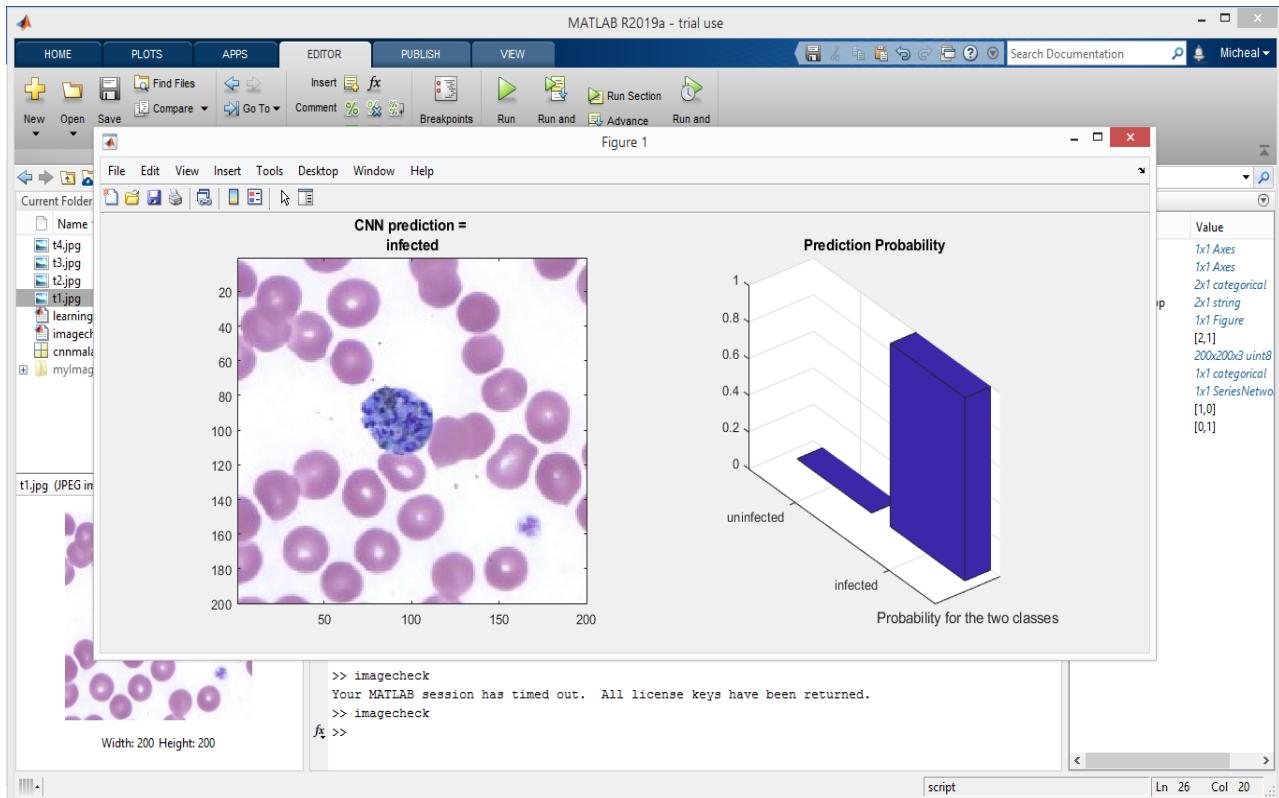


Fig 4.30 The Result of Test Image 1 using the Convolutional Neural Network
Source: MATLAB

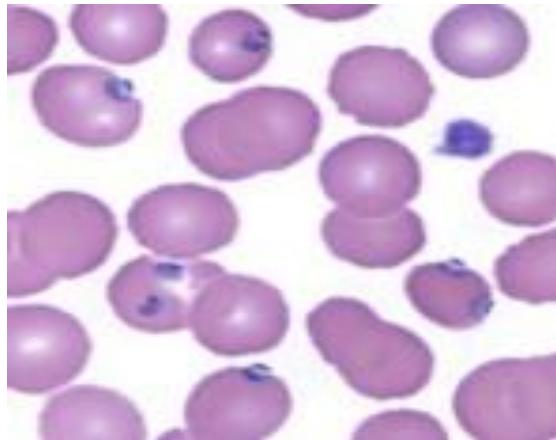


Fig 4.31 Test Image 2
Source: *Faith Mediplex Hospital Laboratory*

Test image 2 shown in Fig 4.31 is dominated by parasites in the schizont stage, which appear as round purple clusters containing many dark purple spots. The convolutional neural network also exhibits a good performance on test image 2 by identifying the infected cells in the image of the blood and classifying the image accurately as infected and plots a histogram showing the prediction probability for the infected class as seen in Fig 4.32.

