DE-42 (EE) ABDUL MOIZ, MUSTAFA, JAVERIA HASSAAN

# AI-BASED CBC

**COLLEGE OF**

**ELECTRICAL AND MECHANICAL ENGINEERING**

**NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY RAWALPINDI**

**2024**

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

# PROJECT REPORT

# DE-42 EE

## AI-BASED CBC

Submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree of

**Bachelor of Engineering**

**in**

**Electrical**

**2024**

**Sponsoring DS:** **Submitted By:**


بسم الله الرحمن الرحيم
١٤١٨

# CERTIFICATE OF APPROVAL

It is to certify that the project **"AI-BASED CBC"** was done by **NC Abdul Moiz**, **NC Muhammad Mustafa Shahid**, **NC Javeria Sultan, and ASC Muhammad Hassaan Kashif** under the supervision of **Dr. Mahrukh Liaqat.**

This project is submitted to **Department of Electrical Engineering**, College of Electrical and Mechanical Engineering (Peshawar Road Rawalpindi), National University ofSciences and Technology, Pakistan in partial fulfilment of requirements for the degree of Bachelor of Electrical Engineering.

**Students:**

1. **Abdul Moiz**

   NUST ID: ___345108___ Signature: ____________________

2. **Muhammad Mustafa Shahid**

   NUST ID: ___338950___ Signature: ____________________

3. **Javeria Sultan**

   NUST ID: ___339744___ Signature: ____________________

4. **Muhammad Hassaan Kashif**

   NUST ID: ___359571___ Signature: ____________________

**APPROVED BY:**

Project Supervisor: ______________________________ Date: ______________

**Dr. Mahrukh Liaqat**

# DECLARATION

We affirm that the content presented in this Project Thesis is original and has not been submitted in support of any other degree or qualification at this or any other educational institution. We acknowledge that any act of plagiarism will result in full responsibility and may lead to disciplinary action, including the potential cancellation of our degree, based on theseverity of the offense.

1. **Abdul Moiz** ____________________________

2. **Muhammad Mustafa Shahid** ______________________________

3. **Javeria Sultan** __________________________

4. **Muhammad Hassaan Kashif** ___________________________

# COPYRIGHT STATEMENT

# ACKNOWLEDGEMENTS

To begin with, we are very appreciative to Allah Ta'ala for the moisture and the strength which He used to make us capable of dealing and topping the crisis that we found ourselves during this venture. However, aside from the two of us the directions and assisstance of each and everyone is critical to the success of any enterprise. We are extremely grateful towards Dr. Mahrukh Liaqat, our instructor and mentor, without whose diligent guidance, relentless support and inspiration our success would have been unthinkable.

Additionally, the bond of love between precious parents must also be given the utmost respect, because their unrelenting support and endurance were equally significant in our personal growth. Not once can they appear during a time we did need them.

In the last, we take the opportunity to thank the Electrical Department of Engineers which assisted our representatives throughout our studies and gave us opportunity to prove ourselves in the field. It will be my sincere pleasure to say thanks to instructors and advisors who had been in the front line in making available free materials and rendering support to us.

# ABSTRACT

In the domain of health monitoring, there is a dire need for accurate and innovative solutions that can optimize the efficiency and make the medical procedures cost inefficient. Our project introduces an innovative approach by incorporating Artificial Intelligence (AI) and Embedded Systems to assemble a Complete Blood Count (CBC) kit. Utilizing advanced machine learning algorithms to swiftly and effectively analyze real-time blood samples making it one of its kind. Personalized health assessment, early anomaly detection, vitals trend analysis and empowering users with an accessible health management tool are our value proposition. Integration of Raspberry Pi 5 and camera verifies its maneuverability and accessibility. Comprehensive report generation via web application, fulfilling the purpose of save, review and research. This advanced technology ensures sustainability for diverse healthcare environment, facilitating accuracy and cost efficient solutions to crucial health monitoring practices. Cloud computation covering for the resources cost to run this project, provides safe storage and computational power to tackle this tool's need in divergent environments. This project represents a compelling contribution to the growth of personalized diagnostics and wellness management.

# SUSTAINABLE DEVELOPMENT GOALS

The project that targets the design for the Complete Blood Count is pursuing SDG 9, which addresses resilient infrastructure, industrialization, and innovation.


*Figure 1 SDG-9*

The objective of the project is to develop a sophisticated blood count kit which involves the utilization of image enhancement and machine learning algorithms that by this way improve computational performance and accuracy. This array enabled the creation of a highly specialized and modern structure because of its particular setup which was made for faster image enhancing algorithms.

Offering novel medical diagnosis with the implementation of image processing and machine learning algorithms that are a part of the technology can be as well. Such innovation could thus intensify healthcare performance by making counting blood cells faster, and with greater accuracy.

Cloud computation consisting of both software and hardware comes in to play alongside this as well to help enhance the creativity of the designers. It relies on best in class AI technology infrastructure for faster and more precise blood analyzing so the demand for this technology in areas like vision applications or the cloud computing could go up.

