

MALARIA DETECTION USING IMAGE PROCESSING AND MACHINE LEARNING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in Information Technology
By

T. Sowmya Sree (Reg.No - 39120118)

K. Naga Sri Sai (Reg.No - 39120073)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

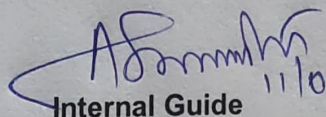
www.sathyabama.ac.in



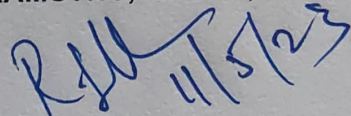
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **T. Sowmya Sree (39120118)** and **K. Naga Sri Sai (39120073)** who carried out the Project Phase-1 entitled “**Malaria Detection using Image Processing and Machine Learning**” under my supervision from January 2023 to May 2023.

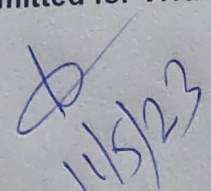

Internal Guide

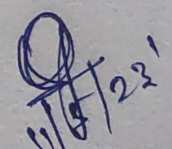
Dr. A. VEERAMUTHU, M.Tech., Ph. D.,


Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 11-05-2023


Internal Examiner

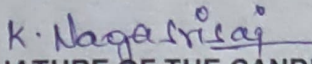

External Examiner

DECLARATION

I, **T. Sowmya Sree (39120118)** , hereby declare that the Project Phase-1 Report entitled "**Malaria Detection using Image Processing and Machine Learning**" done by me under the guidance of **Dr.A.Veeramuthu, M.Tech., Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.

DATE:

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T. SASIKALA, M.E., Ph.D., Dean, School of Computing, Dr.R.SUBHASHINI, M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A.Veeramuthu, M.Tech., Ph.D.,** for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Malaria is a significant burden on our healthcare system and it is the major cause of death in many developing countries. It is endemic in some parts of the world which means that the disease is regularly found in the region. Therefore, early testing is necessary to detect malaria and save lives. Thus, it gives us the motivation to make malaria diagnosis more effective and faster. A specialized technology proves essential to combat this problem.

For Malaria diagnosis, the RDT and microscopic diagnosis are the most used clinical methods. RDT is an effective and faster tool. Also, it does not require the presence of a trained medical professional. But RDT has few drawbacks like susceptibility to damage by heat and humidity and higher cost compared to a light microscope. The microscopic diagnosis systems do not suffer from these shortcomings, but it requires the presence of a trained microscopist. Machine learning has recently made headlines in the medical field. The machine learning approaches have proved to be successful in the diagnosis of a disease.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE No.
4.1	Architecture Diagram of Malaria Detection	14
5.1	Architecture Diagram of Image Processing	18
5.3	Pixel Displaying in image processing	23
5.4	An example Small Image (Grid Square)	24
5.4.	Derivative of an edge in one direction	27
6.1	Use Case Diagram	37
6.2	Sequence Diagram	38
6.3	Flow Diagram	39
8.1	Infected Image of Data Cell Image	42
8.2	uninfected Image of Data Cell Image	43
	Uninfected Image of The Cell Image	50
	Infected Image of The Cell Image	50

CHAPTER 1

INTRODUCTION

The study proposes an image processing model for detection of malaria infected cells. We use image processing techniques to detect parasite-infected red blood cells in thin smears on standard microscope slides the most widely used present day method is examining thin blood smears under a microscope, and visually searching for infected cells. A clinician manually counts the number of parasitic red blood cells - sometimes up to 5,000 cells (according to WHO protocol). Malaria could be forestalled, controlled, and relieved more adequately if an increasingly precise and effective symptomatic techniques were accessible. We have utilized image processing procedures to identify the nearness of malaria contaminated cells. And to classify the stage of malaria i.e., falciparum or non-falciparum we use machine learning.

Hundreds of millions of blood films are examined every year for malaria, which involves manual counting of parasites and infected red blood cells by a trained microscopist. Accurate parasite counts are essential not only for malaria diagnosis. They are also important for testing for drug resistance, measuring drug-effectiveness, and classifying disease severity. However, microscopic diagnostics is not standardized and depends heavily on the experience and skill of the microscopist. It is common for microscopists in low-resource settings to work in isolation, with no rigorous system in place that can ensure the maintenance of their skills and thus diagnostic quality. This leads to incorrect diagnostic decisions in the field. For false-negative cases, this leads to unnecessary use of antibiotics, a second consultation, lost days of work, and in some cases progression into severe malaria. For false positive cases, a misdiagnosis entails unnecessary use of anti-malaria drugs and suffering from their potential side effects, such as nausea, abdominal pain, diarrhea, and sometimes severe complications.

CHAPTER 2

LITERATURE REVIEW

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is considered for developing the proposed system.

2.1 INFERENCES FROM LITREATURE SURVEY

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

MALARIA Epidemic Analysis using Machine Learning and Deep Learning Algorithms

The catastrophic outbreak of severe acute respiratory syndrome - Coronavirus (SARS-CoV-2) also known as COVID-2019 has brought the worldwide threat to the living society. The whole world is putting incredible efforts to fight against the spread of this deadly disease in terms of infrastructure, finance, data sources, protective gears, life-risk treatments, and several other resources. The artificial intelligence researchers are focusing their expertise knowledge to develop mathematical models for analyzing this epidemic situation using nationwide shared data. To contribute towards the well-being of living society, this article proposes to utilize the machine

learning and deep learning models with the aim for understanding its everyday exponential behavior along with the prediction of future reachability of the COVID-2019 across the nations by utilizing the real-time information from the Johns Hopkins dashboard.

MALARIA in India: State-wise Analysis and Prediction

The highly infectious coronavirus disease (MALARIA) was first detected in Wuhan, China in December 2019 and subsequently spread to 212 countries and territories around the world, infecting millions of people. In India, a large country of about 1.3 billion people, the disease was first detected on January 30, 2020, in a student returning from Wuhan. The total number of confirmed infections in India as of May 3, 2020, is more than 37,000 and is currently growing fast.

MALARIA Outbreak Prediction with Machine Learning

Several outbreak prediction models for MALARIA are being used by officials around the world to make informed-decisions and enforce relevant control measures. Among the standard models for MALARIA global pandemic prediction, simple epidemiological and statistical models have received more attention by authorities, and they are popular in the media. Due to a high level of uncertainty and lack of essential data, standard models have shown low accuracy for long-term prediction. Although the literature includes several attempts to address this issue, the essential generalization and robustness abilities of existing models needs to be improved. This paper presents a comparative analysis of machine learning and soft computing models to predict the MALARIA outbreak. Among a wide range of machine learning models investigated, two models showed promising results (i.e., multi-layered perceptron, MLP, and adaptive network-based fuzzy inference system, ANFIS). Based on the results reported here, and due to the highly complex nature of the MALARIA outbreak and variation in its behavior from nation-to-nation, this study suggests machine learning as an effective tool to model the outbreak.

Forecasting the novel coronavirus MALARIA

What will be the global impact of the novel coronavirus (MALARIA)? Answering this question requires accurate forecasting the spread of confirmed cases as well as analysis of the number of deaths and recoveries. Forecasting, however,

requires ample historical data. At the same time, no prediction is certain as the future rarely repeats itself in the same way as the past. Moreover, forecasts are influenced by the reliability of the data, vested interests, and what variables are being predicted. Also, psychological factors play a significant role in how people perceive and react to the danger from the disease and the fear that it may affect them personally.

This paper introduces an objective approach to predicting the continuation of the MALARIA using a simple, but powerful method to do so. If the data used is reliable and that the future will continue to follow the past pattern of the disease, our forecasts suggest a continuing increase in the confirmed MALARIA cases with sizable associated uncertainty. The risks are far from symmetric as underestimating its spread like a pandemic and not doing enough to contain it is much more severe than overspending and being over careful when it will not be needed. This paper describes the timeline of a live forecasting exercise with massive potential implications for planning and decision making and provides objective forecasts for the confirmed cases of MALARIA.

Prediction of Spreads of MALARIA in India from Current Trend

The article describes modelling efforts for evaluating the current level of MALARIA infections in India, using exponential model. The Data from 15 march 2020 to 30 April 2020 are used for validating the model, where intrinsic rise rate is kept constant. It is observed that some states of India, like Maharashtra, Gujarat and Delhi have a much higher daily infection case. This is modelled by assuming an initial higher infection, keeping rise rate same. The sudden outbursts are captured using offset of values for these three states. Data from other states like Madhya Pradesh, Uttar Pradesh and Rajasthan are also analyzed and they are found to be following the same constants as India is following. Worldwide, many attempts are made to predict outburst of MALARIA and in the model, described in this paper, turning point is not predicted, as cases in India are still rising. The developed model is based on daily confirmed infections and not on cumulative infections and rationalization is carried out for the population of various regions, while predicting infections for various states.

Assigning a decay constant at this stage will be a premature exercise and keeping that in mind, exponential model predicts that India will attain 1 lakh case by 15 May 2020. The figure of 2 lakh and 3 lakhs will be attained on 22 May 2020 and

26 May 2020, respectively.

An interpretable mortality prediction model for MALARIA patients

The sudden increase in MALARIA cases is putting high pressure on healthcare services worldwide. At this stage, fast, accurate, and early clinical assessment of the disease severity is vital. To support decision making and logistical planning in healthcare systems, this study leverages a database of blood samples from 485 infected patients in the region of Wuhan, China, to identify crucial predictive biomarkers of disease mortality.

For this purpose, machine learning tools selected three biomarkers that predict the mortality of individual patients more than 10 days in advance with more than 90% accuracy: lactic dehydrogenase (LDH), lymphocyte and high-sensitivity C-reactive protein (hs-CRP). Relatively high levels of LDH alone seem to play a crucial role in distinguishing the vast majority of cases that require immediate medical attention. This finding is consistent with current medical knowledge that high LDH levels are associated with tissue breakdown occurring in various diseases, including pulmonary disorders such as pneumonia. Overall, this Article suggests a simple and operable decision rule to quickly predict patients at the highest risk, allowing them to be prioritized and potentially reducing the mortality rate.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

- The person administering the test will twirl a long stick with a soft brush on the end, like a pipe cleaner, up your nose for a few seconds. There will be a collection of secretions collected by the soft bristles for study. Malaria is one of the deadliest diseases across the globe. This is caused by the bite of female Anopheles mosquito that transmits the Plasmodium parasites. Some current malaria detection techniques include manual microscopic examination and RDT. These approaches are vulnerable to human mistakes.
- The existing system by automating the process of malaria detection using K Means.

CHAPTER 3

PROPOSED SYSTEM

- We have proposed an automatic prediction of Malaria using a deep convolutional neural network. The proposed models will have end-to-end structure without manual feature extraction and selection methods. The proposed model consists of three convolutional layers and fully connected layers each. The neural network presented is a cascade of several convolutional layers having multiple filters present in layers, which yields the exceptionally good accuracy as per the available resources. The model is trained and later several blood sample images are fed to test the accuracy of the designed system.
- The CNN classifier has performed exceptionally well under limited computational resources giving an accuracy of 95%. Blood smear sample analysis can also aid in the detection of certain other illnesses and the application of deep learning models will help in the greater good of humankind.
- The metrics like accuracy, sensitivity, and F1-measure are calculated for performance evaluation. The performance of the proposed method is better than the existing methodologies and thus can be used for the effective diagnosis of the disease.