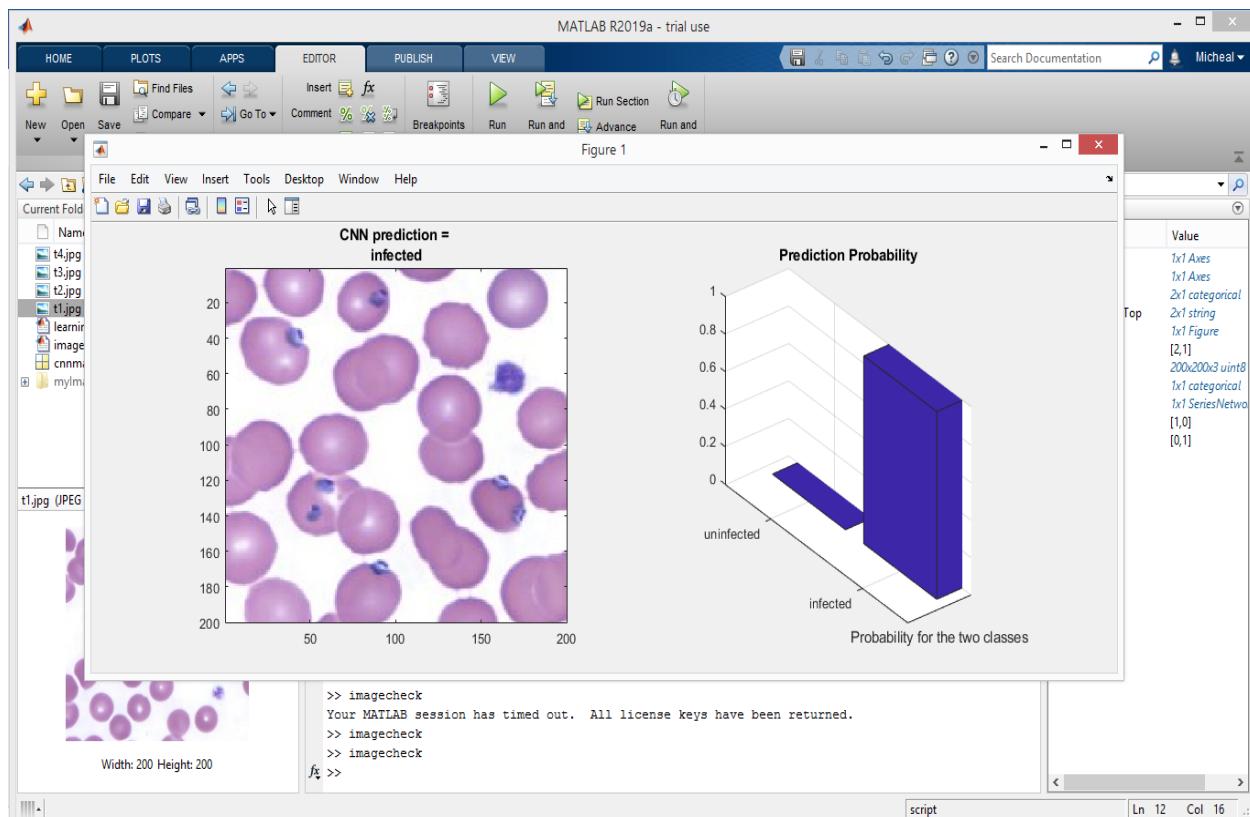


Fig 4.32 The Result of Test Image 2 using the Convolutional Neural Network
Source: *MATLAB*



Fig 4.33 Test Image 3
Source: Faith Mediplex Hospital Laboratory

In Test image 3 shown in Fig 4.33, the red blood cells are evenly colored, no background artifacts and most importantly no malaria parasites. The convolutional neural network exhibits a good performance on test image 3 by classifying it accurately as uninfected and plots a histogram showing the prediction probability for the uninfected class as seen in Fig 4.34.