Primarily, we are also developing sustainable industrialization in the health sector by creating a blood counting kit that can be used by medical providers. This is just one of the ways sustainable industrialization is promoted in the health sector. It would be beneficial to the resources utilization process to computerize and upgrade the procedures used in determining the white blood cell activity. This can help in reducing the need for human efforts thereby making them more efficient besides occupying less tine.

Our initiative significantly contributes to Sustainable Development Goal 3 (SDG 3): Health Issues.


*Figure 2 SDG-3*

The improvement of Diagnosis Technology will be done through generating a blood count kit using image processing with machine learning algorithms to improve diagnostics accuracy. Blood cell counting is a crucial performance, for instance, to detect Anemia, infections and blood disorders, among other conditions because precise and efficient one is vital. As health outcomes lead to timely and correct diagnostics, diagnostic tools advance.

It as Well Is a Health Care Access Cultural Route. The blood cell counting system's pocket friendly solution we build in our project will eventually enable people to have access to healthcare services that are convenient and affordable. What it can do is lessening the workload of medical workers in countries with a lack of funds on hand by means of providing an advanced technology and personal experience to test individuals using this kit and making the services accessible.

# TABLE OF CONTENTS:

## Contents

CERTIFICATE OF APPROVAL ........ iv
DECLRATION ........ v
COPYRIGHT STATEMENT: ........ vi
ACKNOWLEDGMENTS ........ vii
ABSTRACT ........ viii
SUSTAINABLE DEVELOPMENT GOALS ........ ix
TABLE OF CONTENTS ........ xi
TABLE OF FIGURES ........ xiii
LIST OF ABBREVIATIONS ........ xii
1. Chapter 1: INTRODUCTION ........ 1
1.1. Project Overview: ........ 2
1.2. Problem Statement ........ 2
1.3. Approach ........ 4
1.3.1. Dataset Collection: ........ 4
1.3.2. Image Processing: ........ 4
1.3.3. Model Selection: ........ 4
1.3.4. Model Training: ........ 5
1.3.5. Web-App Development and Cloud Computation: ........ 5
1.3.6. Hardware Development: ........ 5
1.3.7. Real Time Testing: ........ 5
1.4. Objectives ........ 5
1.5. Deliverables ........ 6
1.6. Organization of Thesis: ........ 7
2. Chapter 2: Literature Review ........ 8
2.1. Background ........ 9
2.2. Introduction to Complete Blood Count (CBC) ........ 9
2.2.1. Blood Composition ........ 10
2.3. Related Work ........ 11
2.4. Image Processing ........ 13
2.4.1. Image Enhancement ........ 13
2.4.2. Grey Scale Conversion ........ 14
2.4.3. Histogram Equalization ........ 15

2.4.4. Gaussian Smoothening Filter……………………………………………………………...16
2.4.5. Edge Sharpening Filter………………………………………………………………….18
2.4.6. Impact of Image Processing on Output…………………………………………………20
3. Chapter 3: Design and Development ....................................................................................23
3.1. Design Flow ...........................................................................................................................25
3.2. Hardware and Software part..................................................................................................27
3.3. Software part ..........................................................................................................................29
3.3.1. Model Development.........................................................................................................29
3.2.2. Web Application Development .......................................................................................33
3.4. You Only Look Once (YOLO) v8 Nano……………………………………………………38
3.5. Hardware designing using Raspberry Pi 5 ..........................................................................40
3.6. Software/Hardware integration: ............................................................................................43
3.7. Hardware Requirement .........................................................................................................44
3.8. Software Requirement............................................................................................................44
4. Chapter 4: Result Analysis.......................................................................................................45
4.1. Results of Image Processing using Python ...........................................................................46
4.2. Software Results: Cells Detection and Classification using YOLO v8 Nano......................50
4.3. Model Validation ...................................................................................................................51
4.4. Calculations............................................................................................................................52
5. Chapter 5: Conclusion..............................................................................................................56
5.1. Overview.................................................................................................................................57
5.2. Objective Achieved/ Achievements.......................................................................................57
5.3. Limitations .............................................................................................................................57
5.4. Applications ...........................................................................................................................58
5.4.1. Medical Diagnostics...........................................................................................................58
5.4.2. Telemedicine.......................................................................................................................58
5.4.3. Research and Development................................................................................................58
5.4.4. Educational Purposes ........................................................................................................59
5.4.5. Quality Control in Laboratories ........................................................................................59
APPENDIX.......................................................................................................................................61
REFERENCES..................................................................................................................................63

# Table of Figures:

Figure 1 : SDG-9 ........ ix
Figure 2: SDG-3 ........ x
Figure 3 : Original Image ........ 13
Figure 4 : Enhanced Image ........ 13
Figure 5 : Original Image ........ 15
Figure 6 : Histogram Equalized ........ 15
Figure 7 : Original Image ........ 18
Figure 8 : Gaussian Smooth Image ........ 18
Figure 9 : Original Image ........ 20
Figure 10 : Edge Sharpened Image ........ 20
Figure 11 : Design Flow ........ 27
Figure 12 : Original Image ........ 46
Figure 13 : Grey Scale Image ........ 46
Figure 14 : Original Image ........ 47
Figure 15 : Histogram Equalized ........ 47
Figure 16 : Gaussian Blur ........ 47
Figure 17 : Gaussian Smooth Image ........ 47
Figure 18 : Cells Detection ........ 51
Figure 19: Cells Classification ........ 51
Figure 20 : Model Validation ........ 52

