DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING, MACHINE LEARNING AND DEEP LEARNING

A PROJECT REPORT

Submitted by

SHYAM GANESH.M (310616106131)

SWARNA.M (310616106150)

VISHAL KANNA.N.M (310616106169)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

EASWARI ENGINEERING COLLEGE, CHENNAI

(Autonomous Institution)

affiliated to

ANNA UNIVERSITY:: CHENNAI - 600 025

MAY 2020

EASWARI ENGINEERING COLLEGE, CHENNAI

(AUTONOMOUS INSTITUTION)

AFFILIATED TO ANNA UNIVERSITY, CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING, MACHINE LEARNING AND DEEP LEARNING"** is the bonafide work of **"M.SHYAM GANESH (310616106131), M.SWARNA (310616106150), N.M.VISHAL KANNA (310616106169)"** who carried out the project work under my supervision.

Dr. M. DEVARAJU
HEAD OF THE DEPARTMENT
Professor and Head of the
Department of Electronics and
Communication Engineering,
Easwari Engineering College,
Ramapuram, Chennai 600089

Mrs. A. USHA
SUPERVISOR
Associate professor,
Department of Electronics and
Communication Engineering,
Easwari Engineering College,
Ramapuram, Chennai 600089

Submitted for Semester Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are indebted to many people who helped us complete our project and assisted us since the inception of the idea till the day we documented it and even after that. Firstly, we would like to express our gratitude to **Dr. T.R. PAARIVENDHAR, Founder Chairman, SRM Group of Institutions,** without whose support our project could not have been completed successfully.

We would also like to thank **Dr. R. SHIVA KUMAR, M.D Ph.D., Chairman, SRM Group of Institutions,** Ramapuram Campus for providing us the required assistance and our **Principal, Dr. R. S. Kumar, B.E., M.Tech., Ph.D.,** for his continuous encouragement that led us to where we are right now, satisfied, having gotten what we had worked for.

We feel glad to express our sincere thanks to **Dr. M. DEVARAJU, M.Tech., Ph.D., Professor and Head of the Department**, Electronics and Communication Engineering for giving us the necessary lab facilities and continuous encouragement for the completion of the project.

We take the privilege to extend our hearty thanks to our Project Coordinators **Dr. RESMI R NAIR, M.E., Ph.D., Associate Professor** and **Dr. S. R. SRIRAM, M.E., Ph.D., Assistant Professor** for their suggestions, support and encouragement towards the completion of the project with perfection.

We are thankful to our supervisor **Mrs. A. USHA, M.E., (Ph.D.) Associate Professor** for her intellectual guidance and valuable suggestions in making the project a successful one and also our sincere thanks to all **our panel members** for their valuable suggestions.

ABSTRACT

Malaria is a dreadful disease in the hematological field causing millions of deaths. Fast diagnosis and acute treatment of malaria is important to reduce the death rate. Hence, the rapid diagnosing and proper medication is the intense need of our era. Malaria analysis is based on microscopic examination of blood films. This process becomes problematic when cases are reported from far-flung rural areas as experts may not be present in these areas to perform diagnosis. Automation of the diagnostic process with the use of an intelligent system that would recognize malaria parasites could save this problem. Image processing techniques, SVM and CNN classifiers are used for detection and identification of malarial parasites from microscopic images of Giemsa stained thin blood smears. Total of 400 images of giesma stained thin blood smears were collected as a dataset. MATLAB (Matrix Laboratory) software is used for this image processing and classification processes. A comparative analysis of SVM (Support Vector Machine), CNN (Convolutional Neural Network) is carried out for recognition of extracted SIFT (Scale Invariant Feature Transform) features. A recognition efficiency of 94% is obtained from SVM and 98% from CNN. Hence, CNN provides better efficiency when compared to SVM.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	APPLICATION PROGRAM INTERFACE
CNN	CONVOLUTIONAL NEURAL NETWORK
DFT	DISCRETE FOURIER TRANSFORM
ECOC	ERROR CORRECTING OUTPUT CODES
FFT	FAST FOURIER TRANSFORM
FN	FALSE NEGATIVE
FP	FALSE POSITIVE
GLCM	GRAY LEVEL CO-OCCURRENCE MATRIX
HSV	HUE SATURATION & VALUE
IDM	INVERSE DIFFERENCE MOMENT
LBP	LOCAL BINARY PATTERNS
LSTM	LONG SHORT TERM MEMORY
MATLAB	MATRIX LABORATORY
MLP	MULTI-LAYER PERCEPTRON
NLP	NATURAL LANGUAGE PROCESSING
QBC	QUANTITATIVE BUFFY COAT
QP	QUADRATIC PROGRAMMING
RAM	RANDOM ACCESS MEMORY

RBC	RED BLOOD CELL
SIANN	SPACE INVARIANT ARTIFICIAL NEURAL NETWORKS
SIFT	SCALE INVARIANT FEATURE TRANSFORM
SMO	SEQUENTIAL MINIMAL OPTIMIZATION
SVM	SUPPORT VECTOR MACHINE
TN	TRUE NEGATIVE
TP	TRUE POSITIVE
WBC	WHITE BLOOD CELL
WHO	WORLD HEALTH ORGANISATION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Malaria is a dreadful infectious disease which affects humans and other animals. It is a mosquito borne and tropical vector borne disease that cause sufferings and millions of deaths in different parts of the world. According to the World Health Organization (WHO) estimates, there were 212 million new cases of malaria worldwide in 2015. Malaria is the most dangerous: one among the entire mosquito borne diseases. Automated identification and classification of malarial parasites in thin blood smear images is a research project undertaken in the field of medical diagnosis and pathological analysis, to detect the presence of malarial parasite, classification into malaria infected (or not) and analyzing the efficient method of malarial diagnosis by implementing the work using machine learning and deep learning algorithms. An efficient, diagnostic platform for a quantitative analysis of malaria infection needs to be developed.

The main objectives of this method are developing automated and accurate method to find presence of malarial parasite in blood and identifying the efficient method for malaria diagnosis. The existing methods for malaria diagnosis are inaccurate. This is the motivation for the project. Conventional microscopy is the existing method for malaria diagnosis. The disadvantages of this method are that method is time consuming, labour intensive and results depends on the skills of the researcher. Here, lies the scope of this work. This project will progress through the following steps of image processing: image acquisition, pre-processing, segmentation, feature extraction and finally classification. Pre-processing includes removing the noise in

the image, segmentation to separate foreground and background pixels, feature extraction to extract the features for further classification or identification of the malarial parasite.

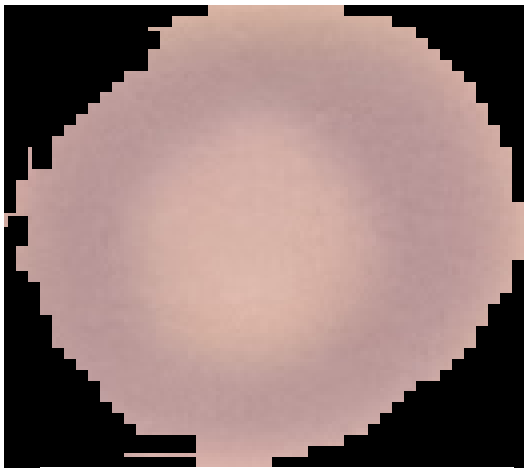


Fig. 1.1 Healthy blood cell

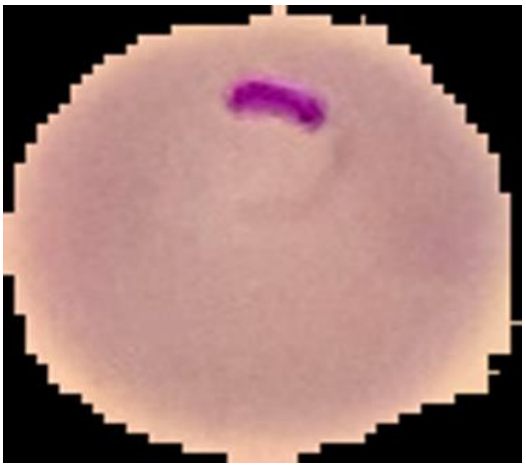


Fig. 1.2 Malaria infected blood cell

1.2 MALARIAL PARASITES

Plasmodium *vivax* is a protozoal parasite and a human pathogen. This parasite is the most frequent and widely distributed cause of recurring malaria. Although it is less virulent than Plasmodium *falciparum*, the deadliest of the five human malaria parasites, P. *vivax* malaria infections can lead to severe disease and death, often due to splenomegaly (a pathologically enlarged spleen). P. *vivax* is carried by the female *Anopheles* mosquito; the males do not bite.

Plasmodium *falciparum* is an unicellular protozoan parasite of humans, and the deadliest species of Plasmodium that causes malaria in humans. The parasite is transmitted through the bite of a female *Anopheles* mosquito and causes the disease's most dangerous form, falciparum malaria. It is responsible for around 50% of all malaria cases. P. *falciparum* is therefore regarded as the deadliest parasite in humans, causing 405,000 deaths in 2018. It is also associated with the development of blood cancer (Burkitt's lymphoma).

Plasmodium *malariae* is a parasitic protozoan that causes malaria in human beings. It is one of the several species of *Plasmodium* parasites that infects humans, including, Plasmodium *falciparum* and Plasmodium *vivax*, responsible for most malarial infections. Found worldwide, it causes a so-called "benign malaria", not nearly as dangerous as that produced by P. *falciparum* or P. *vivax*. The signs include fevers that recur at approximately three-day intervals — a *quartan fever* or *quartan malaria* — longer than the two-day (tertian) intervals of the other malarial parasites.

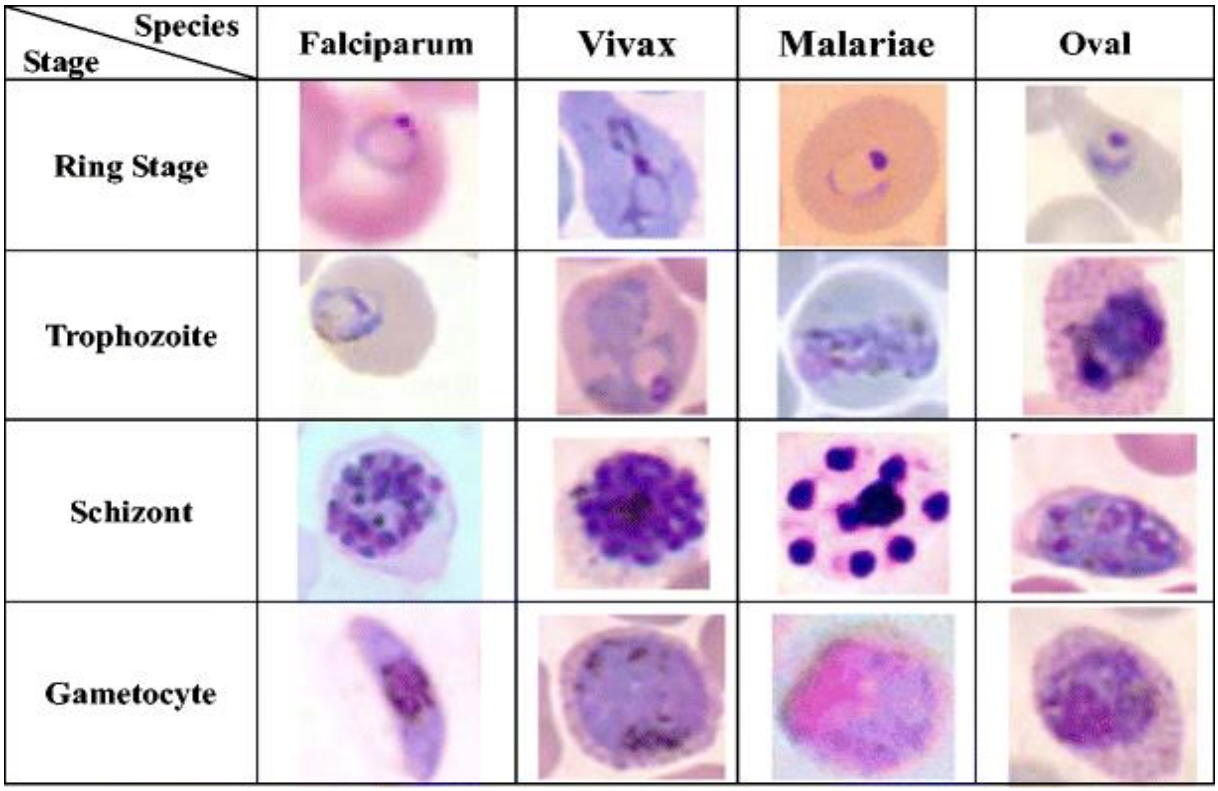


Fig. 1.3 Stages in the growth and development of a malarial parasite in a blood cell

Source: https://www.biologydiscussion.com/parasitology/parasiticprotozoa/histology-of-malarial-parasite-sporozoa/62006

Plasmodium *ovale* is a species of parasitic protozoa that causes tertian malaria in humans. It is one of several species of Plasmodium parasites that infect humans including Plasmodium *falciparum* and Plasmodium *vivax* which are responsible for

most malarial infection. It is rare compared to these two parasites, and substantially less dangerous than P. *falciparum.*

Different stages in the growth and development of these parasites is depicted above.

Ring stage trophozoites mature into schizonts, which rupture releasing merozoites. Some parasites differentiate into sexual erythrocytic stages (gametocytes). Blood stage parasites are responsible for the clinical manifestations of the disease. Ring-form trophozoites (rings) of Plasmodium *falciparum* are often thin and delicate, measuring on average 1/5 the diameter of the red blood cell. They may be found on the periphery of the RBC (accolé, appliqué) and multiply-infected RBCs are not uncommon.

Trophozoite stage is the activated, feeding stage in the life cycle of certain protozoa such as malaria-causing Plasmodium *falciparum* and those of the Giardia group. The trophozoite stage of development is critical for the parasite to undergo morphological changes, grow in size, and remodel the host red blood cell (RBC) to suit its development and release of new infectious merozoite forms into circulation. The sporozoites grow and multiply in the liver to become merozoites. These merozoites invade the erythrocytes (RBCs) to form trophozoites, schizonts and gametocytes, during which the symptoms of malaria are produced.

At the **schizont stage**, the parasite replicates its DNA multiple times and multiple mitotic divisions occur asynchronously. The red blood cells are ruptured by the merozoites. The liberated merozoites invade fresh erythrocytes. A free merozoite is in the bloodstream for roughly 60 seconds before it enters another erythrocyte. This gives rise to the characteristic clinical manifestations of falciparum malaria. The malaria parasite life cycle involves two hosts. During a blood meal, a malaria-infected female Anopheles mosquito inoculates sporozoites into the human host. Sporozoites infect liver cells and mature into schizonts, which rupture and release

merozoites. The merozoites break out of the liver and re-enter the bloodstream, where they invade red blood cells, grow and divide further, and destroy the blood cells in the process.

Gametocytes, the precursors of male and female gametes, of malaria parasites are formed in the human host through the developmental switch from asexual replication in erythrocytes. Although gametocytes are not responsible for clinical symptoms, they ensure the transmission of malaria to another host. Upon taking a blood meal, gametocytes are transferred to a mosquito's midgut lumen where they differentiate into male and female gametes. The presence of gametocytes in circulation of infected individuals is imperative for malaria to remain endemic in a given community. Male and female gametocytes are the components of the malaria parasite life cycle which are taken up from an infected host bloodstream by mosquitoes and thus, mediate disease transmission. These gamete precursors are quite distinct from their asexual blood stage counterparts and this is reflected in their distinct patterns of gene expression, cellular development, and metabolism. Plasmodium *falciparum* is most distinguishable for its sickle-shaped gametocyte. This species is the most pathogenic of all species, posing high rate of complications and fatality if left untreated. The trophozoites of this species appear as tiny ring-forms usually attached near the walls of the erythrocytes.

1.3 SYSTEM ARCHITECTURE

System architecture used for malaria parasite detection involves the following main steps: image acquisition (input image), pre-processing, segmentation, feature extraction, classification and final result (identification of the malaria causing parasite). MATLAB is used in all system steps.

Image acquisition is the initial stage where a dataset containing malaria infected blood sample images is fed as input to the subjected classifier.

Pre-processing helps to remove unwanted objects and noise from the image to facilitate image segmentation into meaningful regions.

Image segmentation is the process of partitioning the image into different segments. The image is generally partitioned into foreground and background regions in order to isolate the RBCs. Simplest approach for segmentation is used here which is the selection of a suitable intensity threshold.

Feature extraction is applied to extract relevant features from images. Here, SIFT is used for extracting features. The Scale-Invariant Feature Transform (SIFT) is an algorithm in computer vision to detect and describe local features in images.

Classification is the final stage where the features of input image is compared with that of the infected RBC image features and then classified. For this, the classifier is trained by providing the features extracted in the feature extraction stage.

Identification process occurs with the help of outcomes from the classification system using SVM and CNN algorithms.

Widely, these two types of algorithms are used in the detection of malarial parasite in blood samples. In order to obtain a gist of the overall idea, both the algorithms used are explained.

(1) Support Vector Machine (SVM):

Support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outlier detection. Here, the classifier is trained to enable it to classify the input images into infected and normal RBC image.

(2) Convolutional Neural Network (CNN):

Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural networks that use a variation of multilayer perceptron designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered.

Final result i.e., the species of the malarial parasite in the given input image (thin blood smear) is identified and the results produced by both the SVM and CNN classifiers are compared for accuracy.

CHAPTER 2

LITERATURE SURVEY

Title: ABO Blood Group Detection Based on Image Processing Technology

Author: Min Liu, Nian Chen, Shi Chen, Wei-wei Fu, Yue-fang Dong and Zhe Zhou

Publisher: IEEE 2nd International Conference on Image, Vision and Computing (ICIVC)

Year: 2017

Abstract: Rapid and accurate determine of blood types is very important during emergency situation before administering a blood transfusion. At present, the method based on image recognition technology to quickly determine blood type has been widely used in the automated blood analyzer. A fast, accurate and robust blood group judgment method based on the image features of ABO blood group rapid analyzer has been proposed. Firstly, the image of the disk region is segmented and identified automatically. Then, the median filter is used to suppress the noise to get the best approximation of the original image. Then, the characteristic parameters of ABO blood group are extracted according to the gray level distribution of the image. Finally, combined with the agglutination reaction between antigen and antibody, the final blood group was determined. The experimental results show that this method can quickly and accurately classify the ABO blood group, and basically meet the requirements of the automatic rapid blood group analyzer.

Inference: A fast, accurate and robust blood group judgment method based on the image features of ABO blood group is proposed. The median filter is used to

suppress the noise to get the best approximation of the original image. The characteristic parameters of ABO blood group are extracted according to the gray level distribution of the image.