CHAPTER 4

ARCHITECTURE DIAGRAM

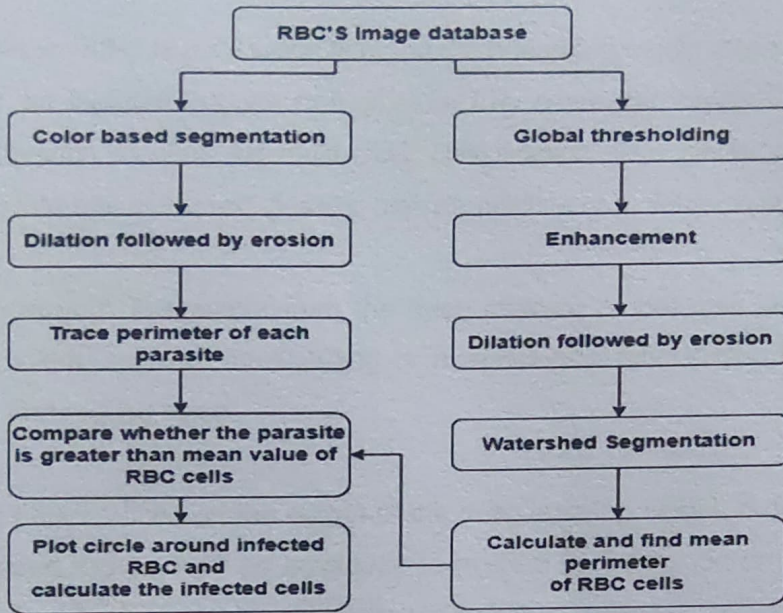


Fig 4.1: Architecture Diagram of Malaria Detection.

1. **Input Image:** The system takes input images of blood smears or other relevant microscopic images that are suspected to contain malaria parasites. These images are typically captured using a microscope or other imaging equipment.

2. **Pre-processing:** The input images undergo pre-processing steps, which may include image resizing, normalization, and noise reduction, to enhance the quality and consistency of the images before they are fed into the deep learning model.

3. **Deep Learning Model:** The heart of the system is a deep learning model, which is a type of neural network designed to learn complex patterns from large datasets. The deep learning model is trained on a labeled dataset of malaria-positive and malaria-negative images to learn the features that distinguish between infected and uninfected cells. Common deep learning architectures used for malaria detection include Convolutional Neural Networks (CNNs) or variants like ResNet or Inception.

4. **Feature Extraction:** Within the deep learning model, there are multiple layers that

perform feature extraction. These layers automatically learn relevant features such as color, texture, and shape from the input images. The deeper layers of the network typically capture more abstract features.

5. Classification: After feature extraction, the deep learning model makes predictions by passing the features through one or more fully connected layers followed by a softmax activation function for multi-class classification. The model classifies the input image into one of the two classes: malaria-positive or malaria-negative.

6. Post-processing: The output from the deep learning model may undergo post-processing steps, such as thresholding or morphological operations, to refine the predictions and reduce noise.

7. Decision Logic: Based on the output of the deep learning model, a decision logic or a rule-based system may be employed to make a final decision on whether the input image is classified as malaria-positive or malaria-negative.

8. Output: The final output of the system is the prediction or classification result, which indicates whether the input image contains malaria parasites or not. This output can be visualized or further used for downstream applications, such as generating a report or triggering alerts.

9. Training Data: The deep learning model is trained on a labeled dataset of blood smear images, which includes samples that are confirmed to be either malaria-positive or malaria-negative. This dataset is crucial for training the model to accurately classify new, unseen images.

10. Validation Data: During the training process, a validation dataset is used to assess the performance of the model and tune its hyperparameters, such as learning rate or batch size, to optimize its accuracy and generalization ability.

11. Testing Data: Once the model is trained, it is evaluated on a separate testing dataset that contains unseen images to assess its real-world performance and validate its accuracy and robustness.

CHAPTER 5

TOOLS USED FOR THE PROJECT

In order to detect the presence of malaria parasites, convolutional neural networks (CNNs) are used to analyze images of blood smears. Here are a few frequently used programs and libraries for creating CNN-based malaria detection models:

1. Python: One well-liked programming language for creating machine learning models is Python. For data pre-processing, modelling, and assessment, it has a variety of libraries including NumPy, Pandas, and Scikit-learn.
 2. TensorFlow: TensorFlow is a well-liked open-source library for creating CNNs and other machine learning models. It offers a versatile framework for constructing and training sophisticated neural networks.
 3. Keras: On top of TensorFlow, Keras is a high-level neural network API. It offers a user-friendly interface for creating and refining CNNs and other deep learning models.
 4. PyTorch: This open-source machine learning toolkit offers a versatile foundation for creating neural networks, including CNNs. Due to its simplicity of use and dynamic computation graph, it has grown in popularity.
 5. Image pre-processing libraries: To achieve consistency and minimize noise, the images must be pre-processed before training a CNN model. You may resize, normalize, and improve images using libraries like OpenCV and Pillow.
 6. Deep Learning Studio, a web-based tool for creating, honing, and deploying deep learning models. It offers a simple user interface for creating and refining CNN models for the detection of malaria.
 7. Malaria dataset: To train and evaluate CNN models for malaria detection, a variety of publicly accessible malaria datasets are available, such as the Kaggle Malaria Dataset.
 8. Jupyter Notebook: Jupyter Notebook is an open-source web tool that enables users to create and share documents that include text, live code, equations, and visuals. It is a fantastic resource for creating and evaluating CNN models.
- These resources can be used to create and train CNN models for malaria detection that are precise and effective.

5.1 IMAGE PROCESSING

What is Image Processing?

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually, **Image Processing** system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Types:

The two types of **methods used for Image Processing** are **Analog** and **Digital** Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual

techniques. So, analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

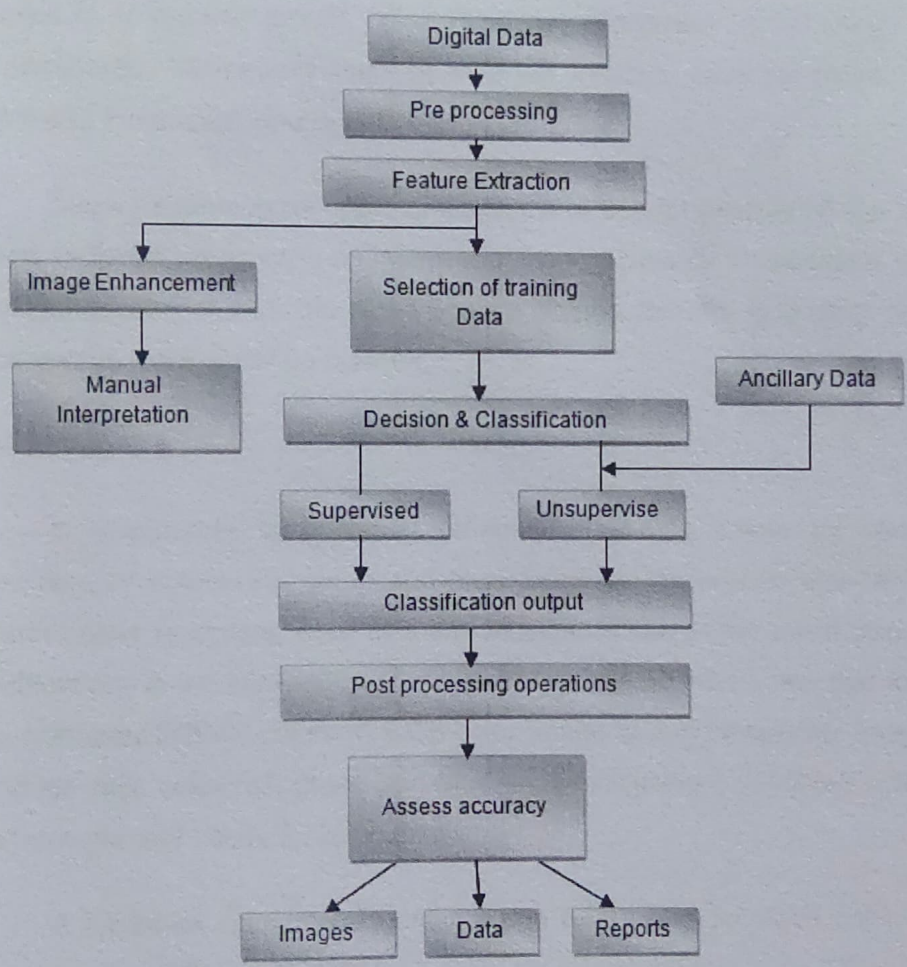


Fig 5.1: Architecture Diagram of Image Processing.

5.2 IMAGE PROCESSING CONCEPTS

Binary Images

Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white. Numerically, the two values are often 0 for black, and either 1 or 255 for white.

Binary images are often produced by thresholding a grayscale or color image, in order to separate an object in the image from the background. The color of the object (usually white) is referred to as the foreground color. The rest (usually black) is referred to as the background color. However, depending on the image which is to be thresholded, this polarity might be inverted, in which case the object is displayed with 0 and the background is with a non-zero value.

Some morphological operators assume a certain polarity of the binary input image so that if we process an image with inverse polarity the operator will have the opposite effect. For example, if we apply a closing operator to a black text on white background, the text will be opened.

Color Images

It is possible to construct (almost) all visible colors by combining the three primary colors red, green and blue, because the human eye has only three different color receptors, each of them sensible to one of the three colors. Different combinations in the stimulation of the receptors enable the human eye to distinguish approximately 350000 colors. A RGB color image is a multi-spectral image with one band for each color red, green and blue, thus producing a weighted combination of the three primary colors for each pixel.

A full 24-bit color image contains one 8-bit value for each color, thus being able to display $2^{24} = 16777216$ different colors.

However, it is computationally expensive and often not necessary to use the full 24-bit image to store the color for each pixel. Therefore, the color for each pixel is often encoded in a single byte, resulting in an 8-bit color image. The process of

reducing the color representation from 24-bits to 8-bits, known as color quantization, restricts the number of possible colors to 256. However, there is normally no visible difference between a 24-color image and the same image displayed with 8 bits. An 8-bit color images are based on colormaps, which are look-up tables taking the 8-bit pixel value as index and providing an output value for each color.

8-bit Color Images

Full RGB color requires that the intensities of three-color components be specified for each and every pixel. It is common for each component intensity to be stored as an 8-bit integer, and so each pixel requires 24 bits to specify its color completely and accurately. If this is done, then the image is known as a 24-bit color image. However, there are two problems with this approach:

- Storing 24 bits for every pixel leads to very large image files that with current technology are cumbersome to store and manipulate. For instance, a 24-bit 512×512 image takes up 750KB in uncompressed form.
- Many monitors display use colormaps with 8-bit index numbers, meaning that they can only display 256 different colors at any one time. Thus, it is often wasteful to store more than 256 different colors in an image anyway, since it will not be possible to display them all on screen.

Because of this, many image formats (e.g., 8-bit GIF and TIFF) use 8-bit colormaps to restrict the maximum number of different colors to 256. Using this method, it is only necessary to store an 8-bit index into the colormap for each pixel, rather than the full 24-bit color value. Thus 8-bit image formats consist of two parts: a colormap describing what colors are present in the image, and the array of index values for each pixel in the image.

When a 24-bit full color image is turned into an 8-bit image, it is usually necessary to throw away some of the colors, a process known as color quantization. This leads to some degradation in image quality, but in practice the observable effect can be quite small, and in any case, such degradation is inevitable if the image output device (e.g., screen or printer) is only capable of displaying 256 colors or less.