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **CBC** | Complete Blood Count |
| **RBCs** | Red Blood Cells |
| **WBCs** | White Blood Cells |
| **Plts** | Platelets |
| **AI** | Artificial Intelligence |
| **YOLO** | You Only Look Once |
| **CNN** | Convolutional Neural Network |
| **GUI** | Graphical User Interface |
| **CLAHE** | Contrast Limited Adaptive Histogram Equalization |
| **AHE** | Adaptive Histogram Equalization |
| **CDF** | Cumulative Distribution Function |
| **ADAMW** | Adaptive Torque Estimation with Weight Decay |
| **HTML** | Hyper Text Markup Language |
| **CSS** | Cascading Style Sheets |

# CHAPTER # 01

# INTRODUCTION

1.1. PROJECT OVERVIEW

1.2. PROBLEM STATEMENT

1.3. APPROACH

1.3.1. DATASET COLLECTION

1.3.2. IMAGE PROCESSING

1.3.3. MODEL SELECTION

1.3.4. MODEL TRAINING

1.3.5. WEB-APP DEVELOPMENT & CLOUD COMPUTATION

1.3.6. HARWARE DEVELOPMENT

1.3.7. REAL TIME TESTING

1.4. OBJECTIVES

1.5. DELIVERABLES

1.6. ORGANIZATION OF THESIS

# INTRODUCTION

This introduction provides a brief description of the research, starting with a critical examination of limitations of today’s CBC tests. It then specifies the project's goals, limitations, and requirements, as well as the deliverables that are intended to be generated. Finally, the chapter finishes with a thesis organizational framework.

## 1.1. Project Overview:

This research focuses on developing a CBC analysis kit using image processing of microscopic blood cell images and using machine learning algorithms to generate results. For the hardware and software components, Python and Google Collab were used for model development and image processing, respectively, while Raspberry Pi 5 was used for integration.
In this experiment, the pictures used were taken utilizing a camera with an attached compound microscope, and the blood staining was placed under the microscope. Now, image processing techniques will be implemented on my images on goggle collab for sharpening and filtering using python. These images were then employed to train our model which successfully used them with the results testifying their high degree of precision for each cell at the testing phase of our research.

A trial was conducted during the test phase and laboratory CBC results were use as the control to ascertain accuracy and performance. The comparison uncovered that the study's outcomes were of the highest accuracy, efficiency, and in terms of utility.

Briefly, prominent is the job of making this ensemble for CBC analysis Fast into a reality. Python and hardware- and software-constructed tools helped to reduce the amount of time needed for integration, while the Raspberry Pi 5 did the same.

## 1.2. Problem Statement:

In a diagnostic setting, provider will often run a hematological test, known as complete blood count (CBC), for an overall health assessment and detecting different hematological (blood) disorders. The manual analysis process, in which one has to scan the pictures by the help of vision, is very time consuming and human errors are something one might find there. Due to the recent high demand for the shorter diagnosis and accurate sampling, it is a must to improve CBC analysis procedures.

Despite being diagnostically important, there are several drawbacks associated with CBC analysis in clinical settings:Despite being diagnostically important, there are several drawbacks associated with CBC analysis in clinical settings:

**Subjectivity and Variability**: This gets really out of hand when the results of CBC tests were interpreted manually, thus, the biases in test results emerge across laboratories and technicians.

**Labor Intensive Process**: Manual microscopic examination of cells to count and classify blood cells often takes too much time which partly results in delay of treatment of a patient, get diagnostic and treatment reports.

**Risk of Errors**: Such mistakes may appear during the manual examination of the mistakes such as the probable miscount or faulty classification. These mistakes in the microscopy process lead to a wrong decision about treatment and diagnosing the disease as well.

**Limited Scalability:** The trend manifested in significant increase of performed CBC tests per day is obviously prolonged by the use of manual examination methods which are no longer able to cope with the pressure, forming bottlenecks in many clinical laboratories.

The problem of failures, inconsistencies and delays due to radiologists' efforts can be addressed by the research through developing an automated CBC analysis system that uses advanced image processing and machine learning techniques. Using computer vision and artificial intelligence, the proposed system attempts to:Using computer vision and artificial intelligence, the proposed system attempts to:

1) Automate the segmentation and electrical-type cell sorting by using microscopic pictures.
2) Increase the reliability and repeatability of the parameters obtained from CBC analysis results.

3) Shorten the time of diagnosis and create lab efficiency across the board order to create, while adding to, efficiently performed lab.
4) Humans who analyze data manually are prone to different types of errors.
5) Perform scalability to test CBC to large number of units in time. Also precision should be held constant.

In regard to testing and comparison with the normal laboratory reports of CBC the accuracy of the research set up and efficacy will be confirmed. The project objective is to design an easy-to-use and quick alternative method to general about CBC examination with not falling into the catch of human faults and lab delay.

This platform for development can enhance local services, such as medical research and imaging. Through such a programme, there will be hopefully progress in these disciplines, which will subsequently present more effective results in the testing processes.