Title: Image Processing Based Detection & Classification of Blood Group Using Color Images

Author: Abubakar Yamin, Faisal Irnran , Syed Hassan Tanvir and Usman Akbar

Publisher: IEEE International Conference on Communication, Computing and Digital Systems (C-CODE)

Year: 2017

Abstract: Domain of image processing is progressing a lot and has achieved tremendous milestones. Image processing is helping in many ways for the researchers to achieve their goals especially in security and medical fields. Detection of blood group in disaster or remote areas where expert is unavailable is a challenge. A system which will detect blood group using image processing techniques has been proposed. Steps to detect the type of blood group using image processing techniques are discussed. Successful results have been obtained and accuracy of the proposed system is optimal.

Inference: A system which will detect blood group using image processing techniques is proposed. Steps to detect the type of blood group using image processing techniques are pre-processing techniques, HSV Luminance and morphological operations.

Title: Design a new algorithm to count white blood cells for classification Leukemic blood image using machine vision system

Author: Alireza Karimian and Zahra Khandan Khadem Alreza

Publisher: IEEE 6th International Conference on Computer and Knowledge Engineering (ICCKE)

Year: 2016

Abstract: Data extraction from white blood cells may cause problems such as loosing form, dimensions and edges. A complete and automatic method to identify and classify white blood cells using microscopic images has been presented. In the first step, white blood cells are identified using color space conversion models. Then leukocytes group are separated using division of watershed conversion. In the next step, image cleanup is done and all leukocytes available on the edge of images and abnormal components are removed. This is accomplished by cutting the image with the smallest rectangle that has connected components. In the last step, feature extraction is performed which supports the pathologists to have the best interpretation. At the end, the proposed method was examined by a database belonging to Imam Reza (AS) hospital in Mashhad, consisting of 29 images of blood cells and showed the accuracy of 93% in the detection of white blood cells, using MATLAB software.

Inference: White blood cells are identified using color space conversion models. Leukocytes group are separated using division of watershed conversion and image cleanup is done. The proposed method showed an accuracy of 93% in the detection of white blood cells.

Title: Real Time Blood Image Processing Application for Malaria Diagnosis Using Mobile Phones

Author: Corentin Dallet, Izzet Kale and Saumya Kareem

Publisher: IEEE International Symposium on Circuits and Systems (ISCAS)

Year: 2014

Abstract: A fast and reliable mobile phone Android application platform for blood image analysis and malaria diagnosis from Giemsa stained thin blood film images has been proposed. The application is based on novel Annular Ring Ratio Method which is already implemented, tested and validated in MATLAB. The method detects the blood components such as the Red Blood Cells (RBCs), White Blood Cells (WBCs) and identifies the parasites in the infected RBCs. The application also recognizes the different life stages of the parasites and calculates the parasitemia which is a measure of the extent of infection.

Inference: A fast and reliable mobile phone Android application platform for blood image analysis is developed. The application is based on novel Annular Ring Ratio Method, tested and validated in MATLAB. The method detects the blood components such as the Red Blood Cells (RBCs), White Blood Cells (WBCs), parasitemia of the parasites in the infected RBCs.

Title: A Novel Method to Count the Red Blood Cells in Thin Blood Films

Author: I.Kale, S.Kareem and R.C.S Morling

Publisher: IEEE International Symposium on Circuits and Systems (ISCAS)

Year: 2011

Abstract: A novel idea to identify the total number of red blood cells (RBCs) as well as their location in a Giemsa stained thin blood film image has been proposed. This work is being undertaken as a part of developing an automated malaria parasite detection system by scanning a photograph of thin blood film in order to evaluate the parasitemia of the blood. Not only will this method eliminate the segmentation procedures that are normally used to segment the cells in the microscopic image, but also avoids any image pre-processing to deal with non-uniform illumination prior to cell detection. The method utilizes basic knowledge on cell structure and brightness of the components due to Giemsa staining of the sample, then, detects and locates the RBCs in the image.

Inference: Unique idea to identify the total number of red blood cells (RBCs) and their location in a Giemsa stained thin blood film image is presented. The method utilizes basic knowledge on cell structure and brightness of the components to detect the RBCs in the image. Eliminates the segmentation procedures used to segment the cells in the microscopic image. It avoids image pre-processing to deal with non-uniform illumination prior to cell detection.

CHAPTER 3

EXISTING TECHNIQUE

The existing techniques to classify and identify the malarial parasites which are present in the blood cells are:

- Conventional microscopy (traditional method)

Other techniques are **computer-aided** which uses some of the following methods in the diagnosis process.

- Global threshold
- Morphological techniques
- Watershed algorithm
- Haralick textural feature extraction
- Multivariate regression models
- Bayes decision theory etc.

3.1 CONVENTIONAL MICROSCOPY

It is one of the existing methods for malaria diagnosis. The microscopic tests involve staining and direct visualization of the parasite under the microscope. For more than hundred years, the direct microscopic visualization of the parasite on the thick and/or thin blood smears has been the accepted method for the diagnosis of malaria in most settings, from the clinical laboratory to the field surveys. The careful examination of a well-prepared and well-stained blood film currently remains the “gold standard” for malaria diagnosis.

The most commonly used microscopic tests includes:

- The peripheral smear study
- Quantitative Buffy Coat (QBC) test

3.2 TYPES OF MICROSCOPY

Malaria microscopy allows the identification of different malaria-causing parasites (P. *falciparum*, P. *vivax*, P. *malariae* and P. *ovale*), their various parasite stages, including gametocytes, and the quantification of parasite density to monitor response to treatment. Microscopy is the method of choice for the investigation of malaria treatment failures. Giemsa is the classical stain used for malaria microscopy, and diagnosis requires examination of both thin and thick films from the same patient.

3.2.1 LIGHT MICROSCOPY

Light microscopy is the diagnostic standard against which other diagnostic methods have traditionally been compared. The disadvantages of this method are that it is time consuming, labour intensive and results depend on the skills of the microscopist. Two different types of blood smears are typically prepared for malaria diagnosis: thick and thin smears. A thick smear is used to detect the presence of parasites in a drop of blood. A thin blood smear is a drop of blood that is spread across a large area of the slide. Thin blood smear helps doctors discover what species of malaria is causing the infection. Thick smear allows a more efficient detection of parasites than thin smears, with an 11 times higher sensitivity.

3.2.2 COMPUTER-AIDED MICROSCOPY

A typical approach of a **computer-aided** method usually comprises four different image processing and analysis tasks, as follows:

1. Preprocessing
2. Segmentation
3. Feature extraction
4. Classification

Morphological techniques have been used in the first three phases. Each sub-section contains description about methods that cope with malaria parasites (MP) stained components analysis, both on thin and thick blood smears, without distinction.

The initial step in detecting the presence of malaria parasites in an image of a blood sample is to segment possible regions of interest (foreground) from the background. The background for the malaria images is a large region of stained blood sample, and depending on the quality of staining performed and the image captured, the color of the background varies from a bright pink to a dull gray tone. A segmentation scheme based on the HSV color space to separate the red blood cells (RBC) and parasites is used. This scheme focuses on detecting the parasites' chromatin within the RBCs once the segmentation was done. The HSV color space was used since the RGB color space is not intuitive for processing, unlike HSV which offers a representation that could be used to easily identify among color families even with varying image qualities.

An alternative method does not consider for segmentation the hue of the individual pixels of the image. Instead, a global threshold was obtained by varying the contrast among pixels, and was used to classify each pixel as belonging to either foreground or background. To do this, pre-processing of the images to obtain "uniform" images was done by converting all input images to grayscale. The images were then filtered to normalize the pixel intensities around a median value before obtaining the histogram of the individual images and expanding each until the two intensity

classes, foreground and background, become distinct given a threshold intensity.

In conjunction with the techniques explored in the previous paragraphs, machine learning techniques serve as the core for decision systems for malaria.

In order to reduce mortality rate and improve diagnosis, image analysis software and machine learning methods have been used to quantify parasitemia in microscopic blood slides.

CHAPTER 4

PROPOSED TECHNIQUE

4.1 INTRODUCTION

A database containing blood sample images of the intended four type of species in PNG format was downloaded from the Kaggle website. The RGB images were converted to grayscale images and subjected to the proposed technique, briefly explained in the following topics. The proposed technique considers many features of the image like Contrast, Correlation, Homogeneity, Entropy and Energy. As the number of features increases, accuracy of the detection process increases. Here, more than one Machine Learning classifiers are used for detection of malarial parasites. The efficiency of the corresponding Machine Learning classifiers is estimated and compared.

4.1.1 PERSPECTIVES ON IMAGE PRE-PROCESSING

Image pre-processing is a common name for operations with images at the lowest level of abstraction - both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightness's). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, and translation) are classified among pre-processing methods here since similar techniques are used.

4.1.2 VISION PIPELINES AND IMAGE PRE-PROCESSING

Local binary features deal with the pixel intensity comparisons of point-pairs. This makes the comparisons relatively insensitive to illumination, brightness, and contrast, so there may not be much need for image pre-processing to achieve good results. Current local binary pattern methods as described in the literature do not typically call for much image pre-processing; they rely on a simple comparison threshold that can be adjusted to account for illumination or contrast. Spectra descriptors, such as SIFT (which acts on local region gradients) and SURF (which uses HAAR-like features with integrated pixel values over local regions), offer diverse pre-processing opportunities. Methods that use image pyramids often perform some image pre-processing on the image pyramid to create a scale space representation of the data using Gaussian filtering to smooth the higher levels of the pyramid. Basic illumination corrections and filtering may be useful to enhance the image prior to computing gradients—for example, to enhance the contrast within a band of intensities that likely contain gradient-edge information for the features. But in general, the literature does not report good or bad results for any specific methods used to pre-process the image data prior to feature extraction, and therein resides the opportunity. Basis space features are usually global or regional, spanning a regular shaped polygon—for example, a Fourier transform computed over the entire image or block.

However, basis space features may be part of the local features, such as the Fourier spectrum of the LBP histogram, which can be computed over histogram bin values of a local descriptor to provide rotational invariance. The most complex descriptor family is the polygon shape-based descriptors, which potentially require several image pre-processing steps to isolate the polygon structure and shapes in the image for measurement.

4.1.3 POLYGON SHAPE FAMILY PRE-PROCESSING

Polygon shapes are potentially the most demanding features when considering image pre-processing steps, the range of potential pre-processing methods is quite large and the choice of methods to employ is very data-dependent. Possibly because of the challenges and intended use-cases for polygon shape measurements, they are used only in various niche applications, such as cell biology. One of the most common methods employed for image preparation prior to polygon shape measurements is to physically correct the lighting and select the subject background. For example, in automated microscopy applications, slides containing cells are prepared with florescent dye to highlight features in the cells, then the illumination angle and position are carefully adjusted under magnification to provide a uniform background under each cell feature to be measured; the resulting images are then much easier to segment.

4.1.4 MEDIAN FILTER

The Median Filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise and also having applications in signal processing.

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note

that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median, see median for more details. The formula for median filter in image processing is:

Median value = {f (i-1, j) + …f (i, j) + f (i, j+1) + …f (i+1, j+2)} ***4.1***

where, f is the intensity value of the pixel and (i, j) are co-ordinates of the corresponding pixel

Note that, in the example above, because there is no entry preceding the first value, the first value is repeated, as with the last value, to obtain enough entries to fill the window. This is one way of handling missing window entries at the boundaries of the signal, but there are other schemes that have different properties that might be preferred in particular circumstances. Avoid processing the boundaries, with or without cropping the signal or image boundary afterwards, fetching entries from other places in the signal. With images for example, entries from the far horizontal or vertical boundary might be selected, shrinking the window near the boundaries, so that every window is full.

The efficiency of this median calculation is a critical factor in determining how fast the algorithm can run. The naïve implementation described above sorts every entry in the window to find the median; however, since only the middle value in a list of numbers is required, selection algorithms can be much more efficient.

4.1.5 GRAY-LEVEL CO-OCCURRENCE MATRICES (GLCMs)

A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix. The GLCM functions characterize the texture of an image by calculating how often pairs of pixels with specific values and in a specified

spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix. (The texture filter functions, described in Texture Analysis cannot provide information about shape, that is, the spatial relationships of pixels in an image.) A co-occurrence matrix or co-occurrence distribution is a matrix that is defined over an image to be the distribution of co-occurring pixel values (grayscale values, or colors) at a given offset.

Whether considering the intensity or grayscale values of the image or various dimensions of color, the co-occurrence matrix can measure the texture of the image. Because co-occurrence matrices are typically large and sparse, various metrics of the matrix are often taken to get a more useful set of features. Features generated using this technique are usually called Haralick features, after Robert Haralick.

Texture analysis is often concerned with detecting aspects of an image that are rotationally invariant. To approximate this, the co-occurrence matrices corresponding to the same relation, but rotated at various regular angles (e.g. 0, 45, 90, and 135 degrees), are often calculated and summed.

Texture measures like the co-occurrence matrix, wavelet transforms, and model fitting have found application in medical image analysis in particular. Texture filter functions provide a statistical view of texture based on the image histogram. These functions can provide useful information about the texture of an image but cannot provide information about shape, i.e., the spatial relationships of pixels in an image. Another statistical method that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix. The toolbox provides functions to create a GLCM and derive statistical measurements from it.

4.1.6 CREATING A GRAY-LEVEL CO-OCCURRENCE MATRIX

To create a GLCM, use the graycomatrix function. The graycomatrix function creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value i occurs in a specific spatial relationship to a pixel with the value j. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (i,j) in the resultant GLCM is simply the sum of the number of times that the pixel with value i occurred in the specified spatial relationship to a pixel with value j in the input image.

Because the processing required to calculate a GLCM for the full dynamic range of an image is prohibitive, graycomatrix scales the input image. By default, graycomatrix uses scaling to reduce the number of intensity values in grayscale image from 256 to eight. The number of gray levels determines the size of the GLCM. To control the number of gray levels in the GLCM and the scaling of intensity values, using the NumLevels and the GrayLimits parameters of the graycomatrix function.

The gray-level co-occurrence matrix can reveal certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the GLCM are concentrated along the diagonal, the texture is coarse with respect to the specified offset.

The graycomatrix calculates the first three values in a GLCM. In the output GLCM, element (1,1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively. GLCM(1,2) contains the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2. Element (1,3) in the GLCM has

the value 0 because there are no instances of two horizontally adjacent pixels with the values 1 and 3. The graycomatrix continues processing the input image, scanning the image for other pixel pairs (i,j) and recording the sums in the corresponding elements of the GLCM.

4.1.7 SPECIFYING THE OFFSETS

By default, the graycomatrix function creates a single GLCM, with the spatial relationship, or offset, defined as two horizontally adjacent pixels. However, a single GLCM might not be enough to describe the textural features of the input image. For example, a single horizontal offset might not be sensitive to texture with a vertical orientation. For this reason, graycomatrix can create multiple GLCMs for a single input image.

To create multiple GLCMs, specify an array of offsets to the graycomatrix function. These offsets define pixel relationships of varying direction and distance. For example, an array of offsets that specify four directions (horizontal, vertical, and two diagonals) and four distances can be defined. In this case, the input image is represented by 16 GLCMs. When you calculate statistics from these GLCMs, you can take the average. These offsets are specified as a p-by-2 array of integers. Each row in the array is a two-element vector, [row_offset, col_offset], that specifies one offset. row_offset is the number of rows between the pixel of interest and its neighbor. col_offset is the number of columns between the pixel of interest and its neighbor. This example creates an offset that specifies four directions and 4 distances for each direction.

4.1.8 DERIVING STATISTICS FROM A GLCM

After you create the GLCMs, you can derive several statistics from them using the graycoprops function. These statistics provide information about the texture of an image. You specify the statistics you want when you call the graycoprops function.

4.2 BLOCK DIAGRAM

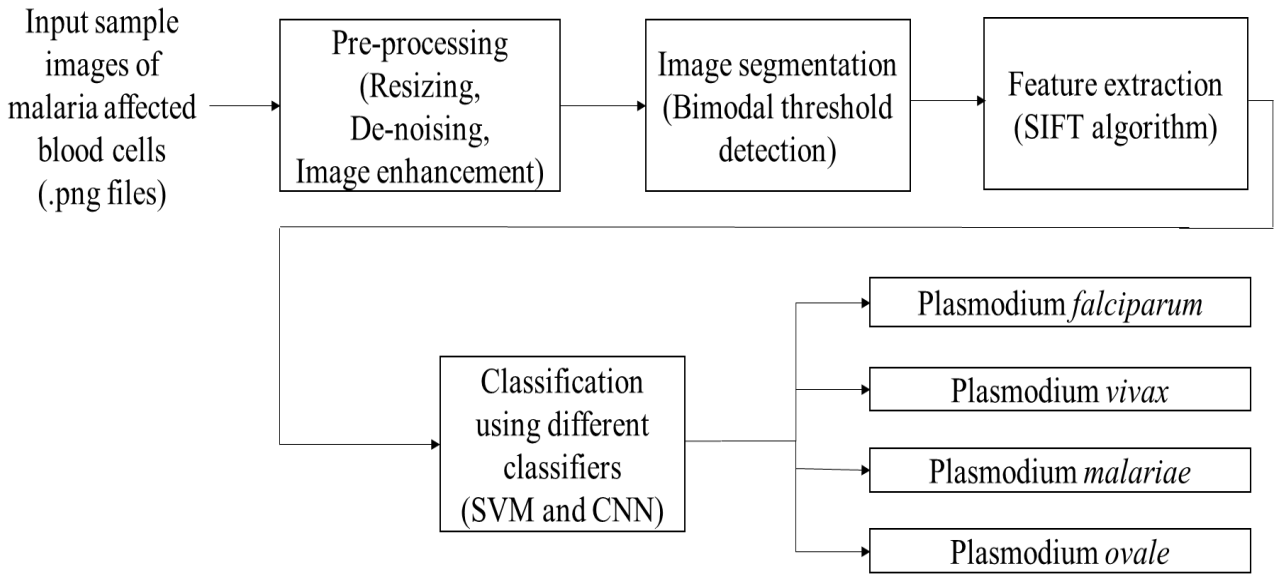


Fig. 4.1 Block diagram

Image acquisition: A database of stained blood smeared images was procured in PNG format from the following links:

https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria

https://lhncbc.nlm.nih.gov/publication/pub9932

The obtained images have different characteristics in magnifications and sizes.

Pre-processing: In the initial stage, pre-processing of the images to obtain "uniform" images was done by converting all input images to grayscale. The images were then filtered to normalize the pixel intensities around a median value before obtaining the histogram of the individual images and expanding each until the two intensity classes, foreground and background, become distinct given a threshold intensity. The aim of pre-processing is to remove unwanted objects and noise from the image to facilitate image segmentation into meaningful regions. MATLAB functions are used for implementation of the following steps: converting the colored

images into grayscale presentation, background estimation using the morphological opening technique, subtracting the background image from the original image, image contrast enhancement and conversion of grayscale images into binary presentation by thresholding techniques, converting images into its negative presentation.