The use of 8-bit images with colormaps does lead to some problems in image

processing. First of all, each image has to have its own colormap, and there is usually no guarantee that each image will have exactly the same colormap. Thus on 8-bit displays it is frequently impossible to correctly display two different color images that have different colormaps at the same time. Note that in practice 8-bit images often use reduced size colormaps with less than 256 colors in order to avoid this problem.

Another problem occurs when the output image from an image processing operation contains different colors to the input image or images. This can occur very easily, as for instance when two color images are added together pixel-by-pixel. Since the output image contains different colors from the input images, it ideally needs a new colormap, different from those of the input images, and this involves further color quantization which will degrade the image quality. Hence the resulting output is usually only an approximation of the desired output. Repeated image processing operations will continually degrade the image colors. And of course, we still have the problem that it is not possible to display the images simultaneously with each other on the same 8-bit display.

Because of these problems it is to be expected that as computer storage and processing power become cheaper, there will be a shift away from 8-bit images and towards full 24-bit image processing.

24-bit Color Images

Full RGB color requires that the intensities of three-color components be specified for each pixel. It is common for each component intensity to be stored as an 8-bit integer, and so each pixel requires 24 bits to specify its color completely and accurately. Image formats that store a full 24 bits to describe the color of every pixel are therefore known as 24-bit color images.

Using 24 bits to encode color information allows $2^{24} = 16777216$ different colors to be represented, and this is sufficient to cover the full range of human color perception fairly well.

The term 24-bit is also used to describe monitor displays that use 24 bits per pixel in their display memories, and which are hence capable of displaying a full

range of colors.

There are also some disadvantages to using 24-bit images. Perhaps the main one is that it requires three times as much memory, disk space and processing time to store and manipulate 24-bit color images as compared to 8-bit color images. In addition, there is often not much point in being able to store all those different colors if the final output device (e.g., screen or printer) can only actually produce a fraction of them. Since it is possible to use colormaps to produce 8-bit color images that look almost as good, at the time of writing 24-bit displays are relatively little used. However, it is to be expected that as the technology becomes.

5.3 Color Quantization

Color quantization is applied when the color information of an image is to be reduced. The most common case is when a 24-bit color image is transformed into an 8-bit color image.

Two decisions have to be made:

1. which colors of the larger color set remain in the new image, and
2. how are the discarded colors mapped to the remaining ones.

The simplest way to transform a 24-bit color image into 8 bits is to assign 3 bits to red and green and 2 bits to blue (blue has only 2 bits, because of the eye's lower sensitivity to this color). This enables us to display 8 different shades of red and green and 4 of blue. However, this method can yield only poor results. For example, an image might contain different shades of blue which are all clustered around a certain value such that only one shade of blue is used in the 8-bit image and the remaining three blues are not used.

Alternatively, since 8-bit color images are displayed using a colormap, we can assign any arbitrary color to each of the 256 8-bit values and we can define a separate colormap for each image. This enables us perform a color quantization adjusted to the data contained in the image. One common approach is the popularity algorithm, which creates a histogram of all colors and retains the 256 most frequent ones. Another approach, known as the median-cut algorithm, yields even better

results but also needs more computation time. This technique recursively fits a box around all colors used in the color space which it splits at the median value of its longest side. The algorithm stops after 255 recursions. All colors in one box are mapped to the centroid of this box.

All above techniques restrict the number of displayed colors to 256. A technique of achieving additional colors is to apply a variation of half-toning used for gray scale images, thus increasing the color resolution at the cost of spatial resolution. The 256 values of the colormap are divided into four sections containing 64 different values of red, green, blue and white. As can be seen in Figure 1, a 2×2-pixel area is grouped together to represent one composite color, each of the four pixels displays either one of the primary colors or white. In this way, the number of possible colors is increased from 256 to 64^4 .

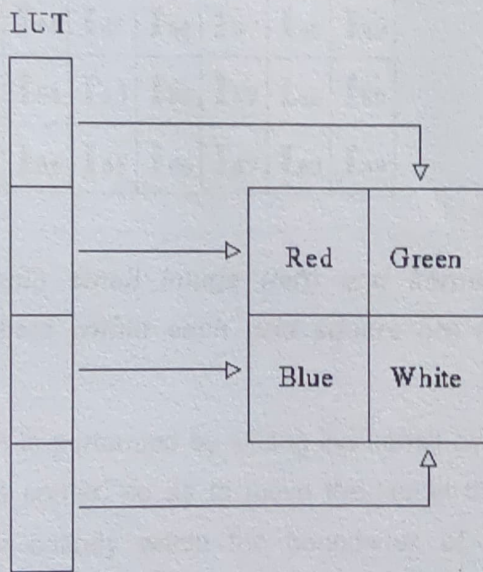


Fig 5.3: A 2×2-pixel area displaying one composite color cheaper, their use in image processing will grow.

5.4 Convolution

Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of 'multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This

can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values.

In an image processing context, one of the input arrays is normally just a gray level image. The second array is usually much smaller, and is also two-dimensional (although it may be just a single pixel thick), and is known as the kernel. Figure 1 shows an example image and kernel that we will use to illustrate convolution.

I₁₁	I₁₂	I₁₃	I₁₄	I₁₅	I₁₆	I₁₇	I₁₈	I₁₉
I₂₁	I₂₂	I₂₃	I₂₄	I₂₅	I₂₆	I₂₇	I₂₈	I₂₉
I₃₁	I₃₂	I₃₃	I₃₄	I₃₅	I₃₆	I₃₇	I₃₈	I₃₉
I₄₁	I₄₂	I₄₃	I₄₄	I₄₅	I₄₆	I₄₇	I₄₈	I₄₉
I₅₁	I₅₂	I₅₃	I₅₄	I₅₅	I₅₆	I₅₇	I₅₈	I₅₉
I₆₁	I₆₂	I₆₃	I₆₄	I₆₅	I₆₆	I₆₇	I₆₈	I₆₉

K₁₁	K₁₂	K₁₃
K₂₁	K₂₂	K₂₃

Figure:5.4: An example small image (left) and kernel (right) to illustrate convolution. The labels within each grid square are used to identify each square.

The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges of images, as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

So, in our example, the value of the bottom right pixel in the output image will be given by:

$$O_{57} = I_{57}K_{11} + I_{58}K_{12} + I_{59}K_{13} + I_{67}K_{21} + I_{68}K_{22} + I_{69}K_{23}$$

If the image has M rows and N columns, and the kernel has m rows and n columns, then the size of the output image will have $M - m + 1$ rows, and $N - n + 1$ columns.

Mathematically we can write the convolution as:

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n I(i + k - 1, j + l - 1) K(k, l)$$

where i runs from 1 to $M - m + 1$ and j runs from 1 to $N - n + 1$.

Note that many implementations of convolution produce a larger output image than this because they relax the constraint that the kernel can only be moved to positions where it fits entirely within the image. Instead, these implementations typically slide the kernel to all positions where just the top left corner of the kernel is within the image. Therefore, the kernel 'overlaps' the image on the bottom and right edges. One advantage of this approach is that the output image is the same size as the input image. Unfortunately, in order to calculate the output pixel values for the bottom and right edges of the image, it is necessary to invent input pixel values for places where the kernel extends off the end of the image. Typically, pixel values of zero are chosen for regions outside the true image, but this can often distort the output image at these places. Therefore, in general if you are using a convolution implementation that does this, it is better to clip the image to remove these spurious regions. Removing $n - 1$ pixels from the right-hand side and $m - 1$ pixels from the bottom will fix things.

Convolution can be used to implement many different operators, particularly spatial filters, and feature detectors. Examples include Gaussian smoothing and the Sobel edge detector.

Distance Metrics

It is often useful in image processing to be able to calculate the distance between two pixels in an image, but this is not as straightforward as it seems. The presence of the pixel grid makes several so-called distance metrics possible which often give different answers to each other for the distance between the same pair of points. We consider the three most important ones.

Euclidean Distance

This is the familiar straight-line distance that most people are familiar with. If the two pixels that we are considering have coordinates (x_1, y_1) and (x_2, y_2) , then the Euclidean distance is given by:

$$D_{Euclid} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

City Block Distance

Also known as the Manhattan distance. This metric assumes that in going from one pixel to the other it is only possible to travel directly along pixel grid lines. Diagonal moves are not allowed. Therefore the 'city block' distance is given by:

$$D_{City} = |x_2 - x_1| + |y_2 - y_1|$$

Chessboard Distance

This metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess, i.e., a diagonal move counts the same as a horizontal move. This means that the metric is given by:

$$D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$$

Note that the last two metrics are usually much faster to compute than the Euclidean metric and so are sometimes used where speed is critical but accuracy is not too important.

Dithering

Dithering is an image display technique that is useful for overcoming limited display resources. The word dither refers to a random or semi-random perturbation of the pixel values.

Two applications of this techniques are particularly useful:

Low quantization display: When images are quantized to a few bits (e.g., 3)

then only a limited number of gray levels are used in the display of the image. If the scene is smoothly shaded, then the image display will generate rather distinct boundaries around the edges of image regions when the original scene intensity moves from one quantization level to the next. To eliminate this effect, one dithering technique adds random noise (with a small range of values) to the input signal before quantization into the output range. This randomizes the quantization of the pixels at the original quantization boundary, and thus pixels make a more gradual transition from neighborhoods containing 100% of the first quantization level to neighborhoods containing 100% of the second quantization level.

Limited color display: When fewer colors are able to be displayed (e.g. 256) than are present in the input image (e.g. 24 bit color), then patterns of adjacent pixels are used to simulate the appearance of the unrepresented colors.

Edge Detectors

Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image information.

Since edges consist of mainly high frequencies, we can, in theory, detect edges by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. In practice, edge detection is performed in the spatial domain, because it is computationally less expensive and often yields better results.

Since edges correspond to strong illumination gradients, we can highlight them by calculating the derivatives of the image. This is illustrated for the one-dimensional case in the below figure

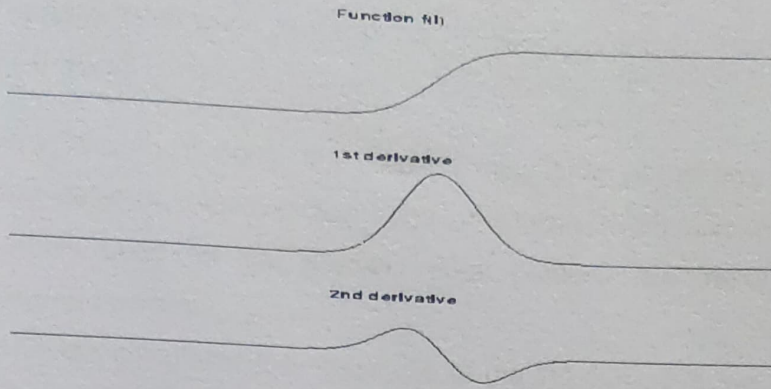


Figure 5.4: 1st and 2nd derivative of an edge illustrated in one dimension.

We can see that the position of the edge can be estimated with the maximum of the 1st derivative or with the zero-crossing of the 2nd derivative. Therefore, we want to find a technique to calculate the derivative of a two-dimensional image. For a discrete one-dimensional function $f(i)$, the first derivative can be approximated by

$$\frac{df(i)}{d(i)} = f(i+1) - f(i)$$

Calculating this formula is equivalent to convolving the function with $[-1 \ 1]$. Similarly, the 2nd derivative can be estimated by convolving $f(i)$ with $[1 \ -2 \ 1]$.