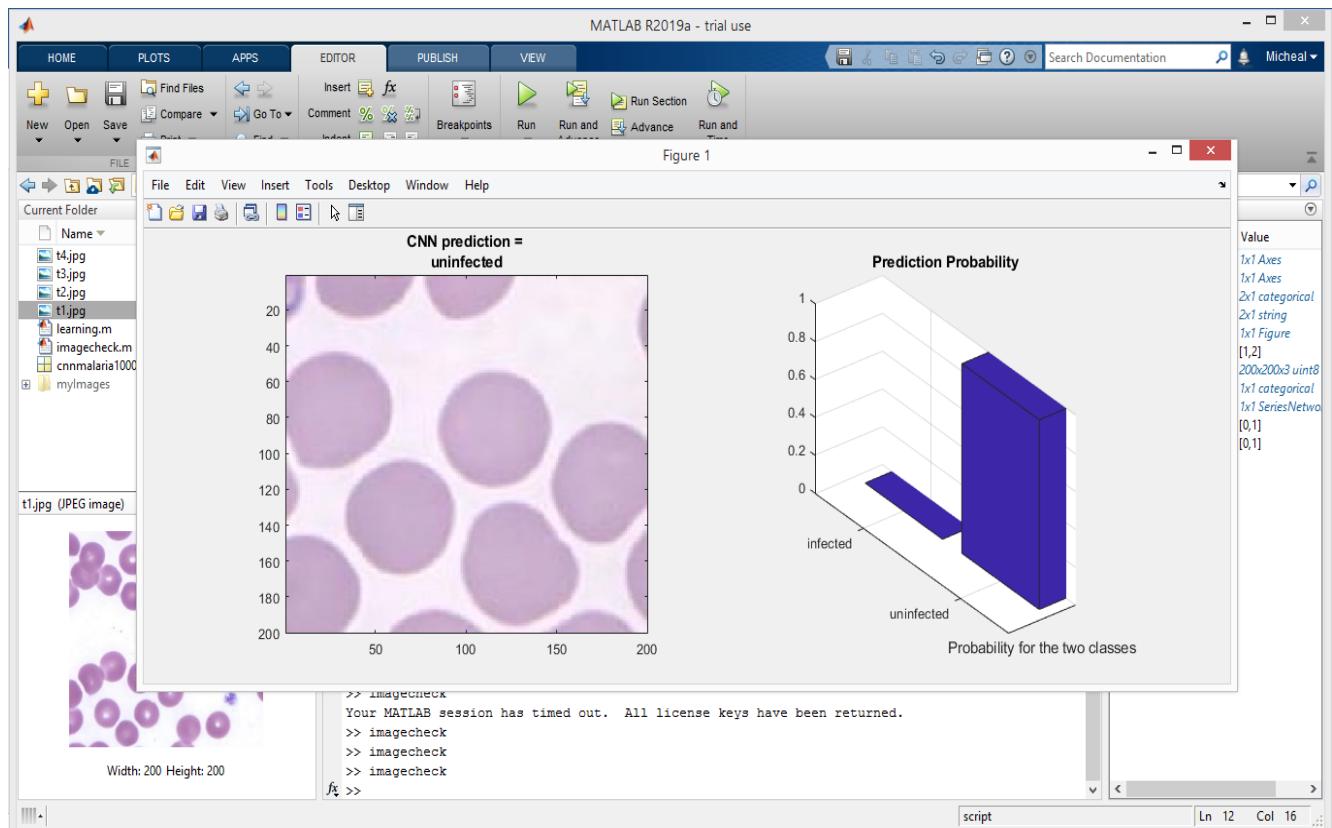


Fig 4.34 The Result of Test Image 3 using the Convolutional Neural Network
Source: MATLAB

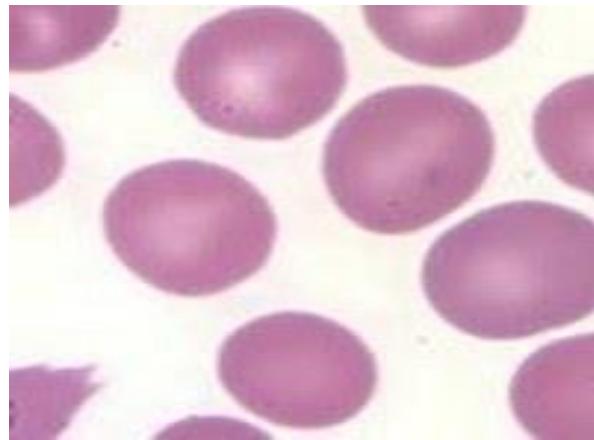


Fig 4.35 Test Image 4
Source: *Faith Mediplex Hospital Laboratory*

In Test image 4 shown in Fig 4.35 above, the red blood cells are also evenly colored, no background artifacts and most importantly no malaria parasites. The convolutional neural network exhibits a good performance on test image 4 by classifying it accurately as uninfected and plots a histogram showing the prediction probability for the uninfected class as seen in Fig 4.36 below.