## 1.3. Approach

The project entitled "Complete Blood Count using Image Processing and Machine Learning Algorithms" aimedto design a analysis kit for CBC using AI technology. The approach involved several steps, including:

### 1.3.1. Dataset Collection:

The first step was the collection of blood smear images for training and testing of our AI model for accurate and efficient results.

### 1.3.2. Image Processing:

The second step is the application of different image enhancement techniques on the training dataset using python for filtering of noise and edge sharpening of blood cells for accurate detection.

### 1.3.3. Model Selection:

The next step was the selection of a model for object detection, identifying and locating objects within the image.

### 1.3.4. **Model Training**:

Next, we developed a python codebase for training our model.

### 1.3.5. Web-App Development and Cloud Computation:

Next step is the development of a web application for real time testing and results generation and integration through cloud computation as well.

### 1.3.6. Hardware Development:

Raspberry Pi 5 computing device along with a compound microscope will be part of the hardware development.

#### 1.3.7. Real Time Testing:

Finally, real time testing will be performed, and results will be compared to verify the accuracy and efficiency of our model.

Using AI technology and implementing the techniques of image processing and machine learning, we were able to construct an efficient and accurate CBC analyzer. The analysis kit might be utilized in a variety of real-time processing applications, including medical imaging and medical diagnostics.

## 1.4. Objectives:

The purpose of this project is to provide a hardware-based solution for efficient and accurate CBC analysis by accomplishing the following objectives:

- Create powerful image processing algorithms that accurately segment and extract features from CBC images.
- Use machine learning models based on extracted features to categorize and quantify various blood cell types.
- Evaluate the built system's performance using real-world CBC datasets and compare it to manual analysis findings.
- Integrate the automated CBC analysis system with existing laboratory operations to ensure smooth deployment and user accessibility.

## 1.5. Deliverables:

This project's deliverables include:

- A functioning model/prototype for efficient and accurate CBC analysis.
- Python codebase for image processing, model training and real time testing.
- Implementation of the model using Raspberry Pi 5.
- Web application for real time testing and result generation.
- Documentation of the design, implementation, and testing processes for the system.

## 1.6. Organization of Thesis:

This thesis is structured as follows:

- **Chapter 1: Introduction**

  This chapter offers a summary of the project, including its goals and problem description. It provides an overview of the limitations of today's methods of CBC analysis and provides an emphasis on the objectives and deployment of the project.

- **Chapter 2: Literature Review**

  This chapter reviews the existing methodologies on CBC analysis available in laboratories. It also gives a brief review of the image processing techniques used in the project and also about the model used for object detection.

- **Chapter 3: Integrated Development: Software, Hardware & Web Application**

  This chapter discusses the system's overall design, including the hardware and software components. It addresses design concerns such as system architecture, block diagrams, and component functioning. This chapter also provides implementation details, such as python coding for image processing and model training, results generation, integrationof hardware and software components and the development of web application.

- **Chapter 4: Calculations and Result Analysis**

  This chapter assesses the system's performance using several measures such as processing time,resource use, and power consumption. It also includes method of calculation and generation of results by comparison with CBC reports.

- **Chapter 5: Conclusion**:

  This chapter outlines the project's research activity and emphasizes the main accomplishments.It finishes with a description of the project's contributions to real time applications including medical imaging and medical diagnostics.

- **Appendices**: These provide more information regarding the system design, model and web-app codebase, and results. They also provide a list of references that were utilized in the project.

# CHAPTER # 02

# LITERATURE REVIEW

2.1. BACKGROUND

2.2. INTRODUCTION TO COMPLETE BLOOD COUNT (CBC)

2.2.1. BLOOD COMPOSITION

2.3. RELATED WORK

2.4. IMAGE PROCESSING

2.4.1. IMAGE ENHANCEMENT

2.4.2. GREY-SCALE CONVERSION

2.4.3. HISTOGRAM EQUALIZATION

2.4.4. GAUSSIAN SMOOTHENING FILTER

2.4.5. EDGE SHARPENING FILTER

2.4.6. IMPACT OF IMAGE PROCESSING ON OUTPUT

# Literature Review

Complete blood cell (CBC) counting has played a vital role in general medical examination. Common approaches, such as traditional manual counting and automated analyzers, were heavily influenced by the operation of medical professionals. In recent years, computer-aided object detection using deep learning algorithms has been successfully applied in many different tasks. In this paper, we propose a CNN based architecture to accurately detect and count blood cells on blood smear images. A public blood smear images dataset is used for the performance evaluation of our developed model. It is not uncommon that blood smear images are in low resolution, and blood cells on them are blurry and overlapping. The original images were preprocessed, including image segmentation, sharpening, and blurring. The experiment results show that our models can recognize blood cells accurately and precisely and calculating count in normal ranges (1).

## 2.1. Background

This chapter introduces the methods of CBC analysis, image processing and real time object detection. It also contains a survey of the literature on the relevant topics to the project. To support and enhance the goal of this work, each of the subjects will be covered briefly. The goal of this chapter is to provide an overview of the project hardware and software components as well as current research in this field.