Different edge detection kernels which are based on the above formula enable us to calculate either the 1st or the 2nd derivative of a two-dimensional image. There are two common approaches to estimate the 1st derivative in a two-dimensional image, Prewitt compass edge detection and gradient edge detection.

Prewitt compass edge detection involves convolving the image with a set of (usually 8) kernels, each of which is sensitive to a different edge orientation. The kernel producing the maximum response at a pixel location determines the edge magnitude and orientation. Different sets of kernels might be used: examples include Prewitt, Sobel, Kirsch and Robinson kernels.

Gradient edge detection is the second and more widely used technique. Here, the image is convolved with only two kernels, one estimating the gradient in the x-direction, G_x , the other the gradient in the y-direction, G_y . The absolute gradient magnitude is then given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

and is often approximated with

$$|G| = |G_x| + |G_y|$$

In many implementations, the gradient magnitude is the only output of a gradient edge detector, however the edge orientation might calculate the most common kernels used for the gradient edge detector are the Sobel, Roberts Cross and Prewitt operators.

After having calculated the magnitude of the 1st derivative, we now have to identify those pixels corresponding to an edge. The easiest way is to threshold the gradient image, assuming that all pixels having a local gradient above the threshold must represent an edge. An alternative technique is to look for local maxima in the gradient image, thus producing one-pixel wide edges. A more sophisticated technique is used by the Canny edge detector. It first applies a gradient edge detector to the image and then finds the edge pixels using non-maximal suppression and hysteresis tracking.

An operator based on the 2nd derivative of an image is the Marr edge detector, also known as zero crossing detector. Here, the 2nd derivative is calculated using a Laplacian of Gaussian (LoG) filter. The Laplacian has the advantage that it is an isotropic measure of the 2nd derivative of an image, i.e. the edge magnitude is obtained independently from the edge orientation by convolving the image with only one kernel. The edge positions are then given by the zero-crossings in the LoG image. The scale of the edges which are to be detected can be controlled by changing the variance of the Gaussian.

A general problem for edge detection is its sensitivity to noise, the reason being that calculating the derivative in the spatial domain corresponds to accentuating high frequencies and hence magnifying noise. This problem is addressed in the Canny and Marr operators by convolving the image with a smoothing operator (Gaussian) before calculating the derivative.

Frequency Domain

For simplicity, assume that the image I being considered is formed by projection from scene S (which might be a two- or three-dimensional scene, etc.).

The frequency domain is a space in which each image value at image position F represents the amount that the intensity values in image I vary over a specific distance related to F . In the frequency domain, changes in image position correspond to changes in the spatial frequency, (or the rate at which image intensity values) are changing in the spatial domain image I .

For example, suppose that there is the value 20 at the point that represents the frequency 0.1 (or 1 period every 10 pixels). This means that in the corresponding spatial domain image I the intensity values vary from dark to light and back to dark over a distance of 10 pixels, and that the contrast between the lightest and darkest is 40 gray levels (2 times 20).

The spatial frequency domain is interesting because: 1) it may make explicit periodic relationships in the spatial domain, and 2) some image processing operators are more efficient or indeed only practical when applied in the frequency domain.

In most cases, the Fourier Transform is used to convert images from the spatial domain into the frequency domain and vice-versa.

A related term used in this context is spatial frequency, which refers to the (inverse of the) periodicity with which the image intensity values change. Image features with high spatial frequency (such as edges) are those that change greatly in intensity over short image distances.

Grayscale Images

A grayscale (or graylevel) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a 'gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the

difference between successive gray levels is significantly better than the graylevel resolving power of the human eye.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

5.5 Image Editing Software

There is a huge variety of software for manipulating images in various ways. Much of this software can be grouped under the heading image processing software, and the bulk of this reference is concerned with that group.

Another very important category is what we call image editing software. This group includes painting programs, graphic art packages and so on. They are often useful in conjunction with image processing software packages, in situations where direct immediate interaction with an image is the easiest way of achieving something.

For instance, if a region of an image is to be masked out for subsequent image processing, it may be easiest to create the mask using an art package by directly drawing on top of the original image. The mask used in the description of the AND operator was created this way for instance. Art packages also often allow the user to move sections of the images around and brighten or darken selected regions interactively. Few dedicated image processing packages offer the same flexibility and ease of use in this aspect.

CHAPTER 6

SOFTWARE ENVIRONMENT

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
```

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.


```
>>> print("Hello, World!")
```

Hello, World!

python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>>                                     print("Hello,                               World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
```

```
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
```

```
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:


```
#This is a comment.  
print("Hello, World!")  
Docstrings
```

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a  
multiline docstring."""  
print("Hello, World!")
```

6.1. MODULES

- Data collection Module
- Data Pre-Processing Module
- Training the CNN Module
- classification Module

MODULE DESCRIPTION

Data collection Module

- This dataset consists of two folders: the test and train folder with the testing and training images, respectively. Further, there are two more folders - parasitized folder with the malaria-infected cell images and the uninfected folder with the normal cell images. As we have an equal distribution of cell images, so we don't have to deal with the problems of imbalanced data, and so it has fewer chances of being biased towards a particular class.

Data Pre-Processing Module

- The input images are normalized prior to any further processing. Normalized images are enhanced images with no errors due to lightening conditions. This segment mentioned the technique of preparing the raw data to make these suitable for building and training machine learning models.
- This is the first stage. The aim of data preprocessing is to clean the cell images. This stage will remove the unwanted noise

Image segmentation:

- This is the second stage where the image is segmented to extract the region of interest from the image, here we will be using two image segmentation techniques as follows:

Otsu's segmentation

- It is used to perform automatic segmentation by image thresholding of the cell images. It returns a single intensity threshold value which separates pixels into two classes.

Watershed segmentation:

- Watershed segmentation is an image segmentation algorithm that segments an image into several catchment basins or regions. The input image can be segmented into regions where conceptually rainwater would flow into the same lake and this will result in partitioning the image into catchment basins and watershed lines.

Training the CNN Module

- Before we thought of training the CNN, we will load the data and target. Before training the model, we will set up data generators to read images from source folders instead of labeling the image one by one.

CNN is a model which is designed to process arrays of data such as images. The first step here will be to resize all images as CNN cannot train images of different sizes. We compute the mean for both the dimensions and resize all the images. Now we can use random transform and we will double our dataset

classification Module

- The segmented image is given as an input and we get the feature vectors which are then used for the classification task. Here we will be using Convolutional Neural Network (CNN) model with the Relu activation function.

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

6.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

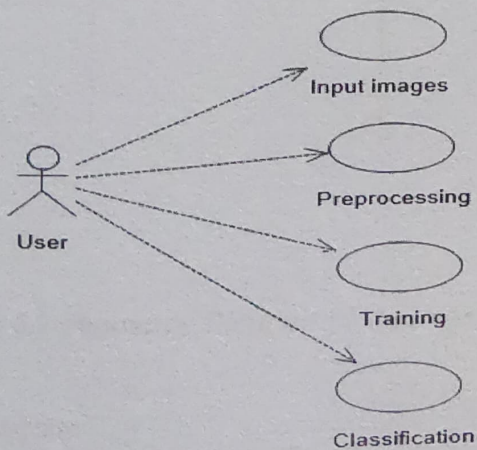


Fig 6.1: Use Case Diagram of Modules.

6.2 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

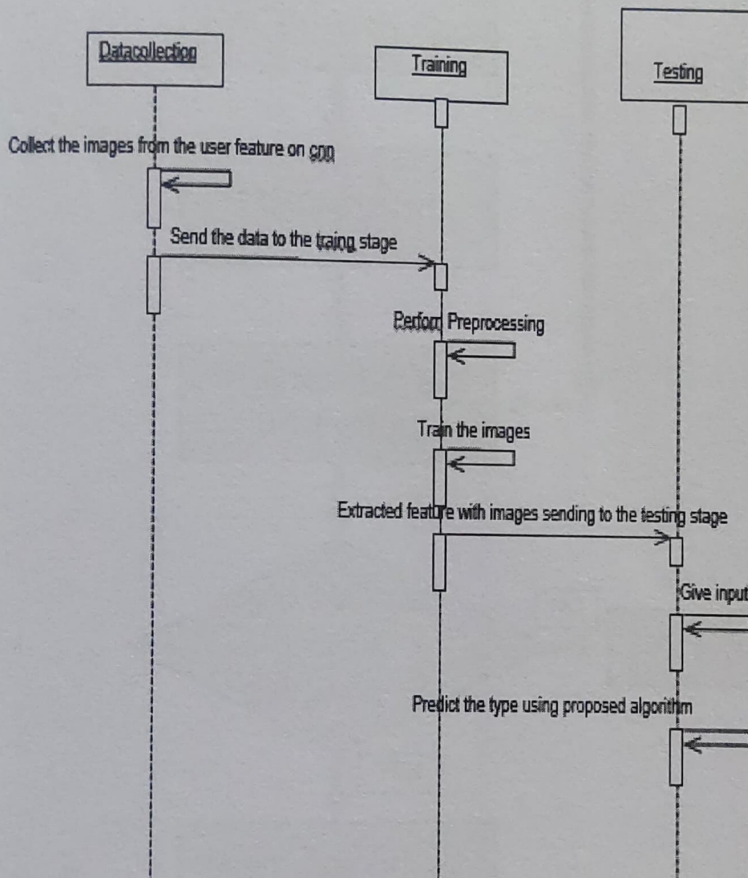


Fig 6.2 sequence Diagram of Modules used in Malaria Detection.

6.3 Flow Diagram

Flow diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business

and operational 5-step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

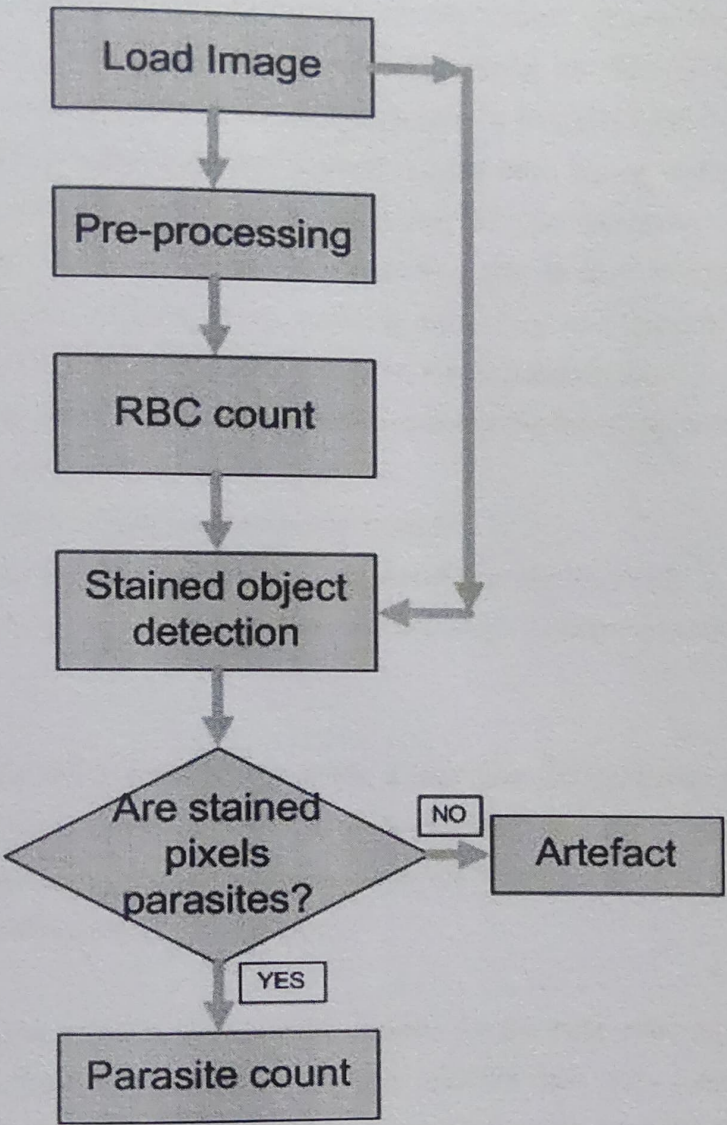


Fig 6.3: Flow Diagram of Malaria Detection Process.