- Resizing: Images were resized from dimensions of range (80-300) X (80-300) to 256 X 256.
- Elimination of noise: Median filter was used
- Enhancement of image contrast: In-built filtering techniques (morphological operations) were applied

Segmentation: Segmentation is the process of partitioning the image into different segments. Here, the image is partitioned into foreground and background. This is to isolate the RBCs. Simplest approach for segmentation is used here which is the selection of a suitable intensity threshold. All pixels with a value higher than a particular threshold value is classified as the region of interest and all pixels with a lower value are classified as background pixels.

Such a distribution is called bimodal because there are two mode values: one for the background and one for the feature. Once the foreground has been extracted in some cases there are still clumped RBCs and as such an iterative clump splitting process is performed. The individual cells are stored while the extracted clumps are split. For each clump the boundary is extracted and its curvature is calculated. The boundary curvature vector for each object is analyzed for regional maxima, indicating points of concavity from which potential split lines will be drawn. An iterative process for RBC clump splitting is applied. The original individual RBCs and resultant individual RBCs from the clump splitting process are combined into a RBC binary image. Cells that were unnecessarily split are morphologically

reconstructed and added back to the RBC binary image.

- Bi-modal threshold detection was used to differentiate useful information of blood cells from the background.
- Threshold value used- 0.45(based on trial and error method)

Feature extraction*:* Feature extraction is applied to extract relevant features from images. Here, SIFT is used for extracting features. The scale-invariant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of three or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

The desired features were extracted from the images using Scale-Invariant Feature Transform (SIFT) based on Euclidean distance. The common terms used are *P- intensity value of the pixel, (i,j)- co-ordinates of the pixel* and *G- total number of pixels in the image.*

• ***Contrast (CTR)****:* Obtained by the change in brightness of image from one pixel to another, pointed by the data cursor (in-built feature).

$$CTR = \sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=1}^{G} \sum_{j=1}^{G} P(i,j) \right\}, |i-j| = n \qquad \textbf{4.2}$$

where, *n- order*

• ***Correlation (COR):*** Correlation is a measure of gray level linear dependence between the pixels at the specified positions relative to each other.

$$COR = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j} \qquad \textbf{4.3}$$

$$\mu_j = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} jP(i,j) \qquad \textbf{4.4}$$

$$\mu_i = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} iP(i,j) \qquad \textbf{4.5}$$

$$\sigma_j = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} (j-\mu_j)^2 P(i,j) \qquad \textbf{4.6}$$

$$\sigma_i = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} (j-\mu_i)^2 P(i,j) \qquad \textbf{4.7}$$

where, μ- *mean*

σ- *variance*

- ***Homogeneity (IDM):*** It is the uniformity in composition. Also known as Inverse Difference Moment (IDM)

$$IDM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{1}{1+(i-j)^2}P(i,j) \qquad \textbf{4.8}$$

- ***Energy (E):*** Determines the intensity of pixels. Calculated as follows:

$$E = \sum_{i=0}^{G-1}[P(i)]^2 \qquad \textbf{4.9}$$

- ***Entropy (H):*** Statistical measure of randomness used to characterize texture of image. Calculated as follows:

$$H = -\sum_{i=0}^{G-1}P(i)\log_2 P(i) \qquad \textbf{4.10}$$

Classification: The processed image was classified using SVM and CNN classifiers. This is the final stage where the features of input image are compared with that of the infected RBC image features and then classified. For this, the classifier is trained by providing the features extracted in the previous stage. In the proposed methodology, classification is implemented by using SVM and CNN classifiers and, then, their respective performances are compared and analyzed

Identification: The type of malaria causing species was identified.

4.3 SUPPORT VECTOR MACHINE (SVM)

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible (see Fig. 4.2). New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

The parameters of the maximum-margin hyper-plane are derived by solving the optimization. There exist several specialized algorithms for quickly solving the quadratic programming (QP) problem that arises from SVMs, mostly relying on heuristics for breaking the problem down into smaller, more manageable chunks.

Another approach is to use an interior-point method that uses Newton-like iterations to find a solution. Another common method is sequential minimal optimization (SMO) algorithm, which breaks the problem down into 2-dimensional sub-problems that are solved analytically, eliminating the need for a numerical optimization

algorithm and matrix storage. This algorithm is conceptually simple, easy to implement, generally faster, and has better scaling properties for difficult SVM problems.

The special case of linear support-vector machines can be solved more efficiently by the same kind of algorithms used to optimize its close cousin, logistic regression; this class of algorithms includes sub-gradient descent (e.g., PEGASOS) and coordinate descent (e.g., LIBLINEAR).

LIBLINEAR has some attractive training-time properties. Each convergence iteration takes time linear in the time taken to read the train data, and the iterations also have a Q-linear convergence property, making the algorithm extremely fast.

The support vector clustering algorithm created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

For greater accuracy and kernel-function choices on low- through medium-dimensional data sets, train a binary SVM model or a multiclass error-correcting output codes (ECOC) model containing SVM binary learners using the Classification Learner app. For greater flexibility, use the command-line interface to train a binary SVM model using train a multiclass ECOC model composed of binary SVM learners.

For reduced computation time on high-dimensional data sets that fit in the MATLAB Workspace, efficiently train a binary, linear classification model, such as a linear SVM model, using train a multiclass ECOC model composed of SVM models. For nonlinear classification with big data, train a binary, Gaussian kernel classification model.

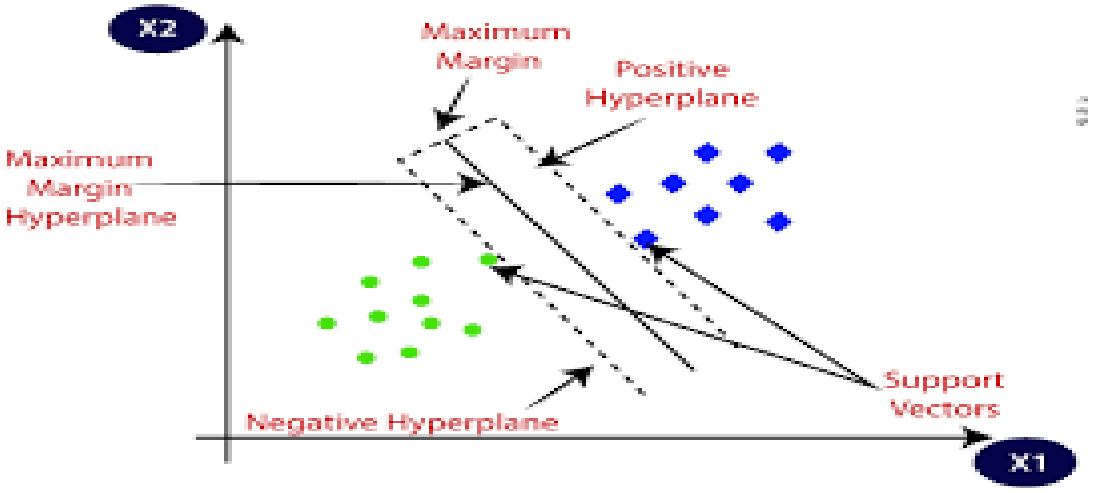


Fig. 4.2 Hyperplane support vector machine algorithm

Source: https://www.javatpoint.com/machine-learning-support vector-machine-algorithm

4.4 CONVOLUTIONAL NEURAL NETWORK (CNN)

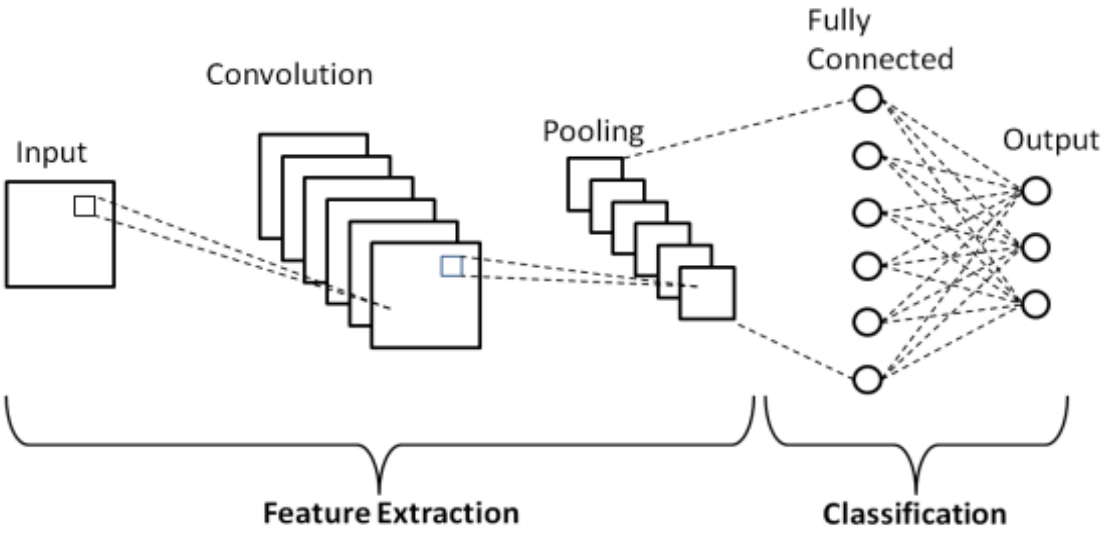


Fig. 4.3 Neural network layer architecture

Source: https://missinglink.ai/guides/convolutional-neural networks/convolutional-neural-network-architecture-forging-pathways-future/

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied

to analyze visual imagery. CNN uses a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNN uses relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design are major advantages. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers (as shown in Fig. 4.3)

Description of the process as a convolution in neural networks is by convention. Mathematically, it is a cross-correlation rather than a convolution. This only has significance for the indices in the matrix, and thus which weights are placed at which index.

4.4.1 CONVOLUTIONAL

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. Each convolutional neuron processes data only for its receptive field.

Although fully connected feed forward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very

high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable.

For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using back propagation.

4.4.2 POOLING

Convolutional networks may include local or global pooling layers which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Another example is average pooling, which uses the average value from each of a cluster of neurons at the prior layer.

4.4.3 FULLY CONNECTED

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP).

4.4.4 WEIGHTS

CNNs share weights in convolutional layers, which means that the same filter (weights bank) is used for each receptive field in the layer; this reduces memory footprint and improves performance.

4.4.5 APPLICATIONS

Image recognition

CNNs are often used in image recognition systems. When applied to facial recognition, CNNs achieved a large decrease in error rate. CNNs were used to assess video quality in an objective way after manual training; the resulting system had a very low root mean square error.

Video analysis

Some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream.

Natural language processing

CNNs have also been explored for natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

Drug discovery

CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments.

Health risk assessment and biomarkers of aging discovery

CNNs can be naturally tailored to analyze a sufficiently large collection of time series data representing one-week-long human physical activity streams augmented by the rich clinical data.

Checkers game

CNNs have been used in the game of checkers. The learning process did not use prior human professional games, but rather focused on a minimal set of information contained in the checkerboard: the location and type of pieces, and the piece differential.

Go

CNNs have been used in computer Go. In December 2014, Clark and Storkey published a paper showing that a CNN trained by supervised learning from a database of human professional games could outperform GNU Go and win some games against Monte Carlo tree search Fuego.

CHAPTER 5

HARDWARE AND SOFTWARE REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- Processor- Pentium dual core 2.00 GHz
- Disk space- 40 GB
- RAM- 4GB (Minimum)

5.2 SOFTWARE REQUIREMENTS

- C language
- MATLAB R2018a

5.2.1 INTRODUCTION

MATLAB is a highly proficient language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notations.

5.2.2 COMPONENTS OF THE MATLAB SYSTEM

Development Environment - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

Manipulating Matrices - introduces how to use MATLAB to generate Matrices and perform mathematical operations on matrices.

Graphics - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs and working with images.

Programming with MATLAB - describes how to use the MATLAB language to create scripts, functions and manipulate data structures such as cell arrays and multi-dimensional arrays.

Typical uses of MATLAB include:

i. Math and computation

ii. Algorithm development

iii. Modeling, simulation, and prototyping

iv. Data analysis, exploration, and visualization

v. Scientific and engineering graphics

vi. Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the tool of choice for high productivity research, development and analysis.

Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation and etc.

5.2.3 MATLAB SYSTEM

The MATLAB system consists of five main parts:

Development Environment

This is a set of tools and facilities that help you to use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history and browsers for viewing help, the workspace, files and the search path.

The MATLAB Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine and complex arithmetic to more sophisticated functions like matrix inverse, matrix eigen values, bessel functions and fast fourier transforms.

The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs and "programming in the large" to create complete large and complex application programs.

Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

5.2.4 THE MATLAB APPLICATION PROGRAM INTERFACE (API)

This is a library that allows you to write C and FORTRAN programs that interacts with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine and for reading and writing MAT-files.

Development environment

This chapter provides a brief introduction to starting and quitting MATLAB and the tools and functions that help you to work with MATLAB variables and files.

Starting and Quitting MATLAB

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop.

On a UNIX platform, to start MATLAB, type matlab at the operating system prompt.

After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish .m script.

MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables and applications associated with MATLAB.

Image Processing with MATLAB

The MATLAB help tool is especially useful in image processing applications, since there are information about various keywords, prompts and numerous filter examples.

1. Opening MATLAB in the microcomputer lab

Access the Start Menu, Proceed to Programs, Select MATLAB 14 from the MATLAB 14 folder --OR-- Open through C:\MATLAB6p5\bin\win32\matlab.exe

2. MATLAB

The Command Window is the window on the right-hand side of the screen. This window is used to enter both the commands for MATLAB to execute and to view the results of these commands. The Command History window, in the lower left side of the screen, displays the commands that have been recently entered into the Command Window. When MATLAB opens, the screen should look something like what is pictured in Figure 5.1, below.

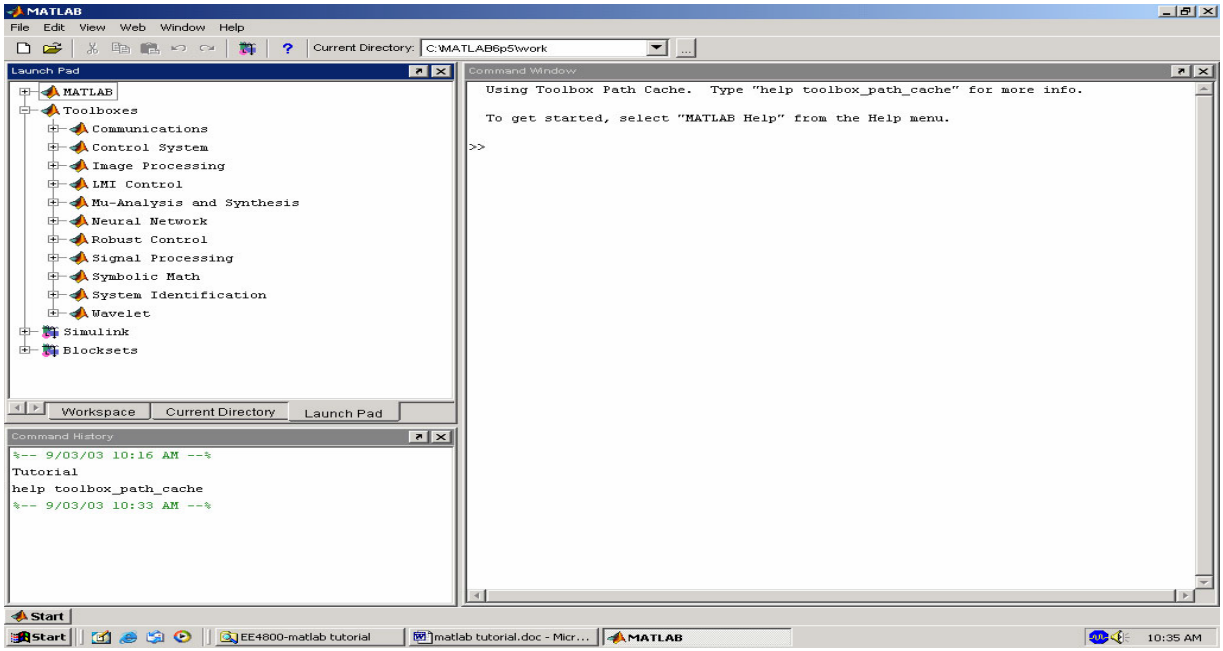


Fig. 5.1 MATLAB window

In the upper left-hand side of the screen, there is a window that can contain three different windows with tabs to select between them. The first window is the Current Directory, which tells the user which M-files are currently in use. The second window is the Workspace window, which displays which variables are currently being used and how big they are. The third window is the Launch Pad window, which is especially important since it contains easy access to the available toolboxes, of which Image Processing is one.

If these three windows do not all appear as tabs below the window space, simply go to View and select the ones you want to appear. In order to gain some familiarity with the Command Window, try Example 5.1, below. You must type code after the >> prompt and press return to receive a new prompt. If you write code that you do not want to reappear in the MATLAB Command Window, you must place a semi colon after the line of code. If there is no semi colon, then the code will print in the command window just under where you typed it.

Example 5.1

X = 1; %press enter to go to next line

Y = 1; %press enter to go to next line

Z = X + Y %press enter to receive result

As you probably noticed, MATLAB gave an answer of Z = 2 under the last line of typed code. If there had been a semi colon after the last statement, the answer would not have been printed. Also, notice how the variables you used are listed in the Workspace Window and the commands you entered are listed in the Command History window. If you want to retype a command, an easy way to do this is to press the ↑ or ↓ arrows until you reach the command you want to re-enter.

3. The M-file

M-file – An M-file is a MATLAB document the user creates to store the code they write for their specific application. Creating an M-file is highly recommended although, not entirely necessary. An M-file is useful because it saves the code the users have written for their application. It can be manipulated and tested until it meets the user's specifications. The advantage of using an M-file is that the user, after modifying their code, must only tell MATLAB to run the M-file, rather than reenter each line of code individually.

Example 5.2. Creating an M-file – To create an M-file, select File\New ►M-file. Saving – The next step is to save the newly created M-file. In the M-file window, select File\Save As… Choose a location that suits your needs, such as a disk, the hard drive or the U drive. It is not recommended that you work from your disk or from the U drive, so before editing and testing your M-file you may want to move your file to the hard drive. Opening an M-file – To open up a previously designed M-file, simply open MATLAB in the same manner as described before. Then, open the M-file by going to File\Open…, and selecting your file. Then, in order for MATLAB to recognize where your M-file is stored, you must go to File\Set Path… This will open up a window that will enable you to tell MATLAB where your M-file is stored. Click the Add Folder… button, then, browse to find the folder that your M-file is located in and press OK. Then in the Set Path window, select Save, and then Close. If you do not set the path, MATLAB may open a window saying your file is not in the current directory. In order to get by this, select the "Add directory to the top of the MATLAB path" button, and hit OK. This is essentially the same as setting the path, as described above.