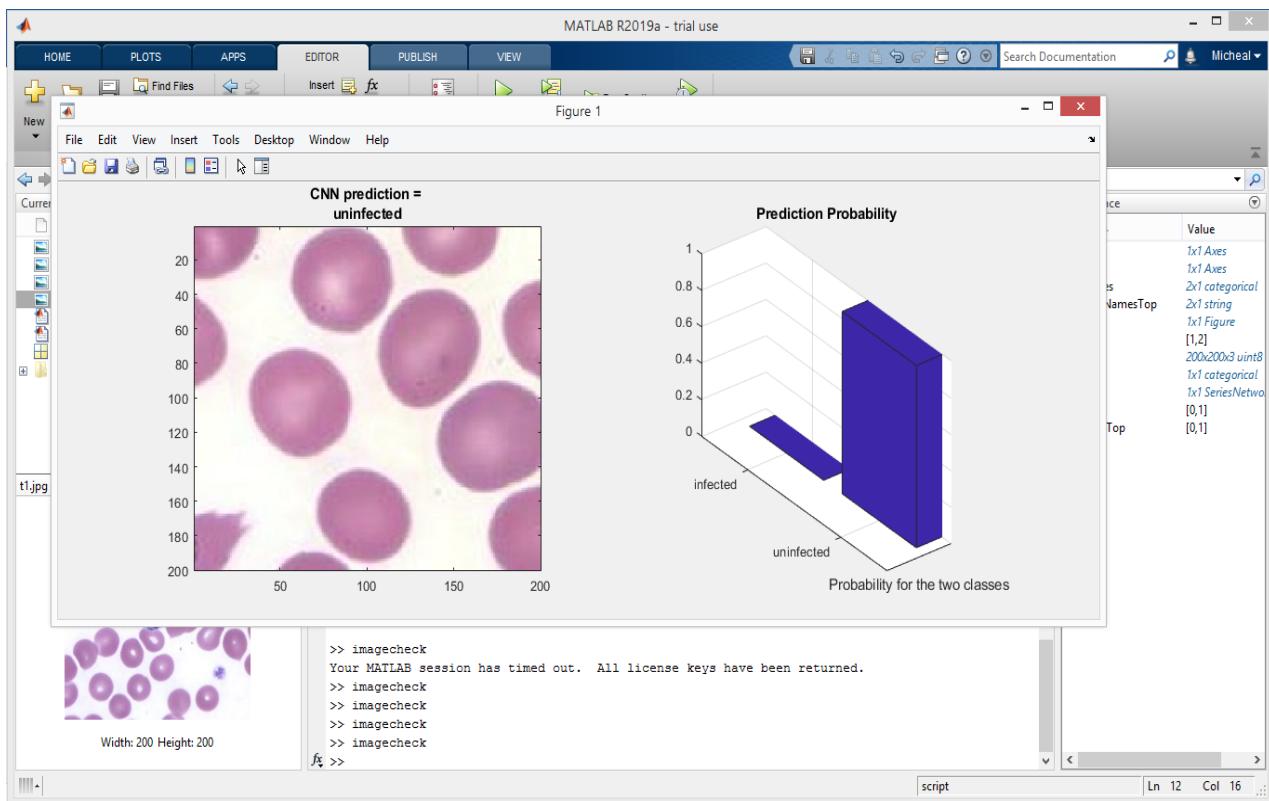


Fig 4.36 The Result of Test Image 4 using the Convolutional Neural Network
Source: *MATLAB*

4.7 Problems Encountered and Solutions

In the initial training stages of the convolutional neural network we encountered some problems which led to a very poor performance in classifying images of blood samples. The reason that the neural network yielded poorer performance was that the network was not properly trained. We experience the following three primary difficulties in the training process of the convolutional neural network:

1. Vanishing gradient: The vanishing gradient in the training process with the back-propagation algorithm occurs when the output error is more likely to fail to reach the farther nodes. There is no point of adding hidden layers if they cannot be trained. The vanishing gradient problem is greatly improved by employing the ReLU activation function.
2. Overfitting: Overfitting refers to a situation where a model learns statistical regularities specific to the training set, i.e., ends up memorizing the irrelevant noise instead of learning the signal, and, therefore, performs less well on a subsequent new dataset. The best solution for reducing overfitting is to obtain more training data which is exactly what we did. A model trained on a larger dataset typically generalizes better.
3. Computational load: The last challenge is the time required to complete the training. The more computations the neural network performs, the longer the training takes. This is relieved to a large extent by the use of a GPU or splitting the training data into sets.

4.8 Suggestions for Further Improvements

1. Preprocessing: The images used for training the neural network were chosen due to low levels of visual noise. Visual noise includes faint ‘background’ red blood cells which cannot easily be distinguished from the background, as well as dark regions within red blood cells caused by their characteristic shape. Visual noise degrades performance. It may be necessary to further characterize and mitigate this noise in order to ensure robust performance.

2. Use of a GPU (graphics processor unit): At present time, using a CPU to train a convolutional neural network with a dataset of 200 images requires anywhere from 3-5 hours until completion. The benefits of using a GPU outweighs using a basic computer CPU, as well as refactoring of the existing MATLAB training code should greatly decrease the amount of time required to train the neural network.
3. Increasing the training dataset: A larger training dataset of say 10,000 images equals a higher performance by the convolutional neural network and eliminates the risk of overfitting.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

Malaria is the leading cause of morbidity and mortality in tropical and subtropical countries. WHO estimates the number of malaria deaths at 435,000 for 2018. Machine learning has great potential to lighten the burden of malaria in temperate regions around the world where mosquitoes thrive especially in remote sub-Saharan Africa. However, the accuracy of the manual method of malaria diagnosis using microscopy depends on the human expert. However, it is prone to some shortcomings which include time consumption and excessive workload for the pathologists. We propose an automated diagnostic system that can exclude the human expert from the process or serve as an aid for an expert to lower workload and improve accuracy. The ultimate goal of this work was to develop a system for detecting malaria using microscopic images of stained blood samples.