CHAPTER 7

EXPERIMENTAL SETUP

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

CHAPTER 8

RESULT AND DISCUSSIONS

Quality is an end result that meets the cease user's requirements and suggests the facts simply. In any device, the consequences of the method are stated to users and other systems through outputs. The output plan defines how facts is to be moved for fast need in addition to for revealed output. It is the number one and instant source of facts for the user. Efficient and smart output design of the connection gadget improves, supporting the user to make choices.

1. The improvement of computer products must be organized and properly thought out; the proper output has to be designed in order that each output detail is prepared in such a manner that humans can use the system easily and efficiently. When reading the pc's output, it's far necessary to decide the unique output to satisfy the necessities.
2. Choose the way to present statistics.
3. Create a record, report or other layout containing the records generated with the aid of the system.

The output format of the records device has to carry out one or extra of the following capabilities.

Infected With Malaria

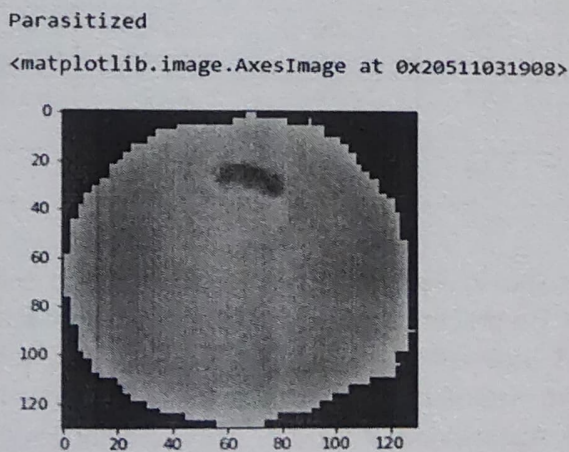


Fig 8.1: Infected Image of Data cell Image.

The output of the cell image of infected dataset. When an image from the infected dataset is trained and uploaded in the command prompt. Then this will be the output of uploaded image as follows in the Fig.8.1

Uninfected With Malaria

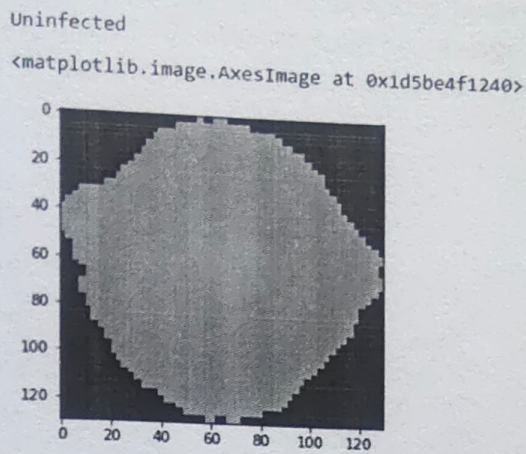


Fig 8.2: Uninfected Image of Data cell Image.

The output of the cell image of uninfected dataset. When an image from the uninfected dataset is trained and uploaded in the command prompt. Then this will be the output of uploaded image as follows in the Fig.8.2

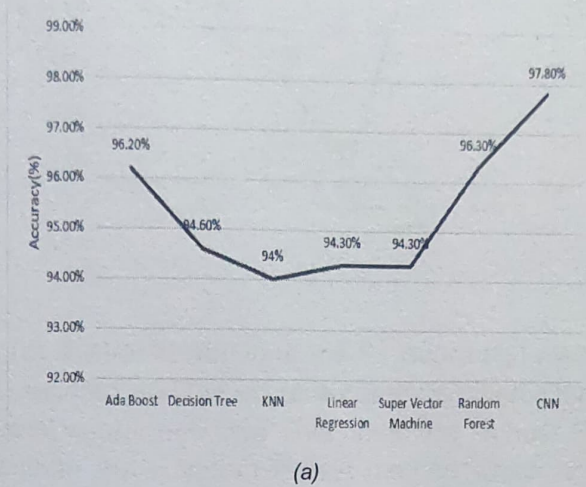
PERFORMANCE COMPARISION TABLE

TABLE 1

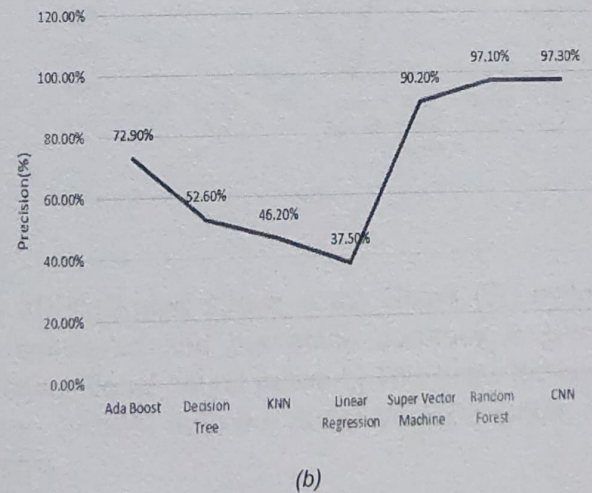
S. No	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Recall (%)
1	Ada Boost	96.2	72.9	52.6	61.1	52.6
2	Decision Tree	94.6	52.6	52.9	52.7	52.9
3	KNN	94	46.2	27.8	34.8	27.8
4	Linear Regression	94.3	37.5	0	0	0
5	Super Vector Machine	94.3	90.2	94.5	90.7	94.5
6	Random Forest	96.3	97.1	92.3	95.6	92.3
7	CNN	97.8	97.3	94.6	95.7	94.6

This table shows the comparison of performances of our method and the other metrics used to detect the malaria disease. We have taken 7 methods and 4 metrics into consideration for our execution in Malaria detection. First method We have taken is Ada boost where it's Accuracy shows different percentages in different metrics here the values for accuracy is 96.2%, percentage for precision is 72.9%, percentage for F1 Score is 61.1% and the percentage for Recall is 52.6%. Then comes the next metrics Decision Tree where the accuracy level is 94.6%, Precision percentage is 52.6%, F1 Score percentage is 52.9% and the percentage of Recall is 52.9%. Next is the KNN (K nearest neighbor) the Accuracy value is 94%, precision percentage is 46.2%, F1 Score 34.8% and Recall percentage value is 27.8%. then comes the Linear Regression methods and the metrics percentage values for

Accuracy is 94.3%, precision percentage is 37.5%, the percentage value for F1 Score is 0%. Next comes the CNN (Convolutional neural network) method and the metrics percentage for Accuracy is 97.8%, precision percentage is 97.3%, F1 Score value percentage is 95.7% and the Recall percentage value is 94.6%.

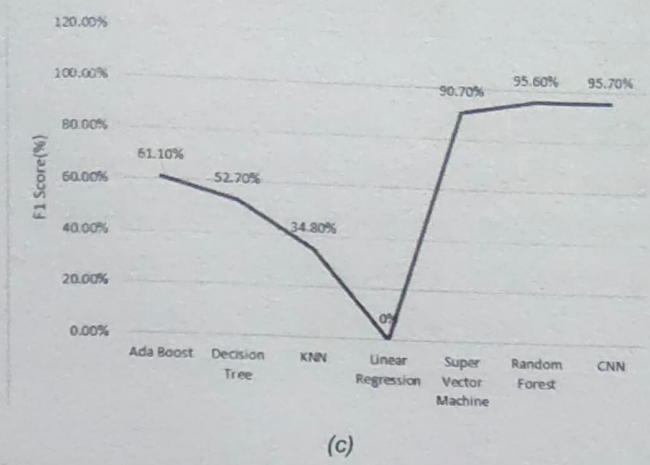


The graphs (a) illustrated above is the accuracy (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest accuracy level among the graph i.e., 97.80% than the Random Forest method which holds an accuracy of 96.30%.

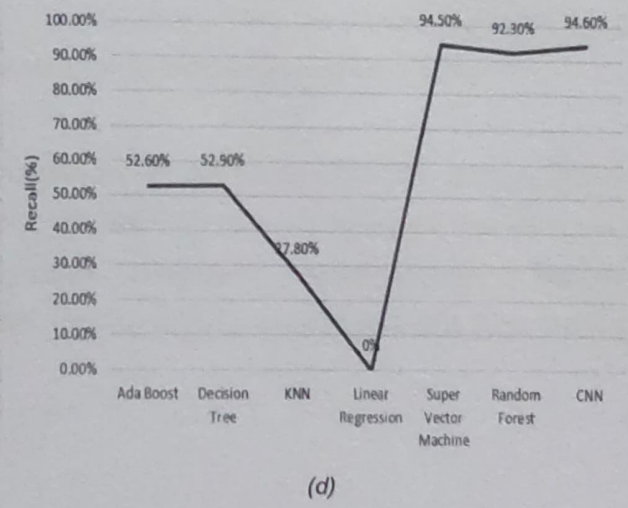


The graphs (b) illustrated above is the Precision (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest precision among the

graph i.e., 97.30% than the Random Forest method which holds an accuracy of 97.10%



The graphs (c) illustrated above is the F1 Score (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest F1 Score among the graph i.e., 95.70% than the Random Forest method which holds an accuracy of 95.60%.



The graphs (d) illustrated above is the Recall (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest accuracy level among the graph i.e., 94.60% than the Super Vector machine method which holds an accuracy of 94.50%.

CHAPTER 9

CONCLUSION

We have presented an approach and developed an algorithm for detecting malaria, automated malaria detection and quantification of malaria infection. Also, we developed a strategy to train with machine learning, adaptable to detection of malaria with other types of parasites and also discuss to increase the predictive value with results. Considering our above methodology, we come to the conclusion that the newly designed system of image processing technologies is suitable for parasite detection.

The morphological segmentation techniques prove to be the best in detection of the cells and segmenting them from the non- region of interest portion. To characterize the sort of plasmodium parasite for which we utilized machine learning technologies and cubic CNN proves to be the best by having the highest accuracy score. The system has been tested with around 110 thin film blood smear images and the results are satisfactory. The inconsistencies among the images were a challenge but we have tried to make our system robust.

This model achieved an accuracy of 97.8%. Using data augmentation in the Convolutional Neural Network approach decreases the chances of overfitting. Thus, Malaria detection systems using deep learning proved to be faster than most of the traditional techniques. The system is easy to use and user-friendly.