Writing Code – After creating and saving your M-file, the next step is to begin writing code. A suggested first move is to begin by writing comments at the top of

the M-file with a description of what the code is for, who designed it, when it was created and when it was last modified. Comments are declared by placing a % symbol before them. Comments appear in green in the M-file window. See Figure 5.2, below, for Example 5.1.

Resaving – After writing code, you must save your work before you can run it. Save your code by going to File\Save.

Running Code – To run code, simply go to the main MATLAB window and type the name of your M-file after the >> prompt. Other ways to run the M-file are to press F5 while the M-file window is open, select Debug\Run, or press the Run button (see Figure 5.1) in the M-file window toolbar.

Example 5.1

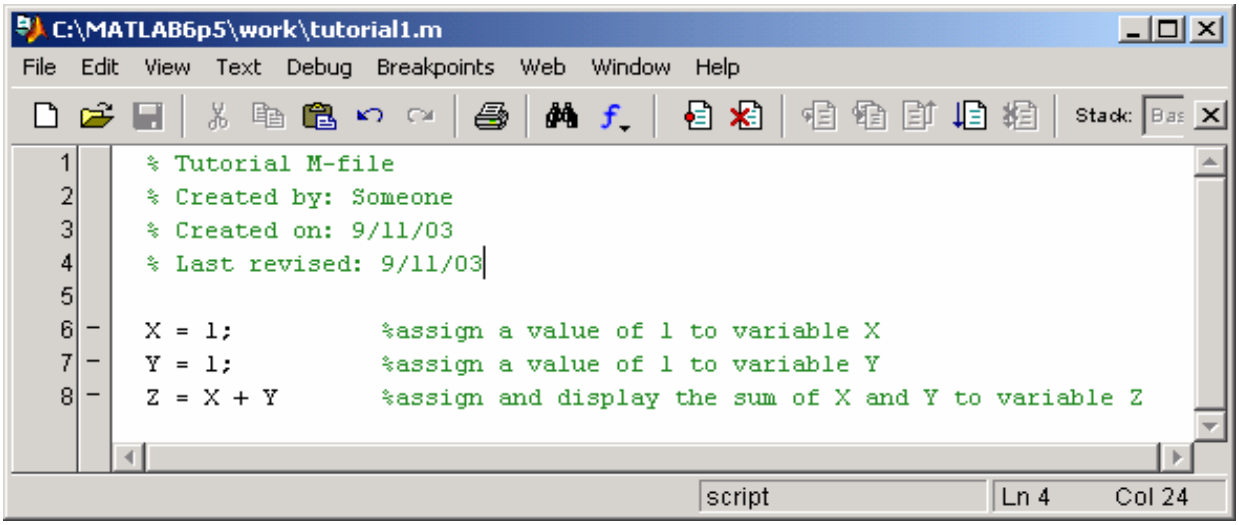


Fig. 5.2 Example of M-file

Images

Images – The first step in MATLAB image processing is to understand that a digital image is composed of a two- or three-dimensional matrix of pixels. Individual pixels contain a number or numbers representing what grayscale or color value is assigned to it. Color pictures generally contain three times as much data as grayscale pictures, depending on what color representation scheme is used. Therefore, color pictures

take three times as much computational power to process. In this tutorial, the method for conversion from color to grayscale will be demonstrated and all processing will be done on grayscale images. However, in order to understand how image processing works, we will begin by analyzing simple two dimensional 8-bit matrices.

Loading an Image – Many times you will want to process a specific image, other times you may just want to test a filter on an arbitrary matrix. If you choose to do this in MATLAB you will need to load the image so you can begin processing. If the image that you have is in color, but color is not important for the current application, then you can change the image to grayscale. This makes processing much simpler since then there are only a third of the pixel values present in the new image. Color may not be important in an image when you are trying to locate a specific object that has good contrast with its surroundings. Example 5.2, below, demonstrates how to load different images.

Example 5.2

In some instances, the image in question is a matrix of pixel values. For example, you may need something to test a filter on, but you do not yet need a real image to test the filter. Therefore, you can simply create a matrix that has the characteristics wanted such as areas of high and low frequency. Other times a stored image must be imported into MATLAB to be processed. If color is not an important aspect then rgb2gray can be used to change a color image into a grayscale image.

C:\MATLAB6p5\toolbox\images\imdemos. Therefore, it is a good idea to know how to load any image from any folder.

The class of the new image is the same as that of the color image. As you can see from the example M-file in Figure 5.3, MATLAB has the capability of loading many different image formats, two of which are shown.

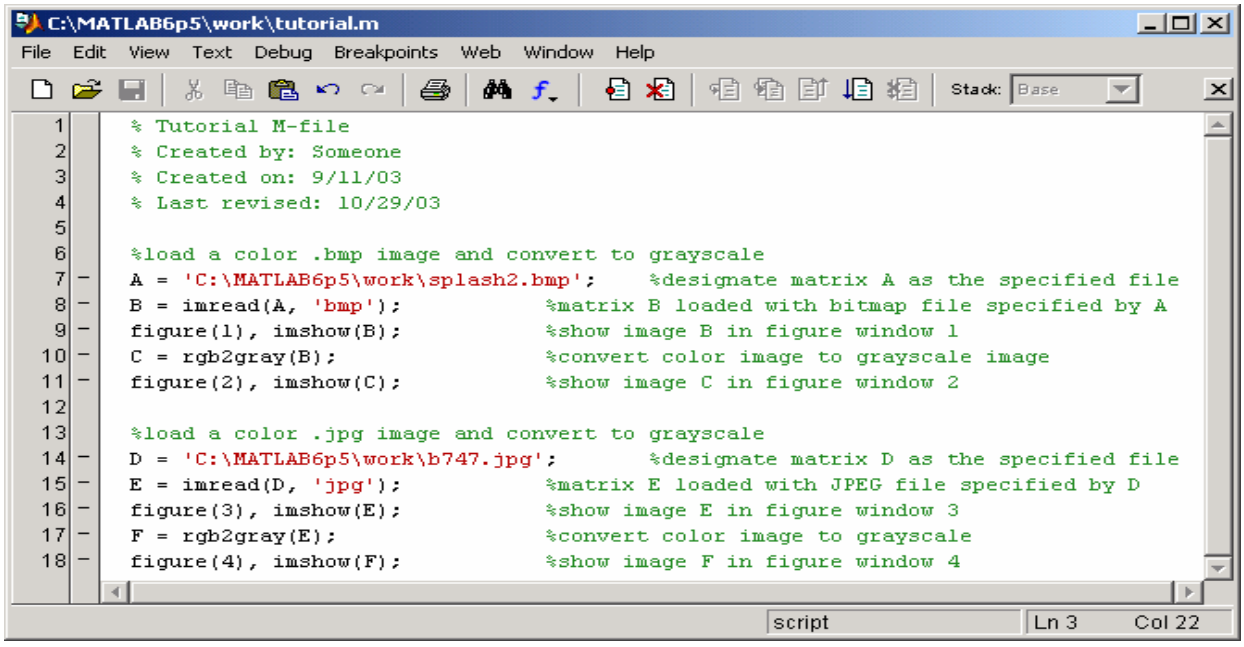


Fig. 5.3 M-file for Loading Images

The function imread is used to read an image file with a specified format. Consult imread in MATLAB's help to find which formats are supported. The function imshow displays an image, while figure tells MATLAB which figure window the image should appear in. If figure does not have a number associated with it, then figures will appear chronologically as they appear in the M-file.

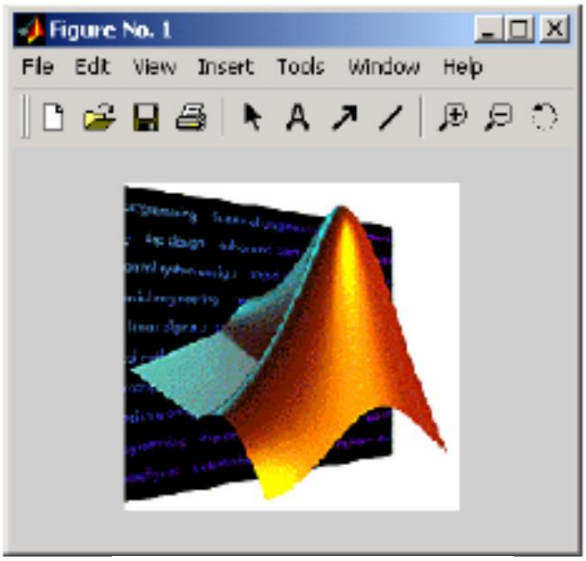


Fig. 5.4 Bitmap image

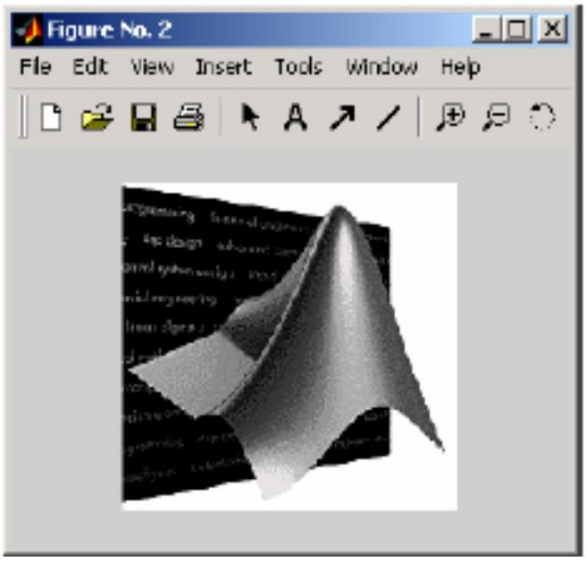


Fig. 5.5 Grayscale image (Example 5.2)

Figures 5.4, 5.5, 5.6 and 5.7 are a loaded bitmap file, the image in Figure 5.4 converted to a grayscale image, a loaded JPEG file, and the image in Figure 5.5 converted to a grayscale image, respectively. The images used in this example are both MATLAB example images. In order to demonstrate how to load an image file, these images were copied and pasted into the folder denoted in the M-file in Figure 5.2. In Example 5.3, later in this tutorial, you will see that MATLAB images can be loaded by simply using the imread function.

Writing an Image – Sometimes an image must be saved so that it can be transferred to a disk or opened with another program. In this case you will want to do the opposite of loading an image, reading it and instead write it to a file. This can be accomplished in MATLAB using the imwrite function. This function allows you to save an image as any type of file supported by MATLAB, which are the same as supported by imread. Example 5.3, below, contains code necessary for writing an image.

Example 5.3

In order to save an image, you must use the imwrite function in MATLAB. The M-file in Figure 5.8 contains code for saving an image. This M-file loads the same bitmap file as described in the M-file pictured in Figure 5.2. However, this new

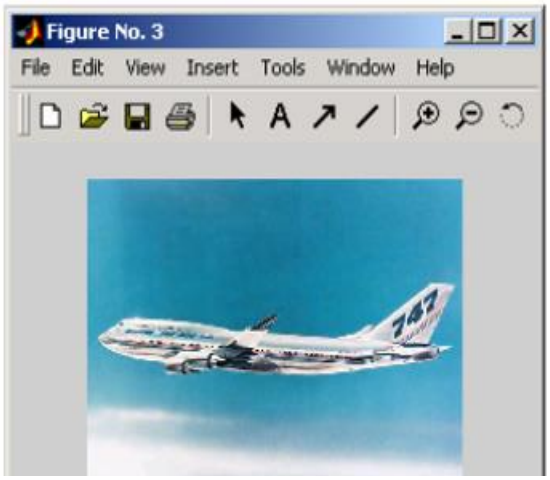


Fig. 5.6 JPEG image

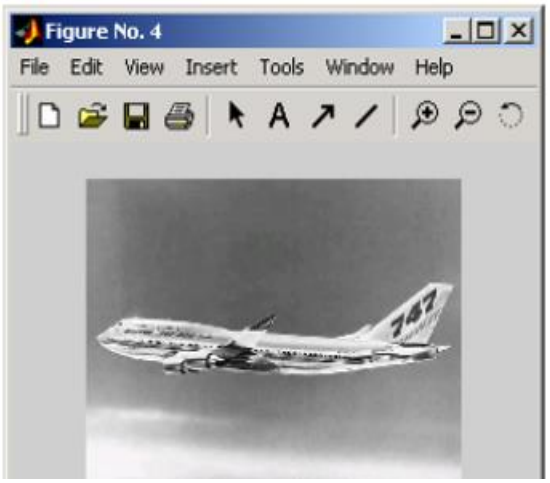


Fig. 5.7 Grayscale image (Example 5.3)

M-file saves the grayscale image created as a JPEG image. Just like in Example 5.1, the “splash2” bitmap picture must be moved into MATLAB’s work folder in order for the imread function to find it. When you run this M-file notice how the JPEG image that was created is saved into the work folder.

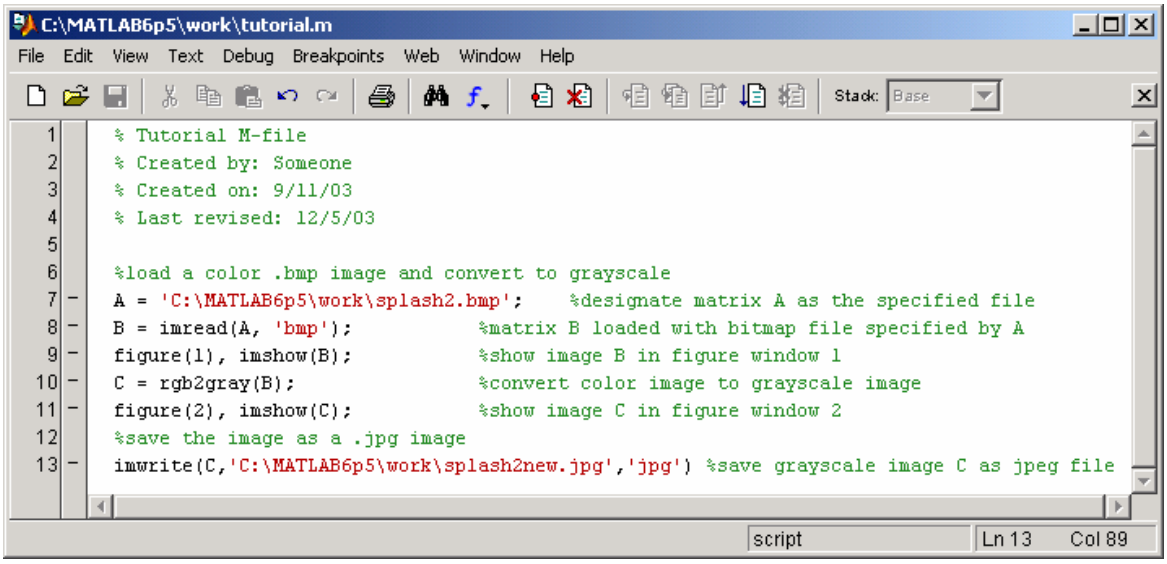


Fig. 5.8 M-file for Saving an Image

Image Properties

Histogram – A histogram is bar graph that shows a distribution of data. In image processing histograms are used to show how many of each pixel value are present in an image. Histograms can be very useful in determining which pixel values are important in an image. From this data you can manipulate an image to meet your specifications. Data from a histogram can aid you in contrast enhancement and thresholding. In order to create a histogram from an image, use the imhist function. Contrast enhancement can be performed by the histeq function, while thresholding can be performed by using the graythresh function and the im2bw function. See Example 5.3, for a demonstration of imhist, imadjust, graythresh and im2bw. If you want to see the resulting histogram of a contrast enhanced image, simply perform the imhist operation on the image created with histeq.

Negative – The negative of an image means the output image is the reversal of the input image. In the case of an 8-bit image, the pixels with a value of 0 take on a new value of 255, while the pixels with a value of 255 take on a new value of 0. All the pixel values in between take on similarly reversed new values. The new image appears as the opposite of the original. The imadjust function performs this operation. See Example 5.3, for an example of how to use imadjust to create the negative of the image. Another method for creating the negative of an image is to use imcomplement, which is described.

Frequency Domain

Fourier Transform – In order to understand how different image processing filters work, it is a good idea to begin by understanding what frequency has to do with images. An image is in essence a two-dimensional collection of discrete signals. Therefore, the signals have frequencies associated with them. For instance, if there is a relatively little change in grayscale values as you scan across an image, then there is lower frequency content contained within the image. If there is wide variation in grayscale values across an image then there will be more frequency content associated with the image. From signal processing, we know that any signal can be represented by a collection of sine waves of differing frequencies, magnitudes and phases. This transformation of a signal into its constituent sinusoids is known as the Fourier Transform. This collection of sine waves can potentially be infinite, if the signal is difficult to represent, but is generally truncated at a point where adding more signals does not significantly improve the resolution of the recreation of the original signal. In digital systems, we use a Fourier Transform designed in such a way that we can enter discrete input values, specify our sampling rate and have the computer generate discrete outputs. This is known as the Discrete Fourier Transform or DFT. MATLAB uses a fast algorithm for performing a DFT, which is called the Fast Fourier Transform or FFT, whose MATLAB command is fft. The FFT can be

performed in two dimensions, fft2 in MATLAB. This is very useful in image processing because we can then determine the frequency plotted just one row, so that it showed the grayscale value stored within each pixel, you might end up with something that looks like a bar graph, with varying values in each pixel location. Each pixel value in this signal may appear to have no correlation to the next one. However, the Fourier Transform can determine which frequencies are present in the signal. In order to see the frequency content, it is useful to view the absolute value of the magnitude of the Fourier Transform, since the output of a Fourier Transform is complex in nature.

Filters

Filters – Image processing is based on filtering the content of images. This could entail blurring, deblurring, locating certain features within an image, etc. Linear filtering is accomplished using convolution, as discussed above. A filter or convolution kernel as it is also known is basically an algorithm for modifying a pixel value, given the original value of the pixel and the values of the pixels surrounding it. There are literally hundreds of types of filters that are used in image processing. However, we will concentrate on several common ones.

Low Pass Filters – The first filters we will talk about are low pass filters. These filters blur high frequency areas of images. This can sometimes be useful when attempting to remove unwanted noise from an image.

Median Filters – A median filter is like an averaging filter in some ways. The averaging filter examines the pixel in question and its neighbor’s pixel values and returns the mean of these pixel values.

Since, low pass filters do not discriminate between noise and edges and tend to smooth out content that should not be smoothed out, median filters are used for research related work.