## 2.2. Introduction to Complete Blood Count ( CBC )

Blood cells carry important information that can be used to represent a person's current state of health. The identification of different types of blood cells in a timely and precise manner is essential to cutting the infection risks that people face on a daily basis (2). A Complete Blood Count is the most common test which is performed in laboratories by healthcare professionals. It is the most basic of tests which is performed providing basic information related to a person's health and blood composition. A CBC measure various components of blood including red blood cells, white blood cells, platelets, and other various parameters such as hemoglobin, hematocrit, and mean corpuscular volume etc. A CBC is a vital tool for diagnosis of various medical conditions, monitoring response to treatments and assessment of overall health condition. It is essential to interpret CBC results in conjunction to clinical information to make accurate diagnosis and treatment decisions.

### 2.2.1. Blood Composition:

Lets look at the basic components of blood in detail:

**Red Blood Cells (RBCs):**

- RBCs are responsible for carrying oxygen to lungs and various tissues and responsible for the removal of carbon dioxide.
- The CBC measure the number of RBCs per volume of blood (expressed as millions of cells per microliter)
- Anomalies in amount of RBC's may point to disorders such as anemia (decreased RBC count) and polycythemia (great number or RBC).

**White Blood Cells (WBCs):**

- WBCs are the most important components of the immune system which kill microbes and deal a lot with diseases and other threats to the body.
- The central vectoring cell counts (CBC) quantifies the secretary (WBCs) number per the blood unit (volume) (expressed as thousands of cells per microliter).
- Distortions in the WBC count point out systemic disease, infections, inflammation or an autoimmune disorders or, much commonly, leukemia (cancer of blood cells).

**Platelets:**

- Platelets are small cell fragments that play a key role in blood clotting. They help prevent bleeding by forming clots at injury site.
- The CBC measures the number of platelets per volume of blood (expressed as thousands of cells per microliter).
- Abnormalities in platelets count can indicate various bleeding disorders (thrombocytopenia if low count) and clotting disorders (thrombocytosis if high count).

## 2.3. RELATED WORKS

**1) Early History of CBC:**

Since early times, scientists have been performing microscopic investigations on blood. They observed and analyze blood cell while continuing to improve microscopic technologies (1) Since early times, scientists have been performing microscopic investigations on blood. They observed and analyze blood cell while continuing to improve microscopic technologies (3). Research on automated blood cell counting started in the 20$^{th}$ century. The Coulter counter, which used the blood's conductivity as a reference to count and estimate RBC sizes, was introduced in the 1950s (4). Cells were pushed via an aperture to perform counting of RBCs. The Technicon SMA 4A–7A could analyze numerous cells at once, while the Coulter counter was adapted and used for WBC and RBC counting (5). In 1970, Technicon Hemalog-8 automated platelet counts were introduced, and in 1980, Coulter's S Plus series analyzers incorporated this technology (6).

**2) Image Processing based methods:**

One year ago, Madhloom and co-authors proposed an approach to automate WBC detection and classification based on different image processing techniques that identified the nucleus of targeted cells (7). In this regard, they converted the images to grayscale and eliminated the darkest region corresponding to the nucleus of white blood cells. Further, localization of the nucleus was conducted using automatic contrast stretching, histogram equalization as well as image arithmetic before segmentation operation was done using automatic global threshold. Subsequently, through passing a minimum filter together with an automatic threshold; leukocyte types would be recognized by their nucleuses. At last, the suggested model's accuracy ranged from 85% to 98%.

Moreover, in 2018 Acharya and Kumar presented a method for RB counting and identification using imaging processing (8). To separate out RBCs from granulometric analysis extracted WBCs were used while counting them at K-medoids algorithm. The labelling algorithm along with circular Hough transform gave RBCs. Comparatively, results demonstrated that counting was better performed by Circular Hough Transform than labeling algorithm because it could tell if there were no changes in RBC count when compared with normal range.

Kaur et al introduced an automated platelet counter for microscopic blood platelet counting in 2016 which employed a circular Hough transform.

**3) Machine Learning based methods:**

With the rapid growth of machine learning algorithms and computational resources, adopting machine learning for medical diagnostics through image analysis has become an effective and efficient approach for information. Computer aided diagnosis, medical images diseases detection and image guided therapy are the various medical researches where machine learning techniques were implemented (7).

Habibzadeh et al. later proposed a WBC classification system that used pre-trained models of ResNet and Inception (8) in 2018. The images were augmented, preprocessed by color distortion, and fed into ImageNet pre-trained ResNet and Inception networks to abstract useful features for recognition. ResNet V1 50 gained the best detection rate with an average of 100%, while ResNet V1 152 and ResNet 101 achieved 99.84% and 99.46%, respectively.

## 2.4. Image Processing

The modification of pictures for their improvement of quality or extraction of valuable information, this is what forms image processing. Filtering, member grouping, and getting the essential characteristics may also be helpful in resizing a photo and making it suitable for a defined purpose.

### 2.4.1. Image Enhancement:

The image enhancement is a component of image processing which works upon the enhancement of apparent quality of the image. This could be done multiple ways like increasing contrast, improves sharpness of the image, eliminating noise or blurring and other ways could make the image more visually pleasing or easier to understand the image. The goal of image enhancement is to either create images more distinguishable in interpretation or to have better quality images while remaining loyal to the data set.