Thus in the process of this work, an accurate, speedy and affordable system of malaria detection using stained blood smear images was developed. The method is based on supervised machine learning using deep learning algorithms which involves training a convolutional neural network and using it to test for the presence of malaria parasites in blood smear images. Images of infected and non-infected blood samples were given to the convolutional neural network as training data and relevant features were extracted from them and eventually further classification can be made by the network based on the features extracted from the images. The classification entailed the detection of malaria parasites. The algorithm successfully trained on 1,000 images with a training and validation accuracy of 100.00% respectively and it was tested on the images of four blood samples that it had not seen before and further classified them as infected or uninfected with a 100% accuracy. The battle against the devastating burden of malaria will continue. Early and accurate diagnosis is one of the keystones in the fight against this. This project work can be described as a required component in the development of a tool that will bring a healthy and malaria-free world one step closer to reality.

5.5 Conclusion

The detection of Malaria parasites is done by pathologists manually using Microscopes. So, the chances of false detection due to human error is present, which in turn can result into fatal conditions. This work curbs the human error while detecting the presence of malaria parasites in the blood sample by using a convolutional neural network. The system is well trained in a robust manner and achieved a high percentage of accurate prediction and no false-positives as at this time. This goes to show that neural networks can learn how to process different staining and lighting variations if only enough training data are being presented to the network.

5.4 Recommendations

A convolutional neural network for detecting malarial parasites in images of blood samples is developed, and its capabilities and limitations are characterized. Although the neural network exhibits basic functionality and a high-level of accuracy on test images, further training is necessary in order to improve robustness to the point where it can be usefully applied to a wide variety of images and then deployed for commercial use in hospitals, clinics, etc. A number of areas of potential improvement were identified, and will provide guidance as development continues. The current system provides a useful framework which can be refined and extended in order to improve accuracy, tolerance of image noise, and further capabilities.

REFERENCES

- Marco, H., Pelly, H., and Nick, T., (2016): Introduction to Machine Learning. Accessed 25/05/19. (March 2016) http://www.sciencehq.com/computing_technology/introduction-to-machine-learning.html
- Rouse, W., (2018): Machine Learning for Dummies. Accessed 23/05/19. (October 2018) <https://becominghuman.ai/Fmachine-learning-for-dummies-explained-in-2-mins.html>
- Ponce, K., (2005): Machine Learning and Artificial Intelligence for Everyone. Accessed 20/05/19. (July 2005) https://vas3k.com/blog/machine_learning.html
- Tandara, A., and Barwick, E., (2014): Tutorial on Artificial Intelligence and Its evolution, University of Calgary, October 2014, Volume 3, Issue 2, Pages 17-21, Source type: Scholarly Journals Language of publication. Copyright American Centre of Science and Education (October 2014).
- Bashier, B., (2016): Pattern recognition in neural networks and Cognitive science. In K. Dew & A. Matheson (Eds.), Understanding cognitive equalities in Aotearoa New Zealand (pp. 97–106). Dunedin, New Zealand: Otago University Press.
- Weizenbaum, G., (2006): Is Machine Learning Fake News? Accessed 24/05/19. (December 2006) <https://blog.netapp/blogs/is-machine-learning-fake-news.html>
- Gary, R., (2011): Adaptation in Neural Networks and Artificial Systems. Accessed 19/05/19. (August 2011) <https://pythontips.com/introduction-to-machine-learning-and-its-usage-in-remote-sensing.html>
- Durant, W., (1929). The Pleasures of Philosophy (Rev ed.). “Is Man a Machine?”, (pp 363–367) ISBN-13: 978-0671581107. Touchstone; Revised edition (December 31, 1999).

Khan, R., de Luca, R., & Li, J. (2016). The Machine Learning Pipeline: A New Zealand action research project. *Studies in Higher Education*, 40(5), 933–944. <https://doi.org/10.1080/03075079.2014.html>

Asahd, S., & Mohssen, C., (2018). Statistical Learning Methods as the Backbone of Intelligent Software: An Observational Study of Machine Learning. *Qualitative Science Research*, 17, (pp 1442–1443). <https://techxplore.com/infusing-machine-inductive-biases-capture.html>

Junaid, A., (2011): Computational Architecture and Algorithms used in Machine Learning. Accessed 28/05/19. (March 2011) <https://www.drift.com/blog/machinelearning-explained.html>

Bilyk, T., (2012): Supervised Machine Learning in Artificial Intelligence. Accessed 02/06/19. (April 2012) <https://www.total.com/machine-learning/machine-learning-theory-an-introductory-primer.html>

Ruchibal, D., (2014): The Convolution Theory and Its application to Images. Accessed 23/05.19. (July 2014) <https://towardsdatascience.com/convolutional+neural+networks=convolutional+neural+networks.html>

Mohammad, H., (2017): Machine Learning in the Healthcare Industry. Accessed 01/06/19. (May 2017) <https://www.healthcatalyst.com/clinical-applications-of-machine-learning-in-healthcare.html>