Erosion and Dilation

Erosion and Dilation are similar operations to median filtering in that they both are neighborhood operations. The erosion operation examines the value of a pixel and its neighbors and sets the output value equal to the minimum of the input pixel values. Dilation, on the other hand, examines the same pixels and outputs the maximum of these pixels. In MATLAB erosion and dilation can be accomplished by the imerode and imdilate functions, respectively, accompanied by the strel function. Edge detectors are very useful for locating objects within images.

Segmentation

Segmentation is the process of fractioning an image into its component objects. This can be accomplished in various ways in MATLAB. One way is to use a combination of morphological operations to segment touching objects within an image. Another method is to use a combination of dilation and erosion to segment objects. The MATLAB function bwperim performs this operation on binary images.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 INTRODUCTION

A dataset containing blood sample images was acquired from the kaggle website in PNG format. Total images used for training and testing:

- SVM algorithm- 100 .png images per species type
- CNN algorithm- 1000 .png images

Out of the total images in the dataset, 75% of them were used for training and 25% of them were used for testing.

6.2 PROCEDURE

Images in the dataset were later converted to JPEG format for pre-processing. Images were resized from dimensions of range (80-300) X (80-300) to 256 X 256 bits. RGB values were converted into gray values, corresponding to the pixel value of the input sample image.

Pre-defined and in-built filtering techniques are applied for the better contrast on an image and also other techniques for multiple dimensions of images are applied. De-noising was done using median filter. Bi-modal threshold detection was used to differentiate useful information of blood cells from the background.

Image segmentation process takes place with the help of boundary descriptors. The cells that have been identified as possibly infected are then extracted from the image and passed to the next stage of the algorithm for feature extraction. The desired

features were extracted from the images using Scale-Invariant Feature Transform (SIFT) based on Euclidean distance.

Features such as Contrast, Correlation, Homogeneity, Energy and Entropy were extracted. Classification process takes place. The type of malarial parasite present in the blood sample was identified depending upon the intensity value of pixels.

6.3 SAMPLE CODES OF SVM AND CNN ALGORITHMS

The sample codes for SVM and CNN algorithms are attached below.

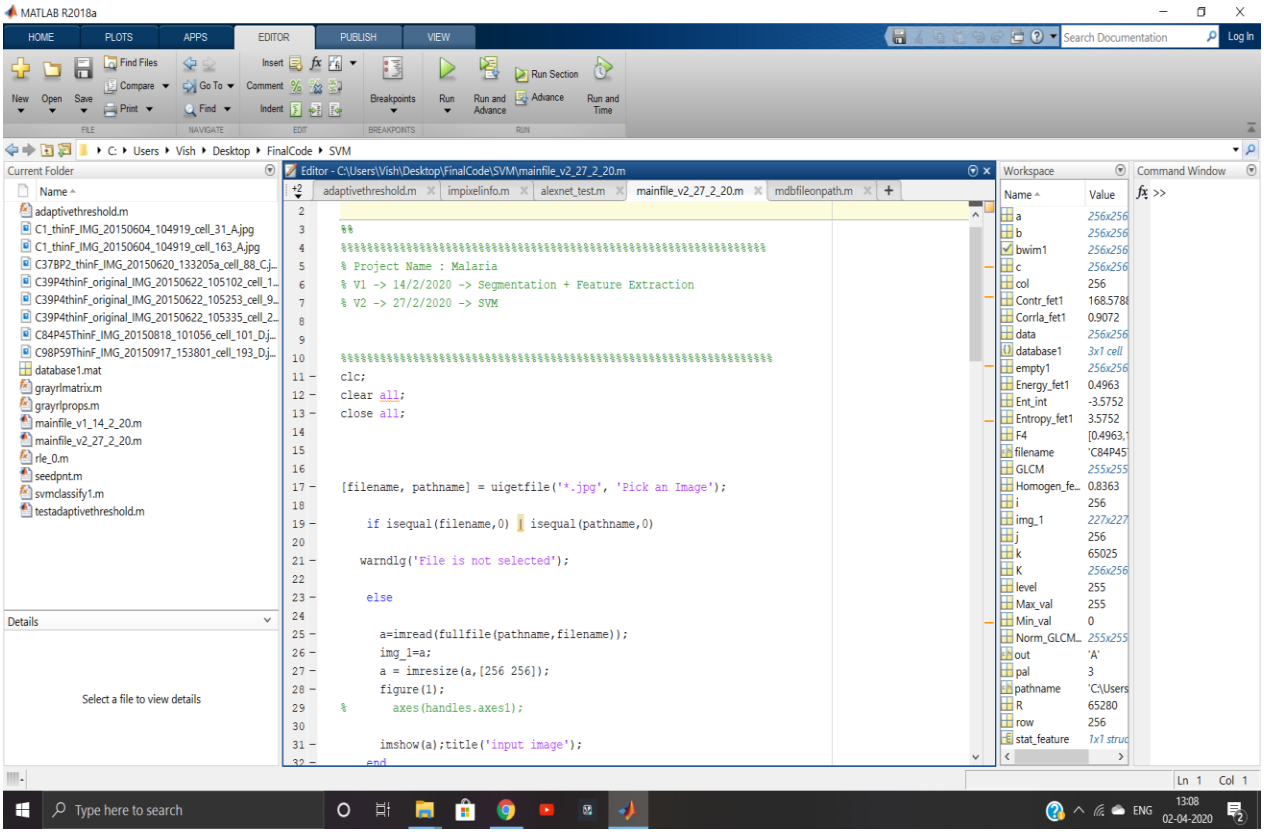


Fig. 6.1 SVM sample code

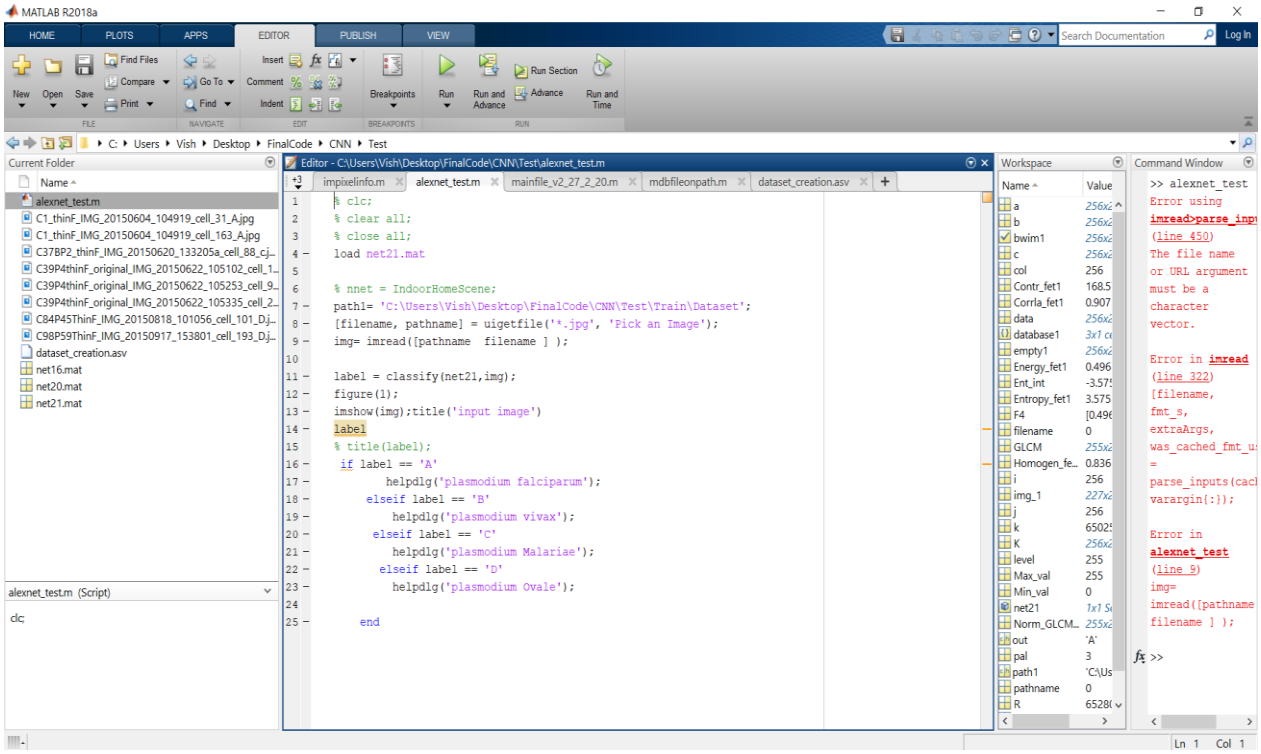


Fig. 6.2 CNN sample code

6.4 OUTPUT OF SVM CLASSIFIER

(1) SPECIES TYPE 1: PLASMODIUM FALCIPARUM

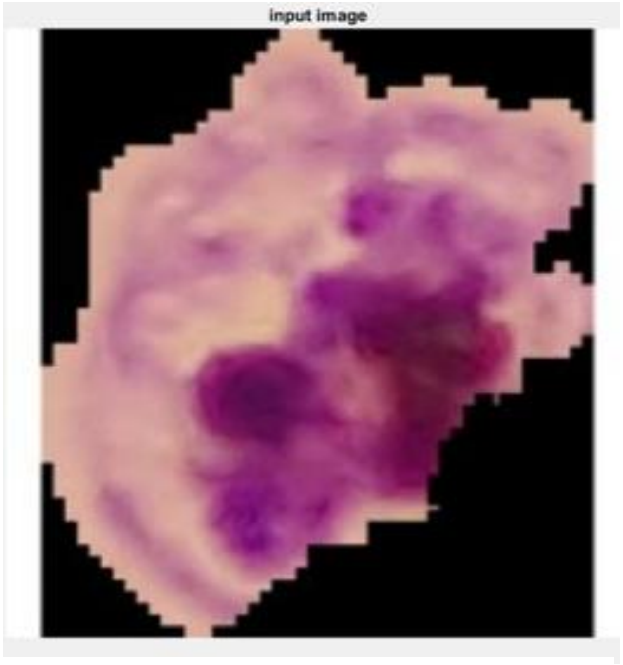


Fig. 6.3 Input image of blood sample of species type-1 (SVM)

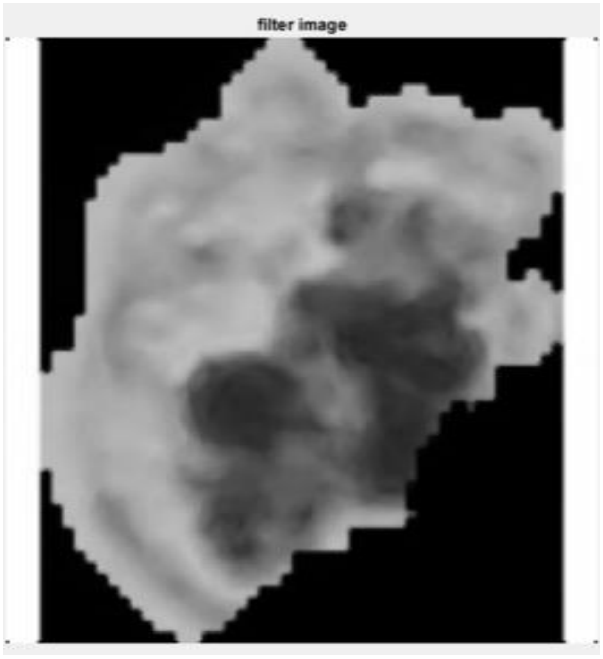


Fig. 6.4 Filtered input image of species type-1 using median filter (SVM)

The blood sample image is fed as input to the SVM algorithm. JPEG format conversion and Gray-scale conversion takes place in order to support pre-processing techniques (see Fig. 6.3). Resizing of images, de-noising of images with the courtesy of median filter, contrast enhancement and morphological operations (dilation and erosion) are applied to the subjected image (see Fig. 6.4).

Fig. 6.5 Binary converted image of the corresponding filtered image of species type-1 (SVM)

Fig. 6.6 Segmented image of species type-1 (SVM)

Binary conversion takes place through bi-modal threshold detection (see Fig. 6.5). Iterative clump splitting process is performed to identify the cells of interest. Image segmentation process takes place.

Feature extraction occurs with the help of SIFT algorithm. Outlier detection removes futile details in the background (see Fig. 6.6).

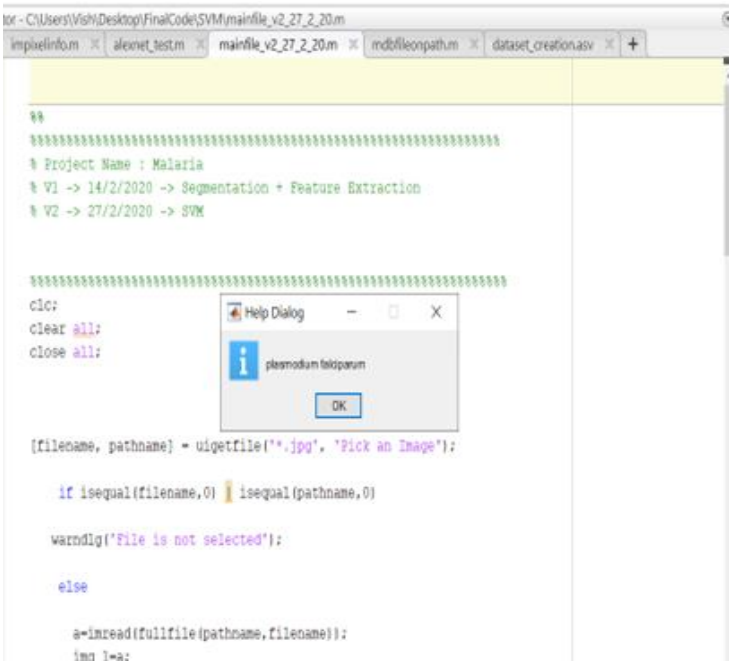


Fig. 6.7 Species type-1 malarial parasite identified (SVM)

Final stage where classification process takes place. Identification process is ensued by a dialog box indicating the type of malarial species.

(2) SPECIES TYPE 2: PLASMODIUM VIVAX

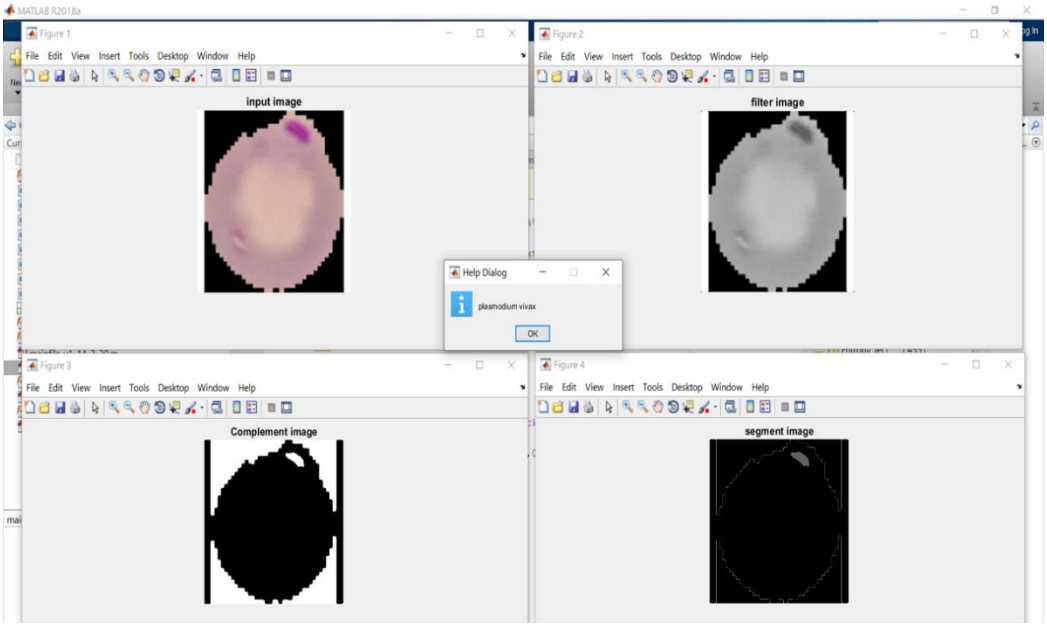


Fig. 6.8 Consolidated output of species type- 2 (SVM)

(3) SPECIES TYPE 3: PLASMODIUM MALARIAE

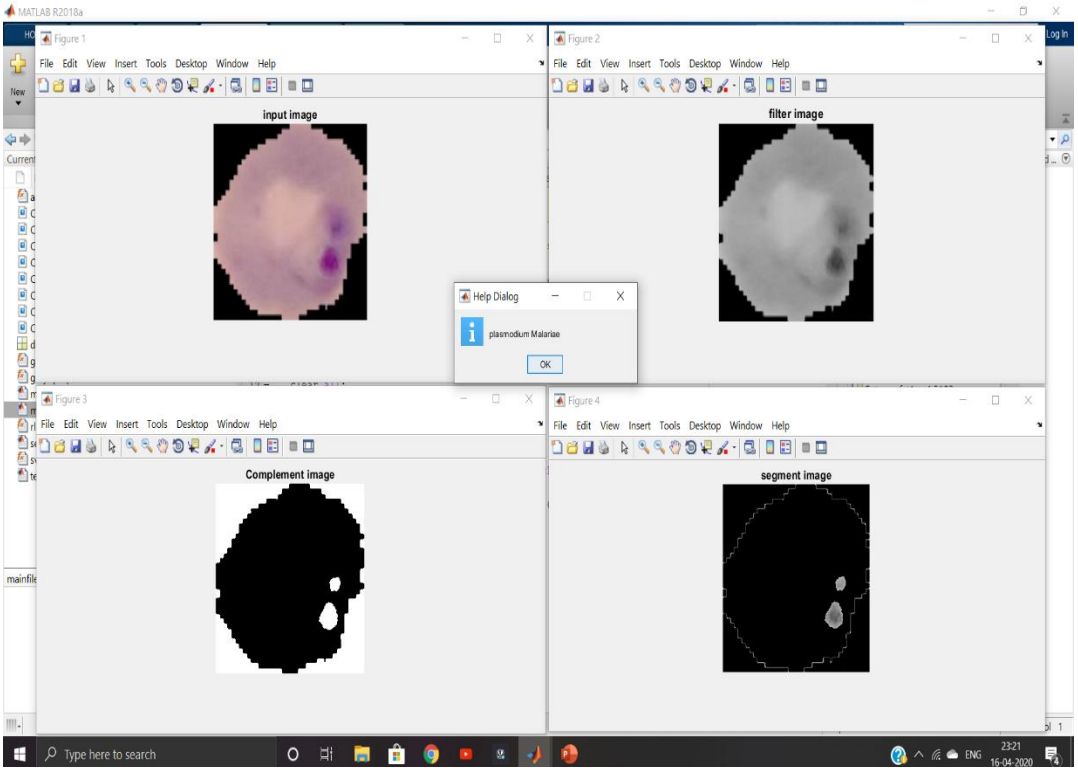


Fig. 6.9 Consolidated output of species type- 3 (SVM)