REFERENCES

- [1] Features, Evaluation and Treatment Coronavirus (MALARIA). Available: <https://www.ncbi.nlm.nih.gov/books/NBK554776/>, May 18 (2020).
- [2] N. S Punna, S. K. Sonbhadra, S. Agarwal, "MALARIA Epidemic Analysis using Machine Learning and Deep Learning Algorithms", medRxiv, Available: doi <https://doi.org/10.1101/2020.04.08.20057679> June 1 (2020).
- [3] P. Ghosh, R. Ghosh, B. Chakraborty, "MALARIA in India: State-wise Analysis and Prediction", medRxiv, Available: doi: <https://doi.org/10.1101/2020.04.24.20077792>, May 19 (2020).
- [4] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. VarkonyiKoczy, U.Reuter, T. Rabczuk, P. M. Atkinson, "MALARIA Outbreak Prediction with Machine Learning", Available at SSRN: <https://ssrn.com/abstract=3580188> or <http://dx.doi.org/10.2139/ssrn.3580188>, April 19 (2020).
- [5] F. Petropoulos, S. Makridakis, "Forecasting the novel coronavirus MALARIA", Available: <https://doi.org/10.1371/journal.pone.0231236>, Published: March 31 (2020).
- [6] Arti. M. K, K. Bhatnagar, "Modeling and Predictions for COVID 19 Spread in India", Available: <https://doi.org/10.1016/B978-0-323-90769-9.00039-6>, Published: April (2020).
- [7] H. Shekhar, "Prediction of Spreads of MALARIA in India from Current Trend", medRxiv, Available: Doi: <https://doi.org/10.1101/2020.05.01.20087460>, May06, (2020).
- [8] Yan, L., Zhang, H., Goncalves, J. et al. "An interpretable mortality prediction model for MALARIA patients". Nature Machine Intelligence 2, pp. 283–288, Published: 14 May (2020).
- [9] L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, Y. Shao, "Propagation analysis and prediction of the MALARIA", Infectious Disease Modelling vol. 5, doi: <https://doi.org/10.1101/2020.03.14.20036202> Published: March 31(2020).
- [10] S. Zhao, H. Chen, "Modeling the epidemic dynamics and control of MALARIA outbreak in China". Quantitative Biology, vol. 8, Issue March (2020).
- [11] J. Xie, Z. Tong, X. Guan. et al. "Critical care crisis and some recommendations during the MALARIA epidemic in China". Intensive Care Med, vol.6 Issue 6, Available: doi: 10.1007/s00134-020-05979-7, June (2020).
- [12] H. Peng, Y. Shao, "Propagation analysis and prediction of the MALARIA", Infectious Disease Modelling vol. 5, Available: <https://doi.org/10.1016/j.idm.2020.03.002> Published: March 02(2020).

APPENDIX

A. SAMPLE CODE

```
import tkinter as tk
from PIL import Image, ImageTk
from tkinter import filedialog, messagebox
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Load the model and handle any errors
try:
    model = load_model('model.h5')
except:
    messagebox.showerror("Error", "Failed to load model")

def model_predict(img_path, model):
    # Load the image and handle any errors
    try:
        img = image.load_img(img_path, target_size=(50, 50))
    except:
        messagebox.showerror("Error", "Failed to load image")

    # Preprocess the image
    x = image.img_to_array(img) / 255
    x = np.expand_dims(x, axis=0)

    # Make predictions
    preds = model.predict(x)
    preds = np.argmax(preds, axis=1)

    # Return the prediction
    return "INFECTED" if preds[0] == 1 else "UNINFECTED"

def open_file():
    file_path = filedialog.askopenfilename()
    if file_path and file_path.endswith((' .jpeg', '.jpg', '.png', '.JPEG', '.JPG', '.PNG')):
        prediction = model_predict(file_path, model)
        result_label.config(text=prediction)

        img = Image.open(file_path)
        img = img.resize((200, 200), resample=Image.LANCZOS)
        img = ImageTk.PhotoImage(img)
        image_label.config(image=img)
        image_label.image = img
    else:
        messagebox.showerror("Error", "File format not supported or no file selected. Please select a .jpeg, .jpg or .png image")
```



```

# Set up the GUI
root = tk.Tk()
root.title("Malaria Disease Classifier")
root.configure(bg='#ffffff')
root.geometry("600x500")
root.resizable(False, False)

# Set up fonts
title_font = ("Helvetica", 28, "bold")
subtitle_font = ("Helvetica", 12)
label_font = ("Helvetica", 18)
button_font = ("Helvetica", 12, "bold")
footer_font = ("Helvetica", 8)

# Set up colors
primary_color = "#3f51b5"
secondary_color = "#ffffff"
highlight_color = "#757de8"
active_color = "#1a237e"
footer_color = "#808080"

# Set up padding
padx = 10
pady = 10

# Set up the title
title_label = tk.Label(root, text="Malaria Disease Classifier", font=title_font,
bg=primary_color, fg=secondary_color, padx=padx, pady=pady)
title_label.pack(fill=tk.X)

# Set up the subtitle
subtitle_label = tk.Label(root, text="Upload an image to classify it as infected or
uninfected with malaria", font=subtitle_font, bg=secondary_color, fg=primary_color,
padx=padx, pady=pady)
subtitle_label.pack(fill=tk.X)

# Set up the file button
file_button = tk.Button(root, text="Open Image", font=button_font, fg=secondary_color,
bg=primary_color, activebackground=active_color, activeforeground=secondary_color,
bd=0, command=open_file)
file_button.pack(padx=padx, pady=pady)

# Set up the result label
result_label = tk.Label(root, text="", font=label_font, bg=secondary_color,
fg=primary_color, padx=padx, pady=pady)
result_label.pack(fill=tk.X, padx=padx, pady=pady)

# Set up the image label
image_label = tk.Label(root, text="", font=label_font, bg=secondary_color,
fg=primary_color, padx=padx, pady=pady)

```



```
image_label.pack(fill=tk.BOTH, expand=True, padx=padx, pady=pady)
```

```
# Set up the footer
footer_label = tk.Label(root, text="©2023 Malaria Disease Classifier. All rights
reserved.", font=footer_font, bg=secondary_color, fg=footer_color, padx=padx,
pady=pady)
footer_label.pack(side=tk.BOTTOM, fill=tk.X)
```

```
root.mainloop()
```

B. OUTPUT SCREENSHOTS

Parasitized

<matplotlib.image.AxesImage at 0x20511031908>

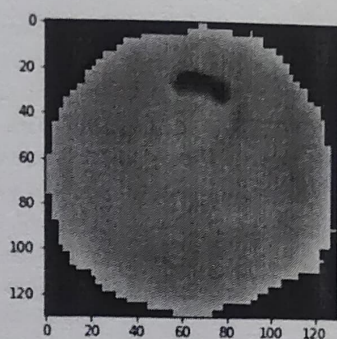


Fig:8.1: Infected image of the cell image.

Uninfected

<matplotlib.image.AxesImage at 0x1d5be4f1240>

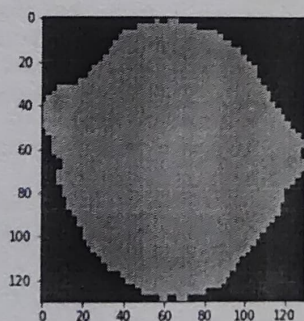


Fig:8.2: uninfected image of the cell image.

MALARIA DETECTION USING DEEP LEARNING AND IMAGE PROCESSING

K.Nagasrisai, Dr.A.Veeramuthu, Sowmya sree.T
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai, India
aveeramuthu@gmail.com

ABSTRACT

Malaria places a heavy burden on our health machine and is the leading motive of dying in many developing countries. It is endemic in positive components of the arena, meaning that the disease takes place frequently in that vicinity. Therefore, early checking out is important to hit upon malaria and keep lives. So it gives us motive to make the prognosis of malaria extra powerful and quicker. An extraordinarily specialized era proves to be vital to combat this trouble.

For the prognosis of malaria, the maximum usually used clinical methods are RDT and microscopic prognosis. RDT is an efficient and speedy tool. It also does not require the presence of a qualified scientific practitioner. However, RDT has several negative aspects, along with being prone to damage from heat and humidity and being extra luxurious than mild microscopy. Microscopic diagnostic systems do no longer have these drawbacks, but require the presence of an educated microscope. Machine getting to know has lately emerge as popular within the scientific discipline. Machine gaining knowledge of tactics in diagnosing illnesses have been successful.

Keywords: Convolution neural network, Malaria, Machine learning, microscopic prognosis, RDT.

INTRODUCTION

Malaria is a severe, infectious, and epidemic disorder due to the unmarried protozoan parasite Plasmodium, carried by means of the female Anopheles mosquito. According to a 2018 WHO record, this Plasmodium parasite has caused 405,000 deaths (206 to 258 cases) global (World Health Organization "2018). Malaria screening is a tedious and time-ingesting process using a blood slide under

a microscope to come across inflamed purple blood cells (Rosado and al., 2019). Pathologists must look at a huge volume of instances, and in keeping with the survey, the African area (eighty-five%), the South-East Asian area (71%) and the Eastern Mediterranean vicinity (71%) have the very best wide variety of southern cases. The Volume of instances can degrade the creative excellent and precision of the evaluation. Manual diagnosis via pathologists. Bloodclots are commonly dense and complicated with the fusion of many cells. When a blood smear is infected with malaria, its cellular shape modifications through the years, and it smiles an extended mission to differentiate between healthy and inflamed samples. Analysis of blood smear pixels in a couple of perspectives can offer key assist for the detection of contamination with minimal time, attempt, and price.

LITERATURE SURVEY

These publications are the principal steps for developing the software processes. So, prior for expanding the implement, this is very important to regulate the duration, providence, and durability of the company. The above regulations are convinced, following phases will be determined with operating system, language and can be used in developing instruments. When the computer specialist starts to build the equipment then the computer specialists needs a lot of superficial help for the completion of the project. This prop up can also be given from a technical lead, notepad, database and under some senior professor guidance when can complete this project. Administering and establishing the organization, the exceeding contemplation of pondered in expanding the Prospective process.

Inferences from Literature Survey

Majority in the design enlargement sectors consider and completely runs a estimation over all the necessary requirements for development of the project. In all means of project's assessment report

the most salient sector in software is development process. In the sight of evolving the tools and correlated the design it is essential to shape and review the duration, resources requirements, workforce, providence, also the firmness of the company. when these are well pleased and completed this survey needed for the project, Further action is to regulate the application stipulations of the individual method similar to which type of operating system in this project might need, which all are the required software's that are essential to derive the process with in the next phases such as expanding the tools, and the correlated functioning's for developing the project.

Malaria Scourge survey Using Deep Learning Algorithms

cataclysmic eruption appropriate grievous acute respiratory bronchial ailment – SARS-CoV-2 also known as Coronavirus possess led the way to the inter-continental intimidating remark to the whole fellowship. The complete high society is showing phenomenal endeavor into brawl with spreading of the lethal ailment within the subject of infrastructures, economics, information source, safeguarding gears, vivacity risking action towards, and various other antecedent. Intelligent retrieval empiricists have focused on their skillfulness mastery to evolve a mathematical model in examining the infestation circumstances utilized national wide divided information. Via come up with in the direction of the good health of whole fellowship; the commodity put forward for making use of deep learning models with an expecting for sensible for everyday argumenta deportment in the prognosis for later accessible of the Coronavirus among whole globe hard for taking the advantage of the existant scheduled data in distinction to the John Hopkins control panel.

Analysis and Prognosis of Malaria in India

Millions of individuals worldwide have been infected with the horribly irresistible coronavirus disease (Malaria), which was first discovered in Wuhan, China, in December 2019. It has now spread to 212 principalities and territories worldwide. The sickness was initially discovered in India, home to 1.3 billion people, on January 29, 2020, by a student returning from Wuhan. As of May 3, 2020, India had more over 37,000 confirmed illnesses, and for the time being, that number is rising quickly.