Robertson, A., (2009): The Malaria Transmission Cycle. Accessed 28/05/19. (April 2009) <https://www.newsnetwork.mayoclinic.org/mayo-clinic-q-and-a-malaria-continues-to-be-a-significant-travel-related-disease.html>

Morgan, H.-H., Bock, G.-W., & Vathanophas, V. (2010). Ending Malaria: A Race against the Clock. *The Study of Parasite Behavior*, 27(1), (pp 3–4). Accessed 25/05/19 (May 2010) <https://www.thejakartapost.com/ending-malaria-a-race-against-the-clock.html>

Juri, R., (2011): Why is Malaria Such a Difficult Disease to Control? Accessed 19/05/19. (August 2011) <https://www.justscience.in/articles/malaria-difficult-disease-eliminate.html>

Cabezos, N., (2012): Giemsa Staining Solution in the Effective Diagnosis of Malaria. Accessed 22/05/19. (January 2012) <https://www.plataformalac.org/malaria-cases-in-the-americas-on-the-rise.html>

Scribd, P., (2010): Clinical Findings in a Malaria Diagnosis Report. Accessed 19/05/19. (June 2010) <https://www.alison.com/course/global-health-initiative-malaria-awareness.html>

Junhao, O., (2016): A Beginner's Guide to Understanding Convolutional Neural Networks. Accessed 29/05/19. (March 2016) <https://www.Adeshpande/github.io/Beginners-Guide-To-Understanding-Convolutional-Neural-Networks.html>

Appendix A: Project Source Code

The Training Code:

```
%Load the input images for training from the datastore  
  
imagepath = fullfile('myImages');  
  
imds = imageDatastore(imagepath, 'IncludeSubfolders',true,'LabelSource','FolderNames');  
  
%Split inputs into training and validation sets(70% and 30% respectively)  
  
[trainDS,valDS] = splitEachLabel(imds,0.7,0.3, 'randomized');  
  
%Specify the training parameters like learnrate, batchsize, e.t.c.  
  
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001,...  
    'ValidationData', valDS,...  
    'Plots','training-progress',...  
    'MiniBatchSize', 24,...  
    'ValidationPatience', 3, 'ExecutionEnvironment','cpu');  
  
%Train the network  
  
nnet = trainNetwork(trainDS, cnnmalaria, opts);
```

The Implementation Code:

```
%%%  
  
% Create the figure window. First, resize the window to have twice the  
% width, and create two subplots.  
  
h = figure;  
  
h.Position(3) = 2*h.Position(3);
```

```

ax1 = subplot(1,2,1);

ax2 = subplot(1,2,2);

%% %

% In the left subplot, display the image and classification together.

load cnnmalaria1000;

im = imread('t1.jpg'); %Read the image

• imshow(im);

image(ax1,im)

im = imresize(im,[200,200]);

[label,score] = classify(nnet,im);

title(ax1,'CNN prediction = ' char(label));

%% %

% Select the top two predictions by selecting the classes with the highest

% scores.

[~,idx] = sort(score,'descend');

idx = idx(2:-1:1);

classes = nnet.Layers(end).Classes;

classNamesTop = string(classes(idx));

scoreTop = score(idx);

%% %

% Display the top two predictions as a histogram.

bar3(ax2,scoreTop)

xlim(ax2,[0 1])

```

```

title(ax2,'Prediction Probability')

xlabel(ax2,'Probability for the two classes')

yticklabels(ax2,classNamesTop)

ax2.YAxisLocation = 'right';

```

Appendix B: User Guide

Step 1: Start the MATLAB application

Step 2: Place the images of the blood samples in the MATLAB workspace

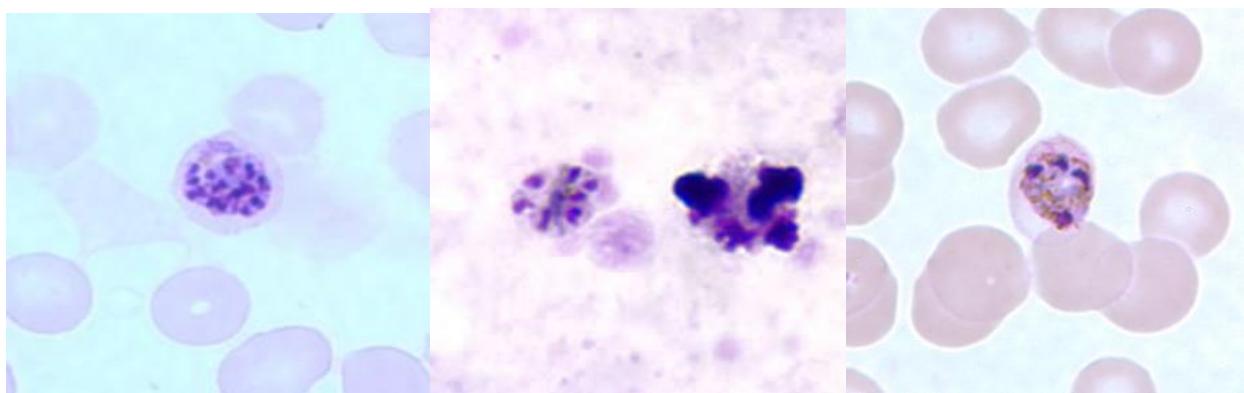
Step 3: Launch the MATLAB script to check the images of the blood samples

Step 4: The convolutional neural network issues a prediction and plots a bar chart to show the estimated prediction probability.