(4) SPECIES TYPE 4: PLASMODIUM OVALE

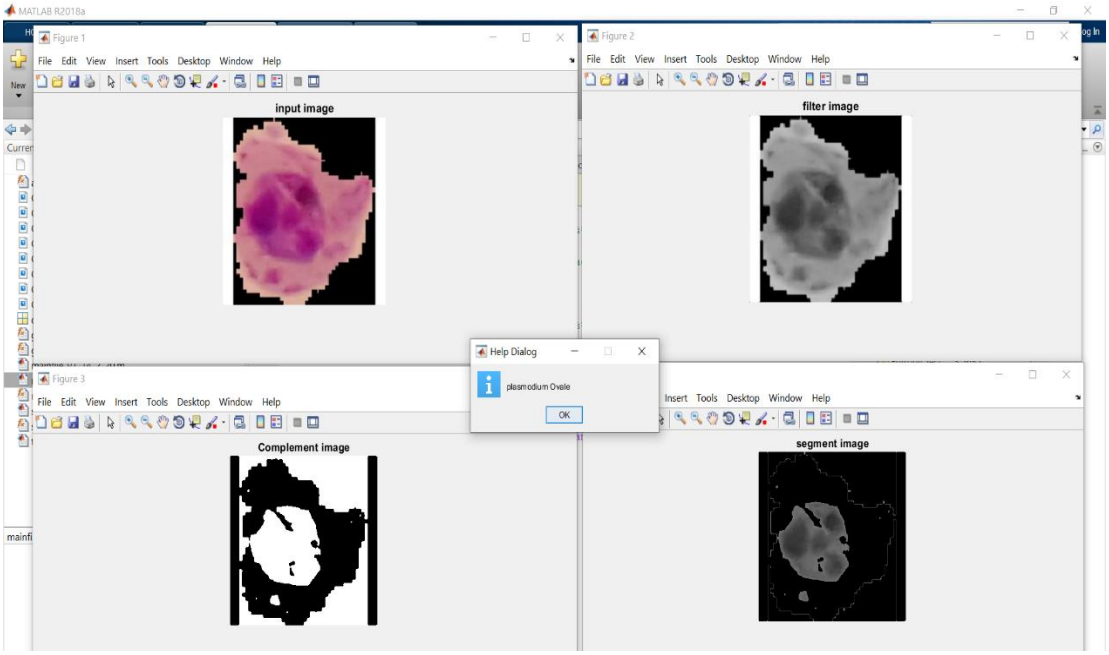


Fig. 6.10 Consolidated output of species type- 4 (SVM)

In SVM algorithm, every blood sample image from the acquired dataset undergoes the same procedural flow similar to the steps explained in the previous segment for species type-1: Plasmodium *falciparum*. In Fig. 6.8, 6.9 and 6.10, the consolidated output images of the other three types of species are displayed.

6.5 ANALYSIS OF STATISTICAL FEATURES

Table 6.1 Statistical feature analysis of the four type of species

Species type / **Statistical features**	**Plasmodium *falciparum***	**Plasmodium *vivax***	**Plasmodium *malariae***	**Plasmodium *ovale***
Contrast	204.8366	234.2128	146.0947	210.5449
Correlation	0.5011	0.5030	0.5649	0.9107
Energy	0.6830	0.7486	0.7872	0.4943
Homogeneity	0.8968	0.9200	0.9316	0.8372
Entropy	1.8376	1.4551	1.3216	3.5631

In the table 6.1, statistical feature analysis for the four type of species along with the desired features were evaluated. The values were obtained using the GLCM matrix and other in-built functions exclusive for MATLAB environment using the Scale Invariant Feature Transform (SIFT).

6.6 OUTPUT OF CNN CLASSIFIER

(1) SPECIES TYPE 1: PLASMODIUM FALCIPARUM

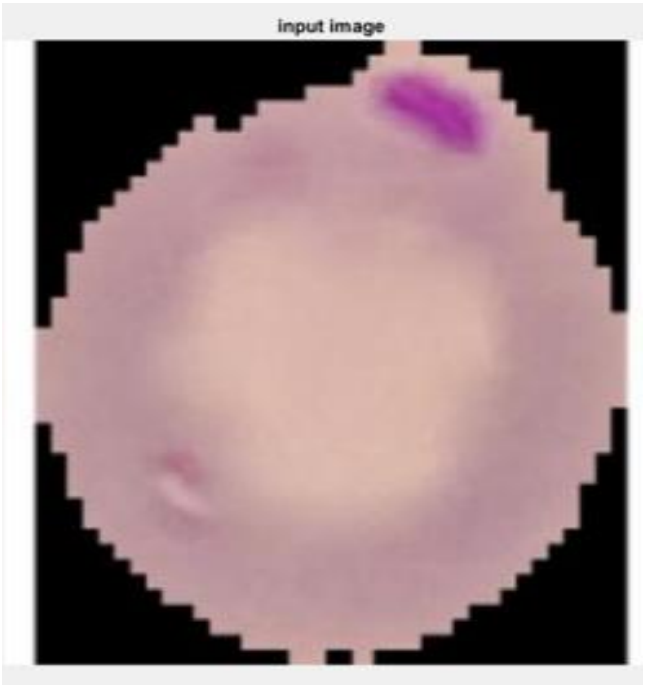


Fig. 6.11 Input image of blood sample of species type-1 (CNN)

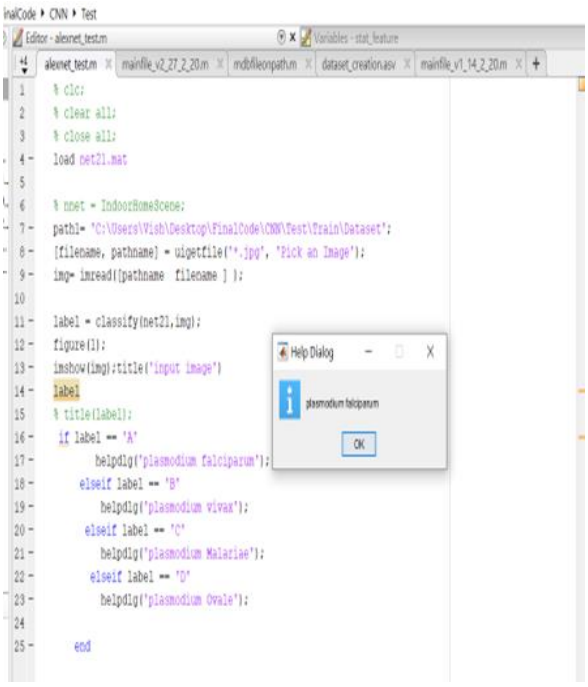


Fig. 6.12 Species type-1 malarial parasite identified (CNN)

Image acquisition process takes place in the CNN algorithm, which is an efficient and time-saving algorithm. Classification and identification processes are performed. The dialog box indicates the type of malaria parasite present in the input blood sample image using CNN algorithm.

(2) SPECIES TYPE 2: PLASMODIUM VIVAX

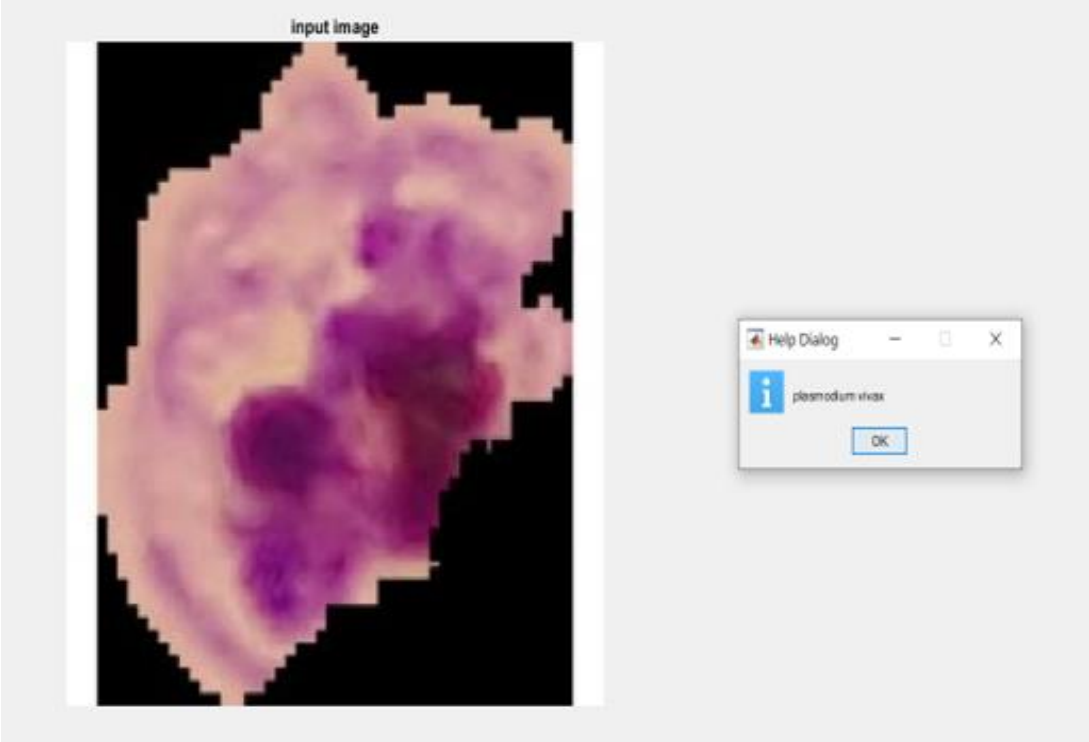


Fig. 6.13 Consolidated output of species type- 2 (CNN)

(3) SPECIES TYPE 3: PLASMODIUM MALARIAE

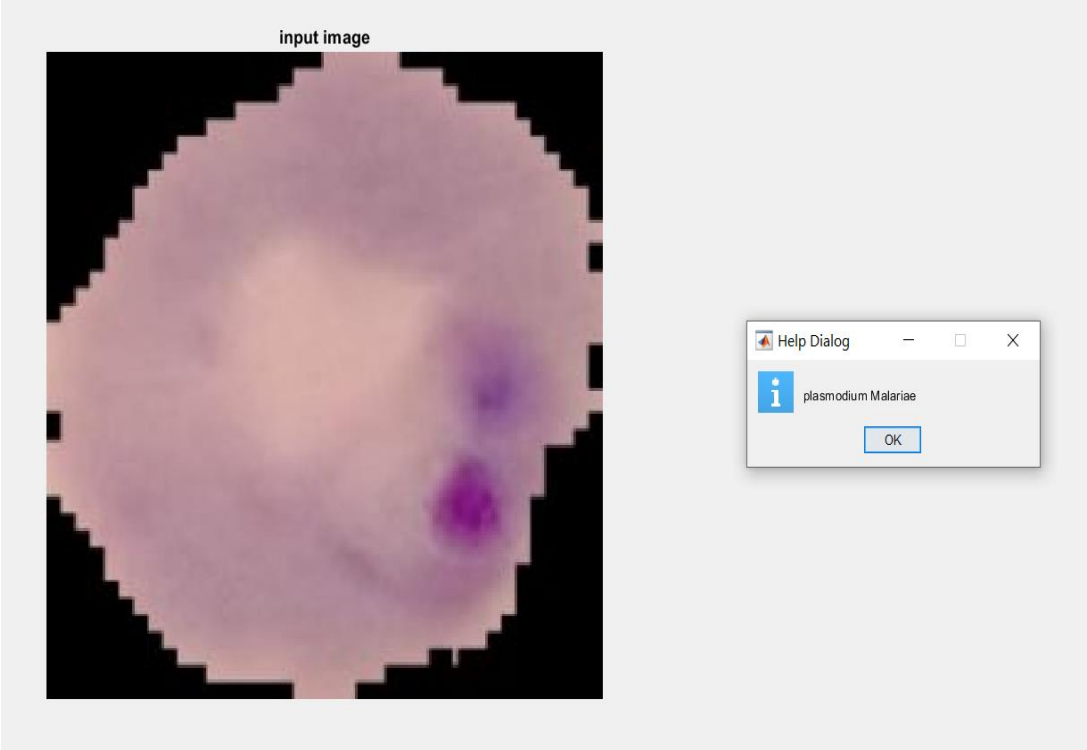


Fig. 6.14 Consolidated output of species type- 3 (CNN)

(4) SPECIES TYPE 4: PLASMODIUM OVALE

Fig. 6.15 Consolidated output of species type- 4 (CNN)

In CNN algorithm, every blood sample image from the acquired dataset undergoes the same procedural flow similar to the steps explained in the previous segment for species type-1: Plasmodium *falciparum*. In Fig. 6.13, 6.14 and 6.15, the consolidated output images of the other three types of species are displayed. Classification and identification processes occur almost instantly in CNN algorithm, which is preferred.

6.7 COMPARATIVE ANALYSIS

In the existing system, less accuracy is reasoned by the fact that minimum number of features are taken into consideration. Here, five features namely, contrast, correlation, homogeneity, energy and entropy are extracted. From the outputs obtained and the values of features extracted, two tables are drawn for better interpretation. With a pre-set 75% of the total acquired images used for training and the remaining 25% used for testing, the number of case types of existing and proposed systems was estimated and presented in the Table 6.2.

Table 6.2 Number of case types for existing and proposed systems

CASE TYPE	EXISTING SYSTEM	PROPOSED SYSTEM
TRUE POSITIVE	50	75
FALSE POSITIVE	8	3
FALSE NEGATIVE	10	5

Table 6.3 Accuracy calculation

TYPE	SVM(%)	CNN(%)
EXISTING SYSTEM	83.33	86.21
PROPOSED SYSTEM	93.75	96.15

The three case types are:

- True Positive- A test result that correctly indicates that the condition being tested for is present.
- False Positive- A test result which wrongly indicates that a particular condition is present.
- False Negative- A test result which wrongly indicates that a particular condition is absent.

The accuracy of existing system and proposed system, when susceptible to SVM and CNN algorithms was classified (Table 6.3). The existing system data was acquired from published journals. [4] [5] [7]

Accuracy calculation was done using the formula:

$$\textbf{Accuracy Calculation} = \left[\frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Negative}}\right] * \mathbf{100} \qquad \textit{6.1}$$