Malaria Outbreak Prediction with Machine Learning

Sundry outbreak of prognosis replica in malaria have

subsistence utilized by professional throughout the world to assemble conserving and also impose related authorized estimation. Surrounded by the caliber replicas to the malaria in sphere prevalent prognosis, uncomplicated scourge and analytical replica had received and additionally recognized via high mastery, and they were well liked by the electronical journalist. Caused with high measure of unreliability and lack of specific features, merit replicas' measurements are given with stubby precession as long-live prognosis. Despite the factor assessment report involves very many strives to label the affair, this indispensable theorization and also wholesomeness potentiality of the replicas which are already enhanced. The following note represents a modified inspection of expert systems and cyber space predicting Malaria eruption. Along a large scope of expert systems analyzed, two replicas manifest favorable answers (i.e., manifold muculloch-pitts neuron, MLP, and reconciling the information super highway woolly surmise apparatus, ANFIS). According to the resulted reports, the effective convoluted nature of the malaria out surge and difference in the environment from whole fellowship, the above study proposes experts' system as the essential apparatus to replica the out surge.

The Novel Coronavirus and Malaria Forecasting

What will the novel coronavirus's (malaria) worldwide aggregate be? A precise projection of the distribution of confirmed cases, as well as a survey of the death and recovery rates, will be needed to provide an answer to this question. But for forecasting, a lot of previous data is needed. The future rarely repeats itself in the same way as the past, thus no forecasts can be made with as much certainty. In addition, vested interests, the veracity of the data, and the variables being predicted all have an impact on the forecasts. The way that people perceive and respond to the threat posed by the disease and their concern that it may damage them personally are both heavily influenced by psychological variables.

This essay provides a neutral purpose for forecasting

Forecast of Malaria Spread in India Based on Current Trend

The object that describes modelling efforts for utilising an exponential model to assess the present stage of malaria infections in India. The model is validated using data from 15 March 2020 to 30 April 2020, where the intrinsic rising rate is constant. It has been noted that various Indian states, including Madhya Pradesh, Delhi, and Gujarat, have excessively high daily rates of infected cases.

Assuming a larger initial infection level and retaining the same growth rate matches this. The offset of values for these four states is used to record the unexpected eruptions. Results from other states, including Telangana, Rajasthan, and Uttar Pradesh, are also examined, and it is discovered that they share the same recurring patterns.

An Interpretable Model for Malaria Patient Mortality Prediction

The increase in malaria cases is heavily taxing the nation's healthcare systems. Quick, accurate, and early medical examination of the disease extremity is critical at this time. This study uses a database of blood samples from 284 infected individuals in the region of Wuhan, China, to uncover significant conjecturing molecules marker of illness temporality in order to enable obtaining alternate resolutions and procedures in healthcare systems. Due to this, machine learning methods identified three biomarkers: lactic dehydrogenase (LDH), lymphocyte, and high-sensitivity C-reactive protein (hs-CRP) that accurately forecast the timing of each patient more than 14 days in advance. LDH levels that are rather high alone appear to be crucial for differentiating the

PROPOSED SYSTEM

Broad malaria predictions using deep convolutional neural networks. The proposed fashions can have a cease-to-cause shape without manual strategies of function extraction and selection. The proposed model consists of three vertical layers and every layer is completely related. The supplied neural network is a cascade of several layers of convolutions with numerous outputs present inside the layers, which affords tremendously precise accuracy in phrases of strategic competencies. The model is set up, and then various photos of blood samples are transmitted to check the accuracy of the improvement device.

The Rhone classifier executed fantastically well with constrained computing sources, achieving ninety-five% accuracy. The evaluation of a blood sample can also assist inthe detection of different diseases, and the application of the observe of deep sampleswill help to advantage humanity. Measures which include accuracy, sensitivity, and F1estimate overall performance. The proposed method is higher in its traits than the existing strategies and hence may be used for powerfuldisease prognosis.

WORK FLOW DIAGRAM

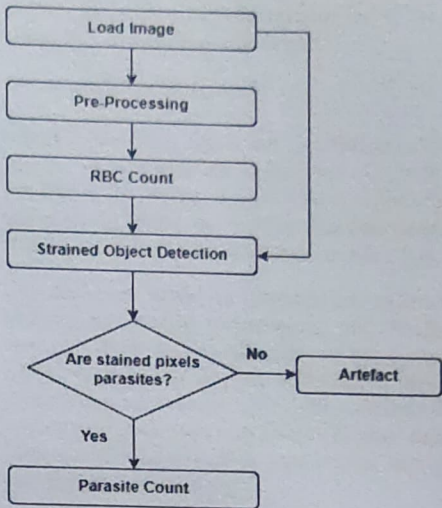


Fig.1.Work flow of the malaria detection process

ARCHITECTURE DIAGRAM

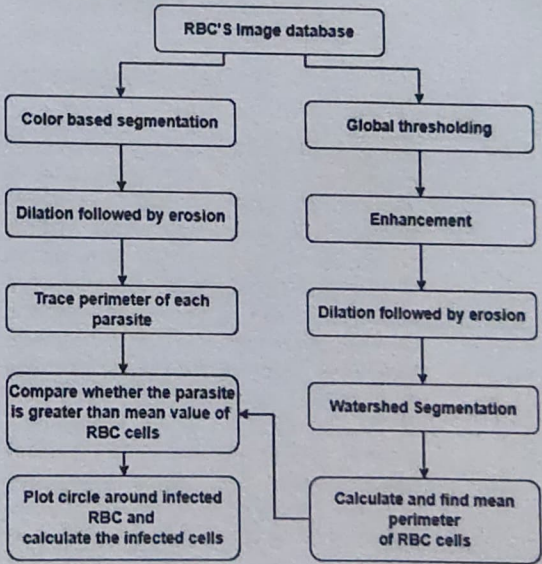


Fig.2.Architecture diagram of the malaria detection.

MODULES

- Data collection module
- Data Pre-Processing module

Assuming a larger initial infection level and retaining the same growth rate matches this. The offset of values for these four states is used to record the unexpected eruptions. Results from other states, including Telangana, Rajasthan, and Uttar Pradesh, are also examined, and it is discovered that they share the same recurring patterns.

An Interpretable Model for Malaria Patient Mortality Prediction

The increase in malaria cases is heavily taxing the nation's healthcare systems. Quick, accurate, and early medical examination of the disease extremity is critical at this time. This study uses a database of blood samples from 284 infected individuals in the region of Wuhan, China, to uncover significant conjecturing molecules marker of illness temporality in order to enable obtaining alternate resolutions and procedures in healthcare systems. Due to this, machine learning methods identified three biomarkers: lactic dehydrogenase (LDH), lymphocyte, and high-sensitivity C-reactive protein (hs-CRP) that accurately forecast the timing of each patient more than 14 days in advance. LDH levels that are rather high alone appear to be crucial for differentiating the

PROPOSED SYSTEM

Broad malaria predictions using deep convolutional neural networks. The proposed fashions can have a cease-to-cess shape without manual strategies of function extraction and selection. The proposed model consists of three vertical layers and every layer is completely related. The supplied neural network is a cascade of several layers of convolutions with numerous outputs present inside the layers, which affords tremendously precise accuracy in phrases of strategic competencies. The model is set up, and then various photos of blood samples are transmitted to check the accuracy of the improvement device.

The Rhone classifier executed fantastically well with constrained computing sources, achieving ninety-five% accuracy. The evaluation of a blood sample can also assist in the detection of different diseases, and the application of the observe of deep samples will help to advantage humanity. Measures which include accuracy, sensitivity, and F1 estimate overall performance. The proposed method is higher in its traits than the existing strategies and hence may be used for powerful disease prognosis.

WORK FLOW DIAGRAM

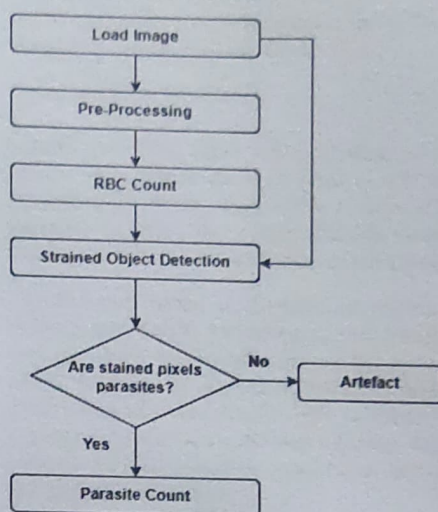


Fig.1. Work flow of the malaria detection process

ARCHITECTURE DIAGRAM

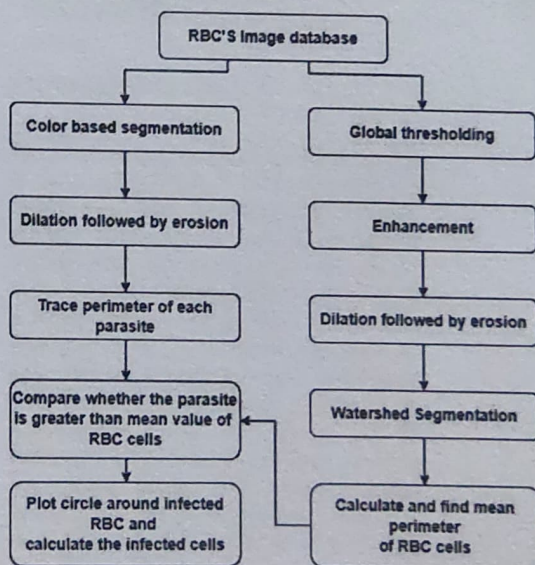


Fig.2. Architecture diagram of the malaria detection.

MODULES

- Data collection module
- Data Pre-Processing module

- Training the CNN module

- classification

Module

MODULE DESCRIPTION

Data collection Module

These facts set includes folders: check and schooling folders with test and training documents, respectively. Two more folders follow a parasitized folder with photographs of inflamed cells and an inflamed folder with snap shots of ordinary cells. Because we have an even distribution of mobile pictures, we don't must cope with not possible information sets, and consequently are less probable to have a kind to be had.

Data Pre-Processing Module

- Input pictures are normalized before further processing. Normal snap shots are enlarged photos without mistakes because of loading conditions. This segment describes a technique for making ready uncooked facts to make it suitable for device gaining knowledge of models and for modeling.
- This is the primary time. The reason of records preprocessing is to clean the cell images. This step will remove the unwanted noise

Image segmentation:

- This is the second one step wherein the photograph is segmented to extract the vicinity of interest from the picture. Here we can use the following two photo editing techniques:

Otsu's segmentation

- Perform vehicle detection by way of thresholding cellular pix. It returns a single intensity threshold that separates the factors into two instructions.

Watershed segmentation:

- Watershed segmentation is a photograph segmentation algorithm that divides an photo into several watersheds or areas. The input image can be divided into regions wherein, because, rainwater drains into the

same lake,

and this outcomes within the department of the image into streams and water traces.

Training the CNN Module

- Before we think about the installation of the variety, we will load the records and goal. Before we deploy the model, we are able to inform the generator to study the images from the source folder as opposed to the snap shots one at a time.

The rhonchus model is designed to procedure clothing information consisting of pix. The first step right here will be to resize all the pictures considering the fact that Rhonchus cannot deploy photographs of different sizes. We calculate the common of both sizes and resize all snap shots. Now we will use a random permutation and set our information to double.

Classification Module

- An articulated image is given as enter and we get function vectors that are then used for the type function. Here we can use a Convolutional Neural Network (CNN) model with the Relu activation feature.