Step 5: The user can choose to mail, print or save the resulting output.

Appendix C: The Training Images

We trained the network with 1000 images of infected sample and non-infected samples for each class which were gotten from the Faith Mediplex Hospital. Below are some examples of the training images in Fig 4.37 and Fig 4.38.



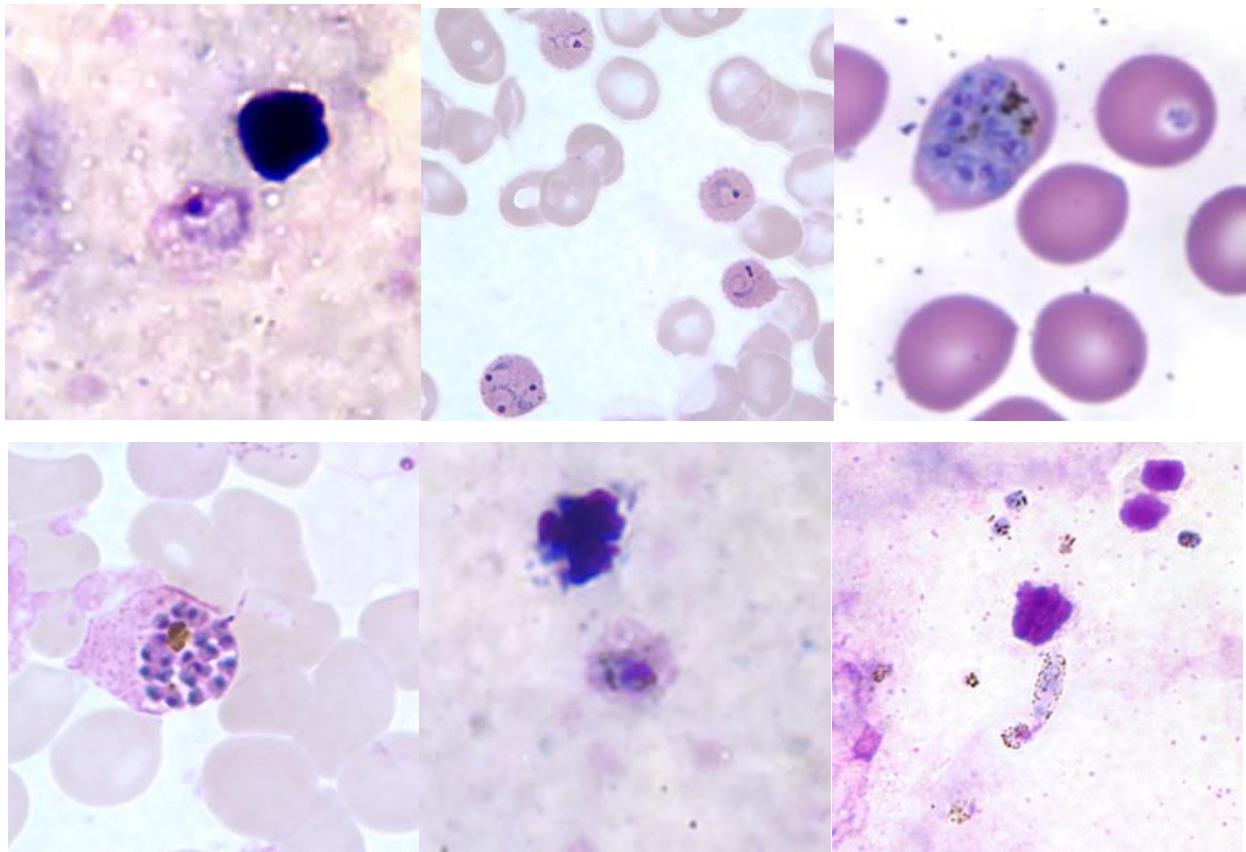
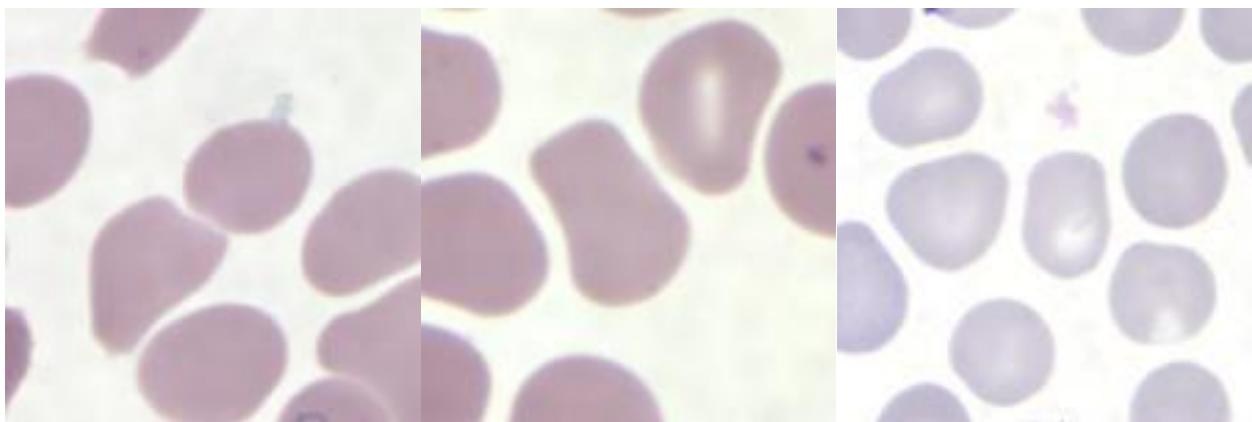