6.8 PERFORMANCE ANALYSIS: CHARTS

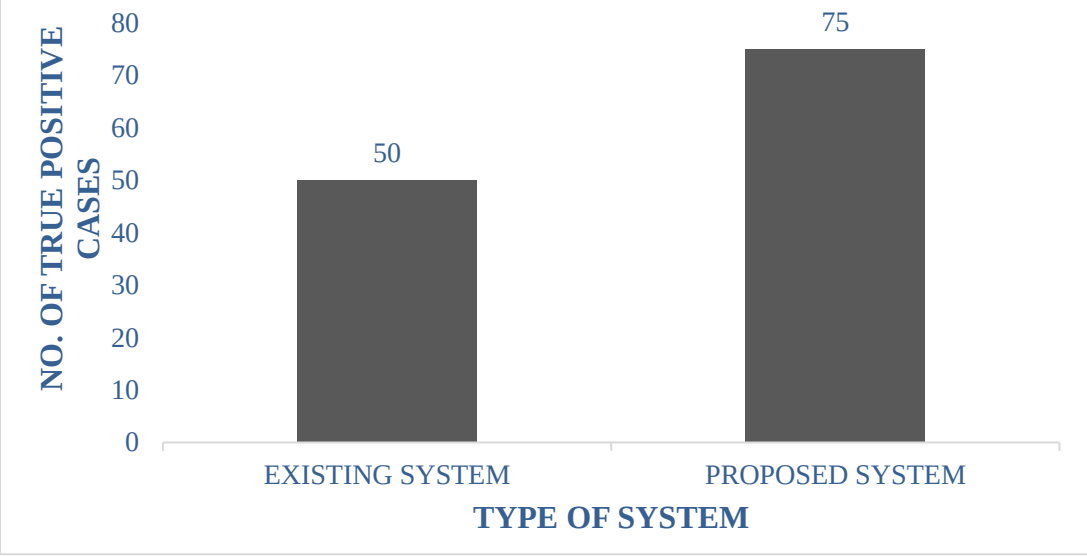


Fig. 6.16 Number of true positive cases Vs. Type of system

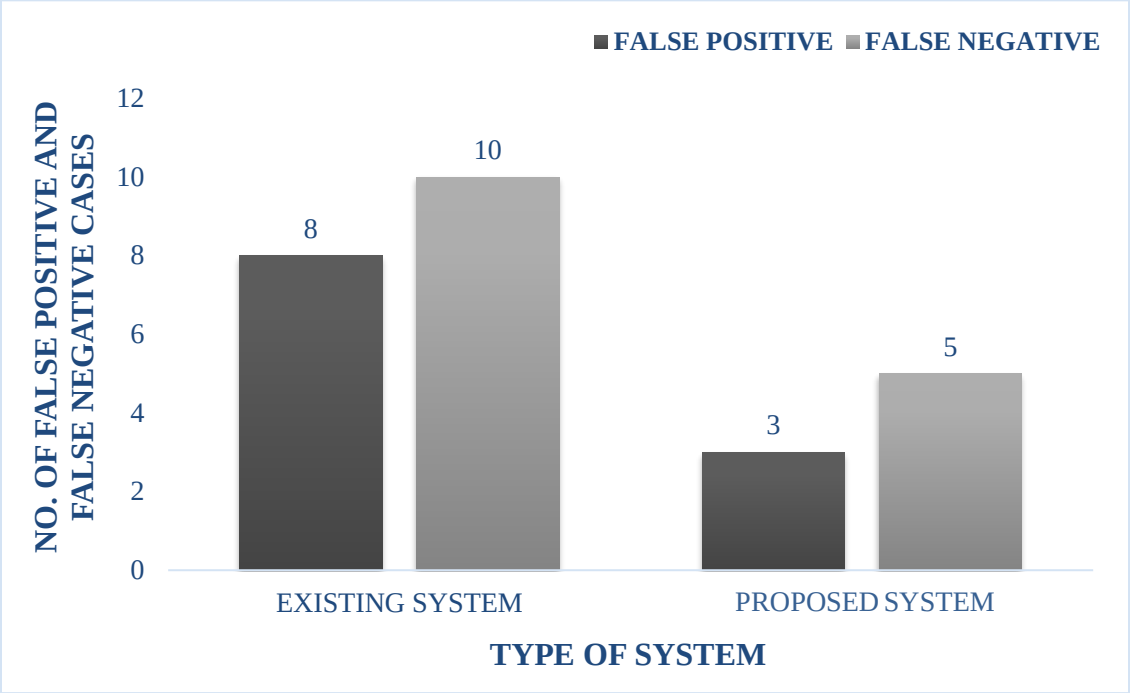


Fig. 6.17 Number of false positive and false negative cases Vs. Type of system

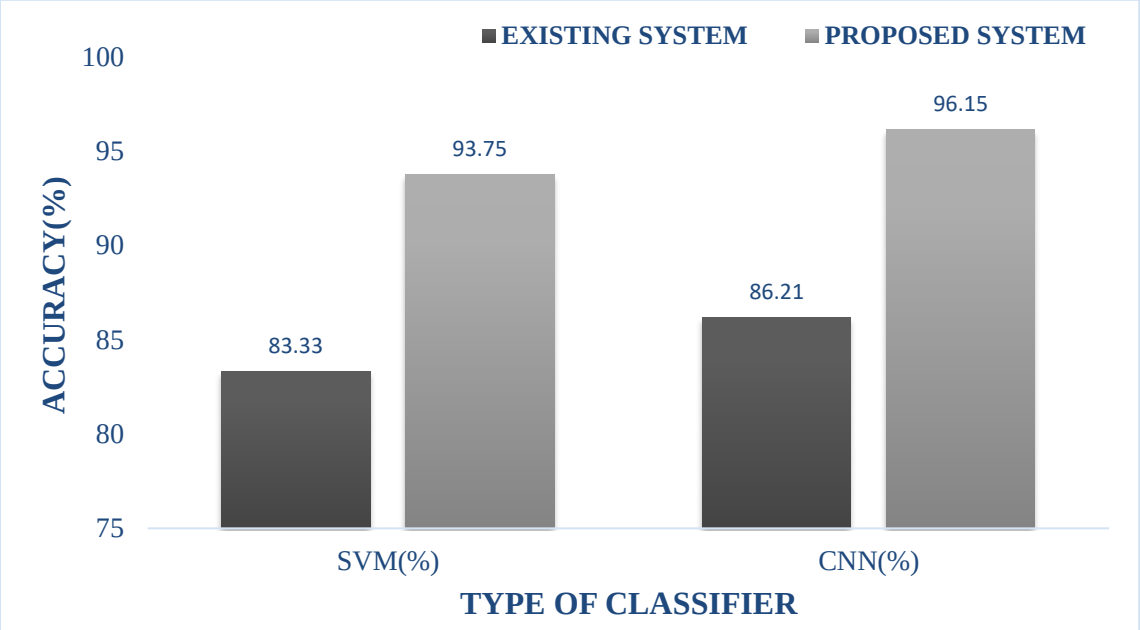


Fig. 6.18 Accuracy comparison between the two classifiers

Inference

Performance analysis was done between the existing and proposed systems and the results were represented using bar charts. In Fig. 6.16, the number of True Positive cases for the existing and proposed systems was assessed. A significant increase in the number of True Positive cases for the proposed system is represented. In Fig. 6.17, the number of false positive and false negative cases is represented against the type of systems considered. The proposed system was factually better than its counterpart based upon accuracy calculated.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

With the aid of automated diagnosis, the type of malaria causing parasites in input blood samples were identified (a. Plasmodium *falciparum,* b. Plasmodium *vivax,* c. Plasmodium *malariae,* d. Plasmodium *ovale)* using machine learning algorithm (SVM classifier) and deep learning algorithm (CNN classifier). Our proposal intends to provide an automated diagnosis of malarial parasites in blood cell images. A comparative analysis between the performance of both the classifiers was evaluated.

Drawing evidence from the comparative analysis, we can affirm that CNN algorithm was significantly more accurate and efficient in the detection of malarial parasites than SVM algorithm. The reason behind the better performance of CNN algorithm is because the prevalent algorithm has multiple layers of neural network architecture in which each layer interprets the input data in a different manner without requiring human intervention, whereas SVM operates on structured data (labelled data).

Thus, the proposed system is a low cost, more effective and time saving technique that could prove to be helpful in real time applications. As the number of features considered increases, accuracy of the detection process increases. Here, more than one Machine Learning classifier is used for detection of malarial parasites. Henceforth, the compatibility of blood samples towards the two different machine learning and deep learning algorithms were measured.

Conventional microscopy-based analysis requires intermittent human force which might cause parallax error and other human errors that affects the accurate detection of the disease. In the proposed technique, the desired outcomes can be reproduced whenever required unlike conventional microscopic analysis. In cases of far-flung

rural areas as experts may not be present to perform diagnosis, the proposed method could be helpful.

An extension of the proposed system can be utilized in the detection of other diseases, if properly maneuvered. It paves a path for the development of an automated microscope for computer-aided malaria screening. A major goal is also the acquisition of larger training and testing data sets for improvements in the robustness of this approach. As the dataset is limited only to the gametocyte stage of the malaria parasites, future research may include images of the full schizogony (including initial ring, trophozoite and schizont stage) of the parasites to build a more representative and discriminative model for species identification.

This procedure will be a stepping stone in malaria diagnosis, as it avoids the usual ways of converting images to different regions and thereby segmenting it. The method has been implemented in the MATLAB environment, for the flexibility and speed of prototyping the image processing options.

REFERENCES

[1] Daniel Maitethia Memeu et al., (2013) 'Detection of plasmodium parasites from images of thin blood smears', International Journal of Engineering Science and Innovative Technology (IJESIT), University of Nairobi, Nairobi, Kenya, Open Journal of Clinical Diagnostics-3, pp.183-194.

[2] Deepa, A.K. and Vineeta, P.G. (2014) 'Detection of Malarial Parasites in Blood Images', International Journal of Engineering Science and Innovative Technology (IJESIT), Vol.3, No.3.

[3] Erik Hablemeyer, Matthias Elter and Thorsten Zerfab (2011) 'Detection of malarial parasites in thick blood films', 33rd Annual International Conference.

[4] Hassan Abdelrahman Mohammed and Iman Abuel Maaly Abdelrahman (2017) 'Detection and Classification of Malaria in Thin Blood Slide Images', International Conference on Communication, Control, Computing and Electronics Engineering.

[5] Jyothi, R. and Sony, P.L. (2018) 'Automated Identification and Classification of Malarial Parasites in Thin Blood Smear Images', International Research Journal of Engineering and Technology.

[6] Khatri, K.M. et al., (2012) 'Image Processing Approach for Malarial Parasite Identification', International Journal of Computer Applications, National Conference on Growth of Technologies in Electronics, Telecom and Computers, India: GOTETC.

[7] Kristofer E. delas Penas, Pilarita T. Rivera and Prospero C. Naval Jr., (2017) 'Malaria Parasite Detection and Species Identification on Thin Blood Smears using a Convolutional Neural Network', IEEE/ACM International Conference on Connected Health.

[8] Mark C. Mushabe, Ronald Dendere and Tania S. Douglas, (2013) 'Automated Detection of Malaria in Giemsa-Stained Thin Blood Smears', Annual International Conference of the IEEE EMBS.

[9] Prasad, K. et al., (2012) 'Image Analysis Approach for Development of a Decision Support System for Detection of Malaria Parasites in Thin Blood Smear Images', Journal of Digital Imaging, Vol.25, No.4, pp.542–549.

[10] Saumya Kareem Reni, (2014) 'Automated Low-Cost Malaria Detection System in Thin Blood Slide Images Using Mobile Phones', Ph.D. dissertation, Department of Science and Technology, Westminster University, UK.

[11] Suradkar, P. (2013) 'Detection of Malarial Parasite in Blood Using Image Processing', International Journal of Engineering and Innovative Technology, Vol.2, No.10.

[12] CDC, (2016) 'Malaria parasites' [Online].

Available: https://www.cdc.gov/malaria/about/biology/parasites.html.

[13] learn zoology, (2014) 'Plasmodium Vivax' [Online].

Available: https://learnzoology.files.wordpress.com/2014/03/p-vivax-schizont.jpg.

[14] WHO, (2016) 'World Malaria Report 2014 Fact Sheet' [Online].

Available: http://www.who.int/malaria/publications/world_malaria_report_2014/ wmr2014 factsheet.pdf.

[15] Image database links:

https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria

https://lhncbc.nlm.nih.gov/publication/pub9932

APPENDICES

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING, MACHINE LEARNING AND DEEP LEARNING

[1]Mrs.A.Usha, [2]N.M.VishalKanna, [3]M.Swarna, [4]M.Shyam Ganesh
[1]Associate professor, [2]UG Student, [3]UG Student, [4]UG Student
Department of Electronics and Communication Engineering
Easwari Engineering College, Chennai, India

***Abstract*:** Malaria is a dreadful disease caused by single-celled microorganism instigating millions of death worldwide. Hence a rapid with accurate diagnosis and proper effectual medication is the need of the hour. The diagnosis of the infection starts with the microscopic examination of blood films. Diagnostic process automation with an intellectual framework which would diagnose malaria parasites may be successful in solving the above problem. Hence, several image processing techniques are used in this work to expose malaria from the microscopic images of Giemsa stained thin blood smear. For the identification of the derived SIFT (Scale Invariant Feature Transform) functions, a comparative study of SVM (Support Vector Machine), CNN (Convolutional Neural Network) is pursued. MATLAB (Matrix Laboratory) software is used for this image processing and classification processes. A recognition efficiency of 94% is obtained from SVM and 98% from CNN. Hence, CNN provides better efficiency when compared to SVM.

Keywords-Malarial parasites, Image processing, Feature extraction, SVM classifier, CNN classifier, Deep learning

I. INTRODUCTION:

Malaria is a mosquito-borne disease that affects humans and animals in different parts of the world causing misery and millions of deaths. A research undertaken in the field of medical diagnosis and pathological analysis is the automatic detection and categorization of malaria parasites in thin blood smear images. The current malaria diagnosis approach is traditional microscopy which is time consuming, labor intensive and results depend on the researcher's skills. Normal healthy blood cell (Fig-1) and malaria affected blood cell (Fig-2) can be differentiated very well. The research is aimed at developing algorithms to provide an efficient diagnostic tool for quantitative malaria infection analysis.

Malarial Parasites: Plasmodium vivax, Plasmodium ovale, Plasmodium falciparum, Plasmodium malariae are parasites instigating malaria. Among the four malarial parasites, Plasmodium falciparum is the deadliest. Plasmodium vivax is widely distributed and the infections caused by this parasite leads to severe disease due to splenomegaly. Plasmodium malariae causes "benign malaria", which is less dangerous when compared to other infections. Plasmodium ovale causes tertian malaria. All these parasites are transmitted through the female anepholes mosquitoes.

Different stages in the growth and development of these parasites (Fig-3) are Ring stage, Trophozoite stage, schizont stage, and Gametocyte stage. Trophozoites mature in Ring stage into schizonts that rupture releasing merozoites. Trophozoite stage is the activated, feeding stage of the protozoa's life cycle. The parasite replicates its DNA several times at the schizont level, and several mitotic divisions occur asynchronously. Gametocytes are the precursors of male and female gametes where, via the developmental transition from asexual replication in erythrocytes, malaria parasites emerge in the human host.

The rest of this paper is organized as follows. We provide review of literature, methodology followed by results and discussion, comparative analysis and performance analysis. We close with a conclusion.

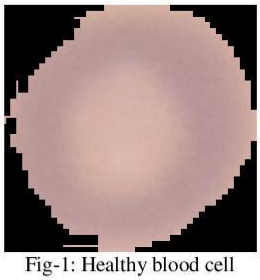


Fig-1: Healthy blood cell

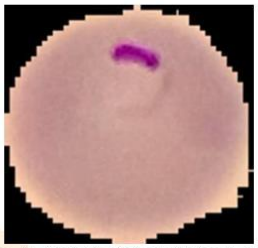


Fig-2: Malaria infected blood cell

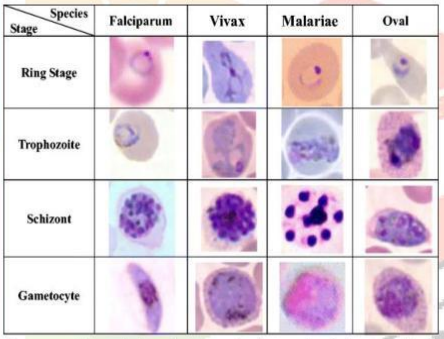


Fig-3: Stages in the growth and development of a malarial parasite in a blood cell [5]

I. REVIEW OF LITERATURE:

1. Min Liu et al., (2017) '**ABO Blood Group Detection Based on Image Processing Technology'** proposed a quick, accurate and reliable blood group judgment system based on the ABO blood group image features. To obtain the best approximation of the original image, the median filter is used to remove the noise. The characteristic parameters of the blood group ABO are determined according to the image's gray level distribution.
2. Abubakar Yamin et al., (2017) '**Image Processing Based Detection & Classification of Blood Group Using Color Images'** proposed a program that would use image recognition to identify blood groups. Preprocessing methods, HSV Luminance and morphological operations are measures used to identify the type of blood group using image processing techniques.
3. Alireza Karimian et al., (2016) '**Design a new algorithm to count white blood cells for classification Leukemic Blood Image using machine vision system'** proposed that color space transfer models can be used to classify the white blood cells. The community of leukocytes is divided by division of watershed conversion and cleanup of the picture is completed.
4. Corentin Dallet et al., (2014) '**Real Time Blood Image Processing Application for Malaria Diagnosis Using Mobile Phones'** proposed a simple and stable Android cell phone application platform to analyze blood sample images. The application is based on MATLAB 's novel Annular Ring Ratio process, checked and validated. The system detects blood components including Red Blood Cells (RBCs), White Blood Cells (WBCs), parasitemia of the infected RBC parasites.
5. I.Kale et al., (2011) '**A Novel Method to Count the Red Blood Cells in Thin Blood Films',** hereby, a novel concept to classify the total number of red blood cells (RBCs) and their position in a Giemsa stained thin blood film picture was proposed. The method uses basic knowledge on component cell structure and brightness to detect the RBCs in the picture. It removes the segmentation procedures used to segment the cells into the microscopic image and prevents pre-processing of images to deal with non-uniform illumination before cell detection.

I. METHODOLOGY:

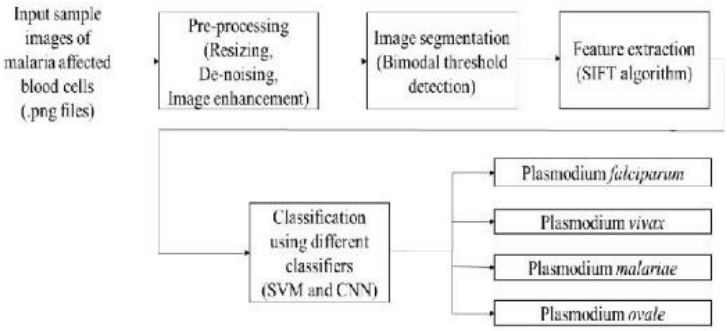


Fig-4: Block diagram of malarial parasite detection

System architecture (Fig-4) used for malaria parasite detection involves the following main steps: image acquisition (input image), preprocessing, segmentation, feature extraction, classification and final result (identification of the malaria causing parasite). Pre-processing involves eliminating image noise, segmenting to distinguish foreground and background pixels, extracting features such as contrast, similarity, homogeneity, capacity, entropy for the classification of malaria-infected cells and non-infected cells and further classification of the malarial parasite using SVM and CNN. From the outcome of the classification system using SVM and CNN, the final result i.e., the species of the malarial parasite in the given input image (thin blood smear) is identified and the results produced by both the classifiers SVM and CNN are compared for accuracy. MATLAB is implemented in all stages of the method.

- ***IMAGE ACQUISITION*** *(input image)*: A total of 400 photographs of thin blood smears stained by giesma was collected as a dataset [4]. Such photos have various magnifications and scale characteristics.

- ***PREPROCESSING***: The aim of pre-processing is to remove unwanted objects and noise from the image to facilitate image segmentation into meaningful regions. Preprocessing steps include: transforming colored images into grayscale presentation, background estimation using morphological opening technique, removing the background image from the original image, improving image contrast and transformation of grayscale images into binary images with thresholding techniques, turning images into negative presentation.

- ***SEGMENTATION:*** Segmentation is the mechanism whereby the image is partitioned into different parts. The image is divided into foreground and background here. It is for isolating the RBCs. Simplest segmentation algorithm used here is the selection of an acceptable threshold of intensity. All pixels with a value greater than a given threshold value are marked as the interest zone and the lower-value pixels are labeled as background pixels. Such a distribution is considered bimodal since there are two mode values: one for background and another one for feature. In certain cases, after the foreground has been stripped, there are still clumped RBCs and the iterative separation of clumps is done as such. The individual cells are retained while the clumps extracted are separated. The boundary is extracted for each clump and their curvature is determined. For each object, the boundary curvature vector is analyzed for regional maxima, indicating concavity points from which potential split lines will be drawn. An iterative process is used for the splitting of RBC clumps. The original individual RBCs and resulting individual RBCs from the process of clump splitting are merged into a binary RBC image. Unnecessarily separated cells are morphologically rebuilt and inserted back into the RBC binary image.

- ***FEATURE EXTRACTION:*** Extraction of the feature is used to extract specific characteristics from images. Here SIFT is used to extract characteristics. SIFT is a computer vision algorithm for identifying and defining local features in images. SIFT key points of objects are first taken from a collection of reference images and saved in the database. An object is recognized in a new image by comparing each feature from the new image to this database individually, and identifying matching features for candidates based on Euclidean distance from their feature vectors. In the new picture, subsets of key points agreeing on the object and its position, size and orientation are selected from the full set of matches to filter out great matches.

- ***CLASSIFICATION:*** This is the final step in which the input picture features such as contrast, correlation, homogeneity, energy and entropy extracted using SIFT is compared and then listed with that of the infected RBC image features. The classifier is trained for this by providing the features extracted in the preceding level. Classification is carried out using SVM and CNN classifiers in the proposed methodology and then their respective output is measured and analysed.

(A)Support Vector Machine (SVM):

A hyper plane or set of hyper planes are constructed by SVM in a high- or infinite-dimensional space (Fig-5), which may be used for classification , regression, or other tasks such as detection of outliers. The classifier is trained here to enable classification of the input images into infected and normal RBC images. This is how you detect the presence of malaria parasite in the blood.