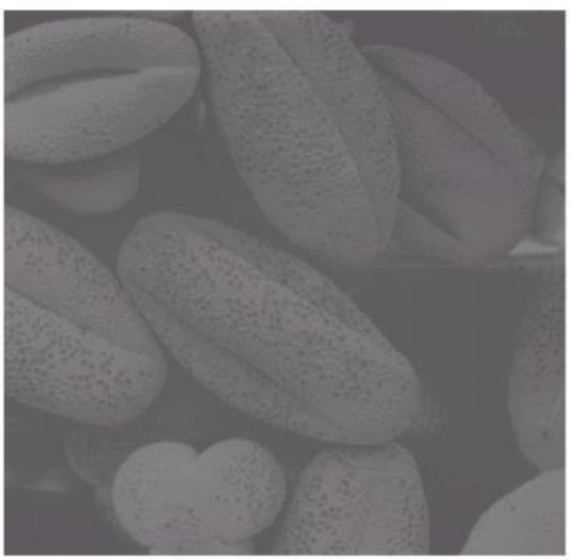

*Figure 3 Original Image*

*Figure 4 Enhanced Image*

### 2.4.2. Grey Scale Conversion:

Greyscale conversion is a specific refinement process in the image processing that is used to covert colorful images to shades of gray for the making the monochrome image. Thus, the method becomes useful for such applications such as image processing, computer vision and image simplification when suffering from computational complexity.

**Reduction in Complexity:** Gray-scale images constitute fewer channels of information than color images having three channels. As a result, the computing power, and processing storage space are little less than that of the color images.

**Image Processing**: For instance, gray scale images give various methods of processing an image a definite advantage which include edge detection, segmentation, feature extraction.

**Visual Consistency**: A color detail may be unnecessarily and make the other more important structural and textual information unuseful in some applications for example some texts recognition or medical imaging.

Some medical applications of grey scale conversion are as follows:

**Medical imaging:** Assists in the analysis of different scans, such as X-rays, in which color does not significantly add to the information.

**Video Surveillance:** Facilitates faster processing by reducing the amount of data handled by the system.

**Machine Learning:** Reduces the computational load during model training by simplifying the data without sacrificing important information.

Grey Scale conversion is a fundamental part of image analysis and computer vision as it aids in simplification of data while retaining the much relevant data for information analysis.

### 2.4.3 Histogram Equalization

Histogram Equalization is a common image processing method for improving image contrast. The approach works by dispersing the pixel values of an image to create a more uniform histogram, which results in a more visually appealing image. In histogram equalization, the transformation function is a monotonically growing function that translates the input pixel values to the output values. Histogram equalization is a global image processing method that modifies the pixel values of a whole image to get the desired outcome. Neighborhood processing approaches, on the other hand, include changing the pixel values of a specific pixel depending on the values of its nearby pixels.

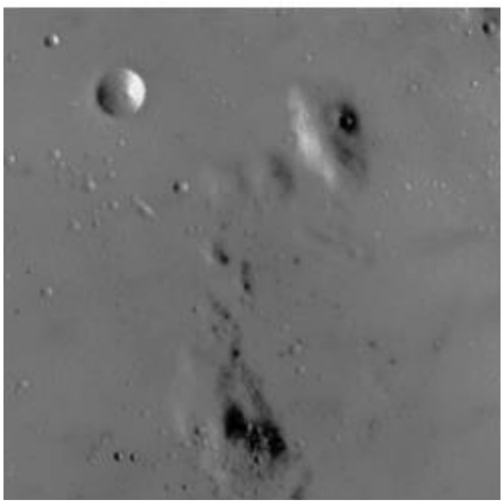

*Figure 5 Original Image*

*Figure 6 Histogram Equalized*

Several other types of the basic histogram equalization algorithm also exist and include contrast limited adaptive histogram equalization (CLAHE) and adaptive histogram equalization (AHE). Therefore, by placing an upper limit for the highest pixel value within each histogram bin, it refines simple histogram equalization methods to minimize noise over-amplification in areas of low image contrast. As a separate technique, AHE calculates histograms for small regions of the image causing an adaptive enhancement in contrast suitable for pictures of non-uniform illumination. There are also many forms of this fundamental histogram equalization strategy like CLAHE and AHE. CLAHE ameliorates plain old vanilla histographic equalization by constraining the maximum pixel value in each histogram bin resulting in minimum overamplification in noise generated components with poor quality. In contrast, AHE uses a different approach where it computes histograms on small sections of images hence giving rise to adaptive brightness improvement particularly suited to photographs taken under nonhomogeneous lighting conditions.

### 2.4.4 Gaussian Smoothening Filter:

The Gaussian smoothening filter is an image processing technique used to reduce the noise and detail in an image. It is widely used in the fields of computer vision and graphics.

**How Gaussian Smoothening Works?**

1) **Gaussian Function:**

- The function serves as the base of the Gaussian smoothing filter

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e \frac{-x^2}{2\sigma^2}$$

- This function is used in two dimensions to produce a Gaussian matrix, often known as a kernel, that is then applied to images in the context of image processing. The function for a 2D Gaussian is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e \frac{-(x^2 + y^2)}{2\sigma^2}$$

- Here, σ (sigma) represents the standard deviation pf distribution. The value of σ determines the amount of blurring: higher σ results in more blurring.