Fig 4.37 Examples of the training images for the infected class

Source: FAITH MEDIPLEX HOSPITAL



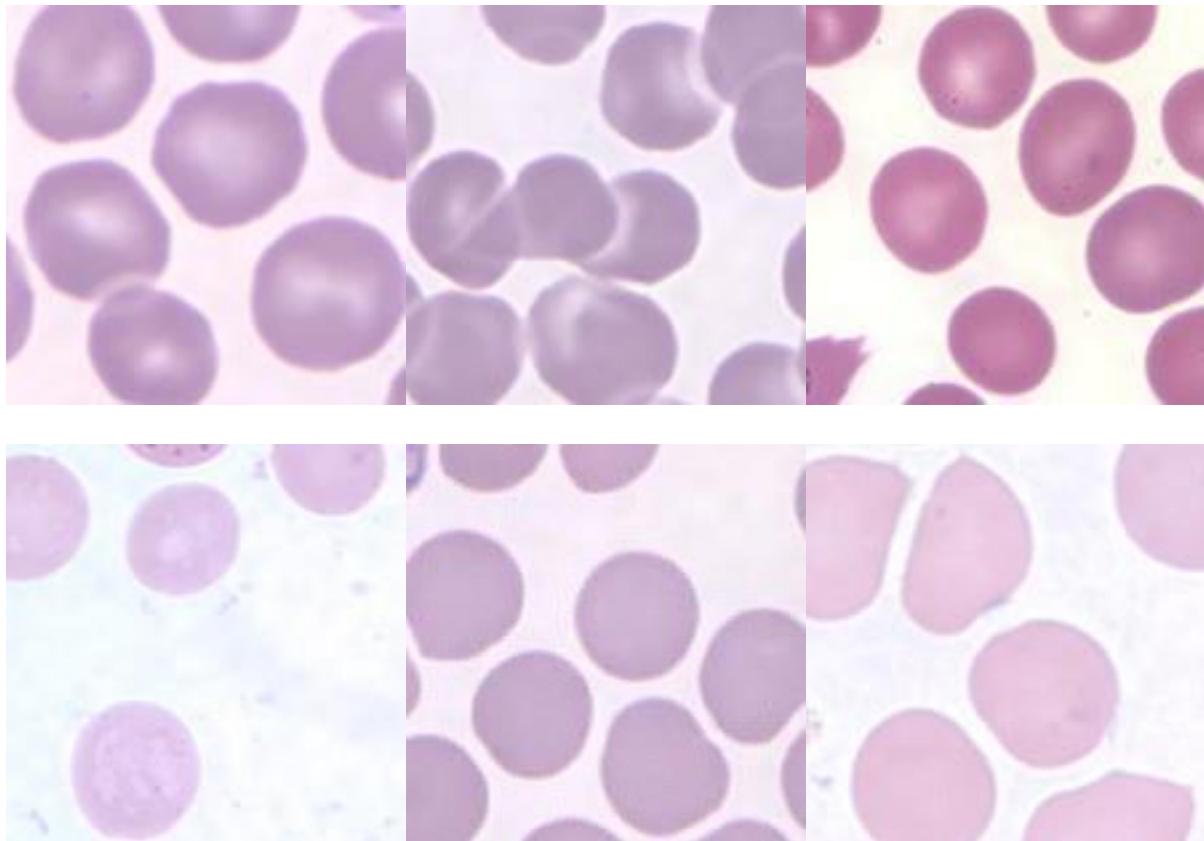


Fig 4.37 Examples of the training images for the uninfected class

Source: FAITH MEDIPLEX HOSPITAL