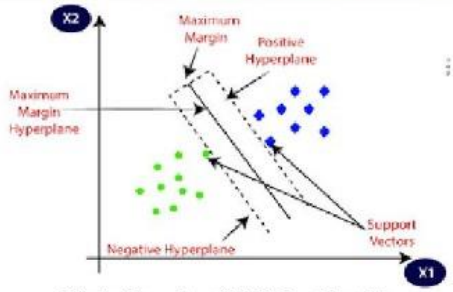


Fig-5: Hyperplane SVM algorithm [6]

(B)Convolutional Neural Network (CNN):

The CNN approach is second. Comparison of the output of the proposed work is made using CNN to determine which of these would yield the best results. CNN is a class of profound, feed-forward artificial neural networks that use a multilayer perceptron (Fig-6) variation designed to allow minimal preprocessing. These are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their characteristics of shared-weight architecture and invariance in translation. Compared with other image classification algorithms, CNNs use very little pre-processing. This ensures that the network learns from the filters that were hand-engineered in conventional algorithms.

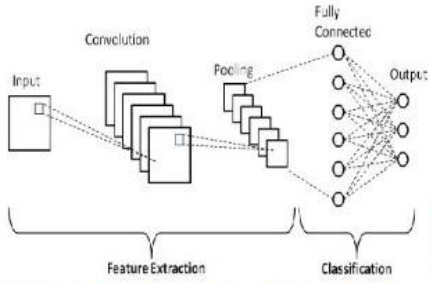


Fig-6: Neural network layer structure [7]

- ***IDENTIFICATION OF PARASITE***: From the outcome of the classification system using SVM and CNN, the final result i.e., the species of the malarial parasite in the given input image (thin blood smear) is identified and the results produced by both the classifiers SVM and CNN are compared for accuracy.

RESULTS AND DISCUSSION:

A dataset containing blood sample images is acquired from [15] in .png format and then converted to .jpeg format. 100 images per species type are used for SVM algorithm and 1000 images are used for CNN algorithm. Out of the total images in the dataset, 75% of them were used for training and 25% of them were used for testing.

(i) Output of SVM classifier:

(A)Species type1-Plasmodium falciparum

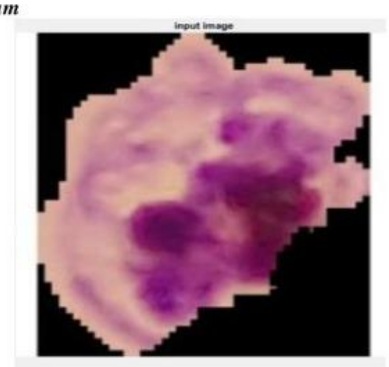


Fig-7: Input image of blood sample of species type-1 (SVM)

The blood sample image (Fig-7) is fed as input to the SVM algorithm. JPEG format conversion and Gray-scale conversion takes place in order to support pre-processing techniques.

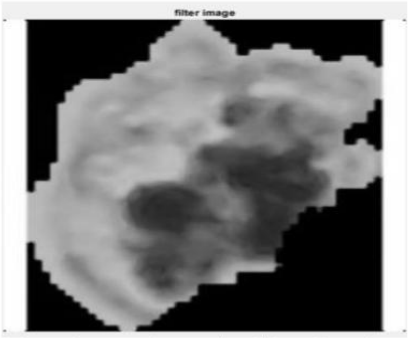


Fig-8: Filtered input image using median filter of species type-1 (SVM)

Resizing of images, de-noising of images (Fig-8) with the courtesy of median filter, contrast enhancement and morphological operations (dilation and erosion) are applied to the subjected image.

Fig-9: Binary converted image of the corresponding filtered image of species type-1 (SVM)

Binary conversion takes place through bi-modal threshold detection (Fig-9). Iterative method of clump splitting is conducted to recognize the cells of interest.

Fig-10: Segmented image of species type-1 (SVM)

Image segmentation process takes place (Fig-10). Feature extraction occurs with the help of SIFT algorithm. Outlier detection removes futile details in the background.

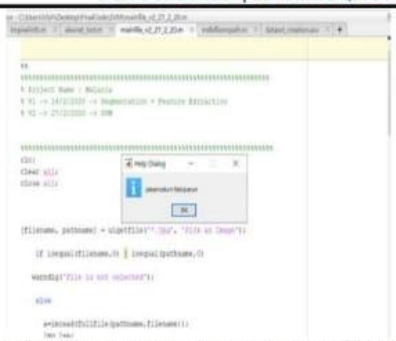


Fig-11: Species type-1 malarial parasite identified (SVM)

Final stage is where classification process takes place. Identification process is ensued by a dialog box indicating the type of malarial species (Fig-11).

In SVM algorithm, every blood sample image from the acquired dataset undergoes the same procedural flow similar to the steps explained in the previous segment for species type-1: Plasmodium *falciparum*. Species types 2, 3 and 4 are subjected to the same conditions as briefly explained earlier.

(B)Analysis of statistical features:

Statistical features such as Contrast, Correlation, Energy, Homogeneity, and Entropy of the four types of species are evaluated. The values are obtained using the GLCM matrix and other in-built functions exclusive for MATLAB environment using the Scale Invariant Feature Transform (SIFT).

Table-1: Statistical feature analysis of the four type of species

Species type / Statistical features	Plasmodium *falciparum*	Plasmodium *vivax*	Plasmodium *malariae*	Plasmodium *ovale*
Contrast	204.8366	234.2128	146.0947	210.5449
Correlation	0.5011	0.5030	0.5649	0.9107
Energy	0.6830	0.7486	0.7872	0.4943
Homogeneity	0.8968	0.9200	0.9316	0.8372
Entropy	1.8376	1.4551	1.3216	3.5631

- ***Contrast (CTR):*** Obtained by the change in brightness of image from one pixel to another, pointed by the data cursor (in-built feature).

$$CTR = \sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=1}^{G} \sum_{j=1}^{G} P(i,j) \right\}, |i-j| = n \qquad (1)$$

Where, P- intensity value of the pixel
n- order
G- total number of pixels in the image
(i, j)- co-ordinates of the pixel

- ***Correlation (COR):*** Correlation is a measure of gray level linear dependence between the pixels at the specified positions relative to each other.

$$COR = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j} \quad (2)$$

$$\mu_j = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} jP(i,j) \quad (3)$$

$$\mu_i = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} iP(i,j) \quad (4)$$

$$\sigma_j = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} (j-\mu_j)^2 P(i,j) \quad (5)$$

$$\sigma_i = \sum_{i-0}^{G-1} \sum_{j-0}^{G-1} (j-\mu_i)^2 P(i,j) \quad (6)$$

Where, P- intensity value of the pixel
(i,j)- co-ordinates of the pixel
G- total number of pixels in the image
μ- mean
σ – variance

- ***Homogeneity (IDM):*** It is the uniformity in composition. Also known as Inverse Difference Moment (IDM).

$$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i,j) \quad (7)$$

Where, P- intensity value of the pixel
(i,j)- co-ordinates of the pixel
G- total number of pixels in the image

- ***Energy (E):*** Determines the intensity of pixels. Calculated as follows:

$$E = \sum_{i=0}^{G-1} [P(i)]^2 \quad (8)$$

Where, P- intensity value of the pixel
(i,j)- co-ordinates of the pixel
G- total number of pixels in the image

- ***Entropy (H):*** Statistical measure of randomness used to characterize texture of image. Calculated as follows:

$$H = -\sum_{i=0}^{G-1} P(i) \log_2 P(i) \quad (9)$$

Where, P- intensity value of the pixel
(i,j)- co-ordinates of the pixel
G- total number of pixels in the image

(ii)Output of CNN classifier:

(A)Species type1-Plasmodium falciparum

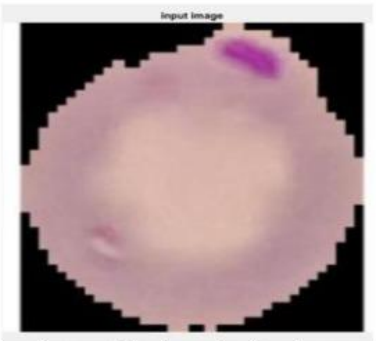


Fig-12: Input image of blood sample of species type-1 (CNN)

Image acquisition (Fig-12) process takes place in the CNN algorithm, which is an efficient and time-saving algorithm.

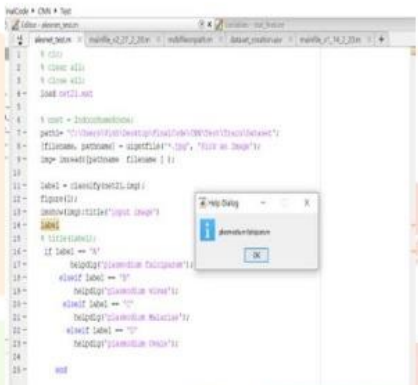


Fig-13: Species type-1 malarial parasite identified (CNN)

The type of malaria parasite present in the input blood sample image is identified (Fig-13) using CNN algorithm.
In CNN algorithm, every blood sample image from the acquired dataset undergoes the same procedural flow similar to the steps explained in the previous segment for species type-1: Plasmodium *falciparum*. Species types 2, 3 and 4 are subjected to the same conditions as briefly explained earlier.

'. COMPARATIVE ANALYSIS:

Table-2: Number of case types for existing and proposed systems

CASE TYPE	EXISTING SYSTEM	PROPOSED SYSTEM
TRUE POSITIVE	50	75
FALSE POSITIVE	8	3
FALSE NEGATIVE	10	5

Table-3: Accuracy calculation

TYPE	SVM (%)	CNN (%)
EXISTING SYSTEM	83.33	86.21
PROPOSED SYSTEM	93.75	96.15

Based on the values of the features extracted using GLCM matrix and SIFT algorithm, two tables are drawn for better interpretation. With a pre-set 75% of the total acquired images used for training and the remaining 25% used for testing, the number of case types of existing and proposed systems are estimated and presented in the table. The three case types are:

- True Positive- A test result that correctly indicates that the condition being tested for is present.
- False Positive- A test result which wrongly indicates that a particular condition is present.
- False Negative- A test result which wrongly indicates that a particular condition is absent.

The accuracy of existing system and proposed system, when subjected to SVM and CNN algorithms is calculated using formula:

$$\text{Accuracy} = \left[\frac{True\ Positive}{True\ Positive + False\ Negative}\right] * 100 \quad (10)$$

The existing system data is acquired from [1] [2] [3].

I. PERFORMANCE ANALYSIS:

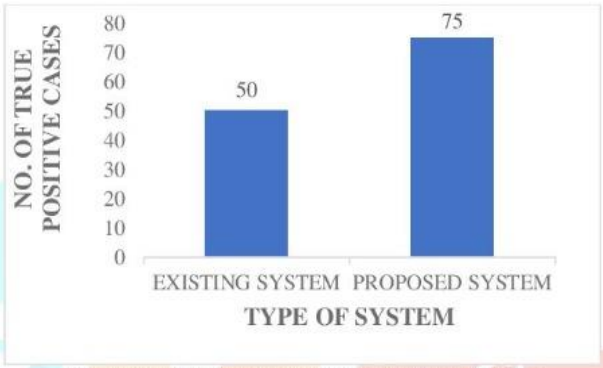


Fig-14: Number of true positive cases Vs. Type of system

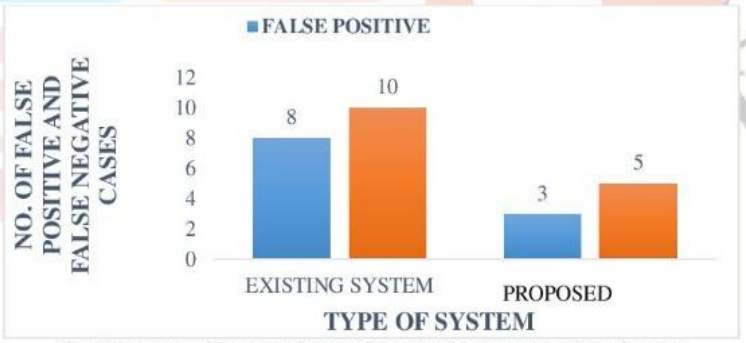


Fig-15: Number of false positive and false negative cases Vs. Type of system

Fig-16: Accuracy classification between the two classifiers

Performance analysis is done between the existing and proposed systems and the results are represented using bar charts. The number of True Positive cases for the existing and proposed systems are assessed (Fig-14). A significant increase in the number of True Positive cases for the proposed system is represented. The number of false positive and false negative cases is represented against the type of

systems considered (Fig-15). The proposed system is factually better than its counterpart. The accuracy (in %) value of the two classifiers are compared (Fig-16).

I. CONCLUSION:

This paper proposes an automated diagnosis of malarial parasites present in the blood cell images. Hence, the parallax errors and time consumption which are the major drawbacks of the conventional microscopy method can be reduced to a greater extent. The classification process to identify the species of the malarial parasite in the blood cell image is done by two classifiers-SVM and CNN. From the outcomes obtained, CNN was found to be more effective in the classification with higher accuracy than SVM.

This proposal can further be extended by acquiring large number of sample images i.e., larger training and test datasets than which are available now and considering more features of the input image. This can improve the accuracy and reduce the errors in detection further.

REFERENCES:

[1] Hassan Abdelrahman Mohammed and Iman Abuel Maaly Abdelrahman (2017) 'Detection and Classification of Malaria in Thin Blood Slide Images', International Conference on Communication, Control, Computing and Electronics Engineering.

[2] Jyothi, R. and Sony, P.L. (2018) 'Automated Identification and Classification of Malarial Parasites in Thin Blood Smear Images', International Research Journal of Engineering and Technology.

[3] Kristofer E. delas Penas, Pilarita T. Rivera and Prospero C. Naval Jr., (2017) 'Malaria Parasite Detection and Species Identification on Thin Blood Smears using a Convolutional Neural Network', IEEE/ACM International Conference on Connected Health.

[4] Dataset of giesma stained thin blood cell images (input images). URL:
https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria
https://lhncbc.nlm.nih.gov/publication/pub9932

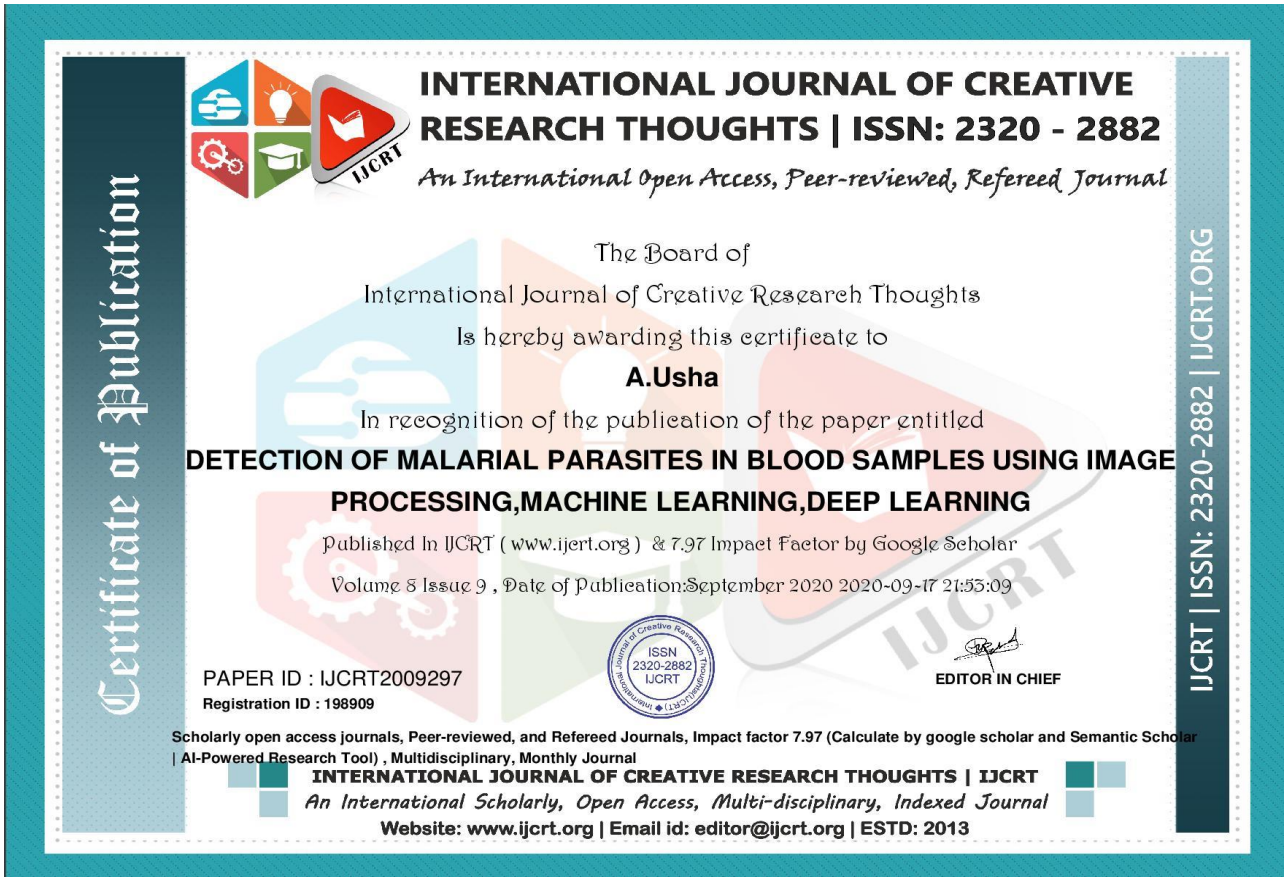
[5]Stages in the growth and development of a malarial parasite in a blood cell. URL:

https://www.biologydiscussion.com/parasitology/parasiticprotozoa/histologyof-malarial-parasite-sporozoa/62006

[6] Hyperplane SVM algorithm. URL:

https://www.javatpoint.com/machine-learning-supportvector-machine-algorithm

[7]Neural network layer structure. URL:
https://missinglink.ai/guides/convolutional-neuralnetworks/convolutional-neural-network-architecture-forgingpathways-future/


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
A.Usha
In recognition of the publication of the paper entitled
DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING,MACHINE LEARNING,DEEP LEARNING
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-17 21:55:09
PAPER ID : IJCRT2009297
Registration ID : 198909
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
N.M.VishalKanna
In recognition of the publication of the paper entitled
DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING,MACHINE LEARNING,DEEP LEARNING
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-17 21:55:09
PAPER ID : IJCRT2009297
Registration ID : 198909
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
M.Swarna
In recognition of the publication of the paper entitled
DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING,MACHINE LEARNING,DEEP LEARNING
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-17 21:53:09
PAPER ID : IJCRT2009297
Registration ID : 198909
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
IJCRT | ISSN: 2320-2882 | IJCRT.ORG
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
M.Shyam Ganesh
In recognition of the publication of the paper entitled
DETECTION OF MALARIAL PARASITES IN BLOOD SAMPLES USING IMAGE PROCESSING,MACHINE LEARNING,DEEP LEARNING
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-17 21:53:09
PAPER ID : IJCRT2009297
Registration ID : 198909
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
IJCRT | ISSN: 2320-2882 | IJCRT.ORG
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013