2) **Creating Gaussian Kernal:**

- The Gaussian kernel approaches the square shape (3x3, 5x5, etc. ) usually. The amount of smoothing with the bigger kernel is larger than that with the small one. Smoothness result since a larger kernel takes more accounts resulting in better smoothing.
- The Gaussian function is used to compute every kernelized term, so that a solution can be derived. The data is normalized, so that the sum of the clemet count of the kernel is one.

3) **<u>Application of the Kernal:</u>**

- To achieve this result, a convolution occur of the Gaussian kernel applied in every pixel in the image. A weighted sum of six closest pixels is supplied for it, where the weights are calculated based on a Gaussian kernel.
- And then the value of every pixel is specified by a weighted mean which involves the current pixel and neighbor pixels where the nearer ones are more influential than those that are relatively far off.

**Functions of Gaussian Smoothing:**

1) **<u>Noise Reduction:</u>**

Pandis's task of smoothing Gaussians diminishes dissimilarity of pixels, comprising of the noise, by averaging pixels, separated from the abutting neighbors. This becomes very important when the pictures are required to undergo some advanced processing for instance edge detection. This means that any noise produced gives a wrong or misleading impression.

2) **<u>Detail Blurring:</u>**

Gaussian smoothing harshly decreases the image's high-frequency content, thus smoothing it out and as a result markedly diminishing the image quality. This emphasizes the aspects where machine learning models can perform better, in this case more appropriately when trying to obtain general patterns instead of smaller details.

3) **Edge Smoothing:**

One way where it is often used in methods is when a heavy gradient is desired, but not hefty borderlines, which is common in portrait modes with a background blurring option on different pictures.

*Figure 7 Original Image*

*Figure 8 Gaussian Smooth Image*

### 2.4.5 Edge Sharpening Filter:

The boundaryPixel sharpening filter consists of an image processing technique to apprehend and to delineate the edge of objects thus it highlights the fragmentation and contrast of the image.

**How Edge Sharpening Works?**

1) **Understanding Edges in Images**:

Edge Image is the border line between different regions when suddenly there's something in the image which shifts the hue stroke, or intensity. In most cases, the transforms results are aligned with texture changes, the picture borders move, or other important points in the picture are sharpened.

2) **High-Pass Filtering:**

- High-pass filtering is a method that is frequently used in edge sharpening. This technique highlights high frequency regions in a picture, which are effectively the edges and correlate to abrupt changes in intensity.

- A low-pass (blurred) copy of the image can be subtracted from the original to create a high-pass filter. As an alternative, certain kernels can be sharpened directly from the container.

**3) <u>Sharpening Kernels:</u>**

- Convolution combined with an edge-highlighting kernel is a popular method. An example of a common sharpening kernel could be as follows:

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$$

- The way this kernel functions is by giving the central pixel a higher positive weight and the surrounding pixels a negative weight. This enhances edges by magnifying the difference between a pixel and its neighbors when convolved with the image data.

**4) <u>Convolution Process:</u>**

- Using convolution, the sharpening filter is applied to the image by having the kernel "slide" over each pixel. The original pixel value for each pixel is replaced by the sum of the products of the overlapping kernel values and the image pixel values.
- Through this procedure, the image seems crisper, and the edges are effectively made more prominent.

**Functions of Edge Sharpening:**

**1) <u>Improving Image Details:</u>**

This sharpens edges in the image by raising the contrast between neighboring pixels that make up the edges. As a result, minute details are easier to see, and the picture appears less hazy or fuzzy.

**2) <u>Assisting with Feature recognition:</u>**

In computer vision, finer edges allow for more precise object recognition since edge detection algorithms (like Canny) can recognize and follow the outlines of objects more readily.

3) **Increasing Text Readability:**

Sharpening aids in increasing the edges of text characters during OCR (Optical Character Recognition) or in scanned documents, which increases recognition accuracy.

4) **Aesthetic Enhancements:**

Sharpening filters are frequently used in digital photography and video to improve the sharpness and clarity of images, particularly after resizing or when the original image is slightly out of focus.

*Figure 9 Original Image*

*Figure 10 Edge Sharpened Image*

### 2.4.6 Impact of Image Processing on output

A large variety of methods are included in image processing that have the power to greatly modify and enhance image output. These methods are essential in many domains, including security, driverless cars, digital photography, and medical imaging, among others. Here is a thorough examination of the effects of image processing on the result:

**Enhancement of Image Quality:**

**Noise Reduction:** Using Gaussian smoothing and other image processing methods can reduce the random fluctuations in color or brightness of images that are often known as noise. This can help to reveal true features of an image.

**Contrast Enhancement:** Methods such as contrast stretching and histogram equalization increase the appeal and clarity of images by extending the intensity range of the images. This brings about better interpretation and visualization.

**Sharpness Enhancement:** In addition to making objects in a photo look sharper, edge sharpening filters also give the impression that they are more focused and detailed.

**1) Feature Extraction:**

For applications like pattern recognition, computer vision, machine learning etc., being able to extract meaningful features from an image is very important. Efficient image processing algorithms can do this.

Using techniques such as edge detection, segmentation, texture analysis etc., different objects/ patterns in an image can be identified and isolated. These systems are essential for automated systems such as facial recognition and automatic inspection systems.

**2) Color processing:**

Using color enhancement and correction, pictures can become more aesthetically beautiful or realistic. This is especially crucial in sectors where spectator perception and happiness are strongly impacted by visual attractiveness, such as film and photography. False color imaging can be used in scientific applications to show photographs in colors other than those in the visible spectrum, allowing features to be seen that would be impossible to see otherwise.

**3)Compression in Transmission and Storage:**

Compression techniques are part of image processing and help to reduce the size of files, which is important for transmission as well as storage. The use of effective compression methods ensures that image quality is preserved as effectively as possible while saving the least amount of storage capacity. In areas such as the creation of online content, where download speed and bandwidth are essential components, this is particularly important.

**4)Restoration:**

The image restoration process may address these issues. Restoration aids in as nearly recreating the original image as feasible. This is particularly important in fields like astronomy, where images taken from space are often damaged to some extent.

**5)Medical Imaging:**

In order to diagnose, it is essential that the medical imaging data are clear and precise; processing methods can have a major impact on this. Methods such as Xray enhancement, MRI segmentation, and 3D reconstruction of slices can only be performed with advanced image processing techniques. Improved photos offer more trustworthy data for illness diagnosis, treatment planning, and patient progress tracking.

# CHAPTER # 03

# DESIGN AND DEVELOPMENT

3.1. DESIGN FLOW

3.2. HARDWARE AND SOFTWARE PART

3.3. SOFTWARE PART

3.3.1. MODEL DEVELOPMENT

3.3.2. WEB APPLICATION DEVELOPMENT

3.4. YOU ONLY LOOK ONCE (YOLO) v8 NANO

3.5. HARDWARE DESIGINING USING RASPBERRY PI 5

3.6. SOFTWARE / HARDWARE INTEGRATION

3.7. HARDWARE REQUIREMENT

3.8. SOFTWARE REQUIREMENT

# Design and Development

With the schematics and models that come with every part of our project, it is a matter of hard-work, perseverance and initiative. Next, high-quality blood smear images are taken, and after that the images are enhanced through the application of various image enhancement techniques. The purpose of these steps is to improve the quality of such images in terms of visibility and contrast. These methods include histogram equalization for correcting the distribution of brightness, Gaussian filtering process for noise reduction and smoothing of an image, edge sharpening is also included in this process for sharpening cell boundaries in a blood smear thus enabling single cells to be more conspicuous and simpler to analyze.

Upon preprocessing, the CNN which is shortened version of You Only Look Once (YOLO) real time object detection system, is exploited with YOLO v8 nano whose aim is a device with a super speed detection mechanism with very little computational power. In particular, this combination is used in identifying and classifying different types of blood cells: also transport oxygen to different parts of the body through red blood cells (RBCs), white blood cells (WBCs), and platelets. Although the CNN is capable of extracting and dealing with hierarchical features in images, the YOLO v8 nano, on the other hand, is a substitution that permits the determination of the positioning of each cell type in blood smear images to be done instantaneously.

By identifying and indicating every cell, the system then avails the number of the types of cells in the image. The count is significant for doing complete blood count (CBC) which is largely utilized as a medical diagnostic test. These respective counts in turn are used in arriving at findings from which the blood health is assessed and issues like anemia, infection and abnormal blood clotting among others can be diagnosed. The union of such technologies in one digitized work pipeline has made a significant progress in the passage of image processing and AI solutions in medical diagnostics, ensuring on-the-spot accuracy and efficiency of the whole blood testing process.

## 3.1. Design Flow:

Let's break down each task represented in the flow chart:

1. **Raw Data**

   The first point on the basis of which the project is implemented is gathering the unprocessed data strictly needed for the project. Your study demands the gathering of several high-quality slides blood images from an applicable medical source which is the context of your project. The images' resolutions and quality should be good enough to avoid less precise processing and the analysis of images.

2. **Splitting Data**

   To start with data is the first and most important step, data has to be divided into training, a validation, and test sets. Such task is vital for machine learning projects to confirm the affirmative and then validation on separate sets of data again and final testing on another unseen set of data to get the approximate performance at the end of the project.

3. **Image Pre-processing**

   During this phase, we apply the preprocessing of the blood smear images to increase its image quality through techniques like histogram equalization to adjust image contrast, Gaussian filtering to remove the noise, and edge sharpening to make the cell boundaries more visible.

4. **Model Selection & Customization**

   It is important to make a choice of a good machine learning model. For your task, it is about using a classification structure for images known that CNN and customizing YOLO v8 nano for real-time object detecting. First of all, model will be configured to be capable of processing various classes of blood cells: red blood cells, white blood cells, and platelets just to mention a few.

5. **Model Training**

   The model chosen accompanies by the fitment plays the role of being trained by the training dataset set. This entails scrubbing the preprocessed pictures and putting it into