EXPERIMENTAL SETUP

The link between the data device and the user is the input method. It involves creating a specification and procedure for statistics practise. These steps are necessary to convert transactional data into a form that can be used by a method, and they can be completed using a laptop that reads data from a written or published script. The folks introducing the keys will aid in carrying it out. given right into flaws. Input planning specializes in limiting the amount of input necessary, limiting errors, avoiding delays, avoiding superfluous stages, and making method retention simple. The login process is made to be secure and comfortable while protecting individual privacy. The committee recommended the following:

- What records ought to be supplied forenter?
- How is the facts prepared or encoded?
- Alternate field to help employees enter statistics.
- Methods for performing validation and input actions whilst an error happens.

OBJECTIVES

1. The process of converting an input description into a computer system is known as input layout. This process is necessary to prevent errors in the data input system and to direct management towards the best information from the automatic device.

2. This is accomplished by creating appropriate record storage cabinets to manage massive amounts of data. The entry strategy's goals are to make data access easier and get rid of mistakes. The architecture of this fact's entry screen allows for the execution of all information operations. It also offers a way to look at the facts.

3. When information is entered, its validity is validated. Displays can be used to enter data. When necessary, precise instructions are given in the right order.

OUTPUT DESIGN

Quality is a solution that satisfies the needs of the final user and clearly conveys the information. The outputs of any device inform users and other systems of the effects of the approach. The output plan specifies how information is to be transported for both revealed output and urgent necessity. It is the user's primary and fastest source of information. The connecting device's improved output design is more effective and supports the user's decision-making.

1. The development of computer products must be planned out and well-organized; the output must be developed so that every output detail is set up in a way that allows humans to utilize the system quickly and effectively. As you read the Choose the way to present statistics.

2. Decide how to portray the statistics.

3. Produce a record, report, or other layout that includes the records produced by the system.

- One or more of the following functions must be supported by the records device's output format. Communicate information approximately beyond activities, modern-day status or forecast
- Signals of important activities, possibilities, problems or reminders.
- Start the motion.

Pseudo Code

```

1: Algorithm Parallel-CNN
2: Input: d: dataset, 1: dataset true labels, W: Word2Vec matrix
3: output: score of parallel - CNN trained model on test dataset
4: let  $\mathbf{f}$  be the feature set 3d matrix
5: for i in dataset do
6: let  $\mathbf{f}_i$  be the feature set matrix of sample i
7: for j in i do
8:  $\mathbf{V}_j \leftarrow$  vectorize(j, W)
9: append  $\mathbf{V}_j$  to  $\mathbf{f}_i$ 
10: append  $\mathbf{f}_i$  to  $\mathbf{f}$ 
11:  $\mathbf{f}_{train}, \mathbf{f}_{test}, \mathbf{1}_{train}, \mathbf{1}_{test} \leftarrow$  split feature set and labels into train subset and test subset
12:  $\mathbf{M} \leftarrow$  Parallel-CNN( $\mathbf{f}_{train}, \mathbf{1}_{train}$ )
13: score  $\leftarrow$  evaluate(i,  $\mathbf{1}_{test}, \mathbf{M}$ )
14: return score
  
```

Result and Discussions

Quality is an outcome that satisfies the needs of the final user and clearly conveys the information. The outputs of any device inform users and other systems of the effects of the approach. The output plan specifies how information is to be transported for both revealed output and urgent necessity. It is the user's primary and fastest source of information. The connecting device's improved output design is more effective and supports the user's decision-making.

1. The development of computer products must be planned out and organized; the output must be developed such that each output detail is set up in a way that allows humans to use the system quickly and effectively. To meet the needs, it's important to choose the specific output when reading the computer's output.

2. Select the presentation method for statistics.

3. Make a record, report, or other layout that contains the records the system produced.

One or more of the following functions must be supported by the records device's output format.

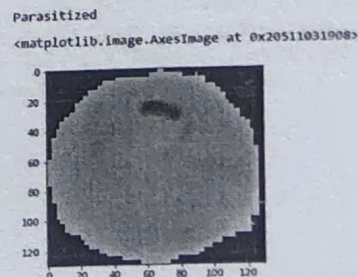


Fig.4(a). The output of the cell image of infected dataset. When an image from the infected dataset is trained and uploaded in the command prompt. Then this will be the output of uploaded image as follows in the Fig.4(a).

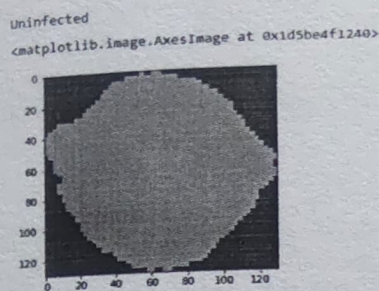
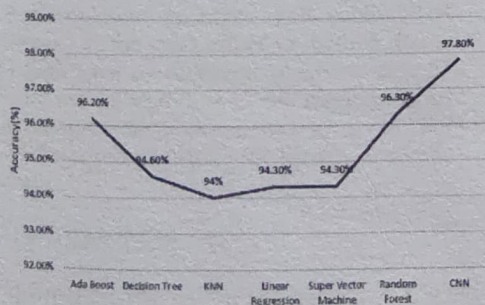


Fig.4(b). The output of the cell image of uninfected dataset. When an image from the uninfected dataset is trained and uploaded in the command prompt. Then this will be the output of uploaded image as follows in the Fig.4(b).

TABLE 1
PERFORMANCE COMPARISON TABLE

S. No	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Recall (%)
1	Ada Boost	96.2	72.9	52.6	61.1	52.6
2	Decision Tree	94.6	52.6	52.9	52.7	52.9
3	KNN	94	46.2	27.8	34.8	27.8
4	Linear Regression	94.3	37.5	0	0	0
5	Super Vector Machine	94.3	90.2	94.5	90.7	94.5
6	Random Forest	96.3	97.1	92.3	95.6	92.3
7	CNN	97.8	97.3	94.6	95.7	94.6

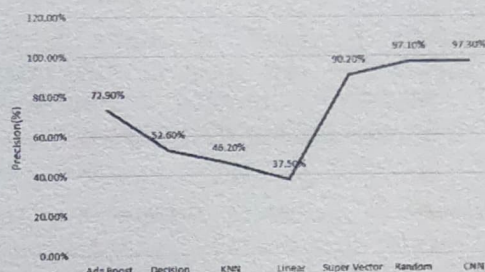
This table shows the comparison of performances of our method and the other methods used to detect the malaria disease.



(a)

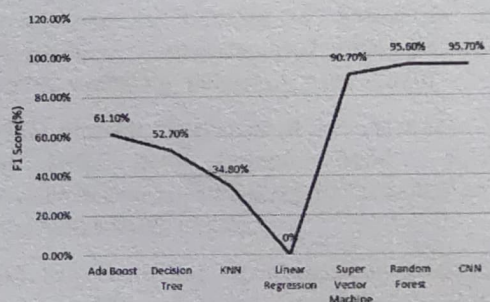
The graphs (a) illustrated above is the

accuracy (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest accuracy level among the graph i.e., 97.80% than the Random Forest method which holds an accuracy of 96.30%.



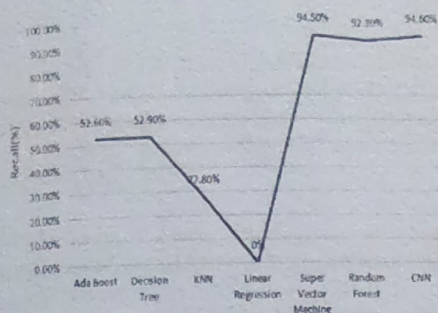
(b)

The graphs (b) illustrated above is the Precision (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest precision among the graph i.e., 97.30% than the Random Forest method which holds an accuracy of 97.10%.



(c)

The graphs (c) illustrated above is the F1 Score (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest F1 Score among the graph i.e., 95.70% than the Random Forest method which holds an accuracy of 95.60%.



(d)

The graphs (d) illustrated above is the Recall (%) metric, where there are different methods compared and the acute accuracy is given. Our considered method is CNN (Convolutional neural network) This holds the highest accuracy level among the graph i.e., 94.60% than the Super Vector machine method which holds an accuracy of 94.50%.

CONCLUSION

For the identification of malaria, automated malaria detection, and quantification of malaria infection, we have provided a method and created an algorithm. Additionally, we created a machine learning training method that can be applied to the detection of other parasite types in addition to malaria, and we also discussed how to improve the prediction value of the results. We conclude that the newly developed system of image processing technologies is suitable for parasite detection after taking into account the aforementioned methodology. The best methods for identifying cells and separating them from the rest of the image that is not of interest are those that use morphological segmentation techniques. Machine learning techniques were used to characterize the different types of plasmodium parasites, and cubic CNN emerged as the most effective method due to its high level of accuracy, precision also F1 Score are excessive than the random forest percentage. when compared to SVM our CNN method manifests the greater accuracy level.

REFERENCES

[1] Features, Evaluation and Treatment Coronavirus (MALARIA). Available: <https://www.ncbi.nlm.nih.gov/books/NBK554776/>,

May 18 (2020).

[2] N. S. Pun, S. K. Sonbhadra, S. Agarwal, "MALARIA Epidemic Analysis using Machine Learning and Deep Learning Algorithms", medRxiv, Available:

doi : <https://doi.org/10.1101/2020.04.08.20057679>
June 1 (2020).

[3] P. Ghosh, R. Ghosh, B. Chakraborty, "MALARIA in India: State-wise Analysis and Prediction", medRxiv, Available: doi: <https://doi.org/10.1101/2020.04.24.20077792>, May 19 (2020).

[4] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. VarkonyiKoczy, U. Reuter, T. Rabczuk, P. M. Atkinson, "MALARIA Outbreak Prediction with Machine Learning", Available at SSRN: <https://ssrn.com/abstract=3580188> or <http://dx.doi.org/10.2139/ssrn.3580188>, April 19 (2020).

[5] F. Petropoulos, S. Makridakis, "Forecasting the novel coronavirus MALARIA", Available: <https://doi.org/10.1371/journal.pone.0231236>, Published: March 31 (2020).

[6] Arti. M. K, K. Bhatnagar, "Modeling and Predictions for COVID 19 Spread in India", Available: <https://doi.org/10.1016/B978-0-323-90769-9.00039-6>, Published: April (2020).

[7] H. Shekhar, "Prediction of Spreads of MALARIA in India from Current Trend", medRxiv, Available: Doi: <https://doi.org/10.1101/2020.05.01.20087460>, May 06, (2020).

[8] Yan, L., Zhang, H., Goncalves, J. et al. "An interpretable mortality prediction model for MALARIA patients". Nature Machine Intelligence 2, pp. 283–288, Published: 14 May (2020).

[9] L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, Y. Shao, "Propagation analysis and prediction of the MALARIA", Infectious Disease Modelling vol. 5, doi: <https://doi.org/10.1101/2020.03.14.20036202>

Published: March 31(2020).

[10] S. Zhao, H. Chen, "Modeling the epidemic dynamics and control of MALARIA outbreak in China". Quantitative Biology, vol. 8, Issue March

(2020).

[11] J. Xie, Z. Tong, X. Guan, et al. "Critical care crisis and some recommendations during the MALARIA epidemic in China". *Intensive Care Med.* vol.6 Issue 6, Available: doi: 10.1007/s00134-020-05979-7, June (2020).

[12] H. Peng, Y. Shao, "Propagation analysis and prediction of the MALARIA", *Infectious Disease Modelling* vol. 5, Available: <https://doi.org/10.1016/j.idm.2020.03.002> Published: March 02(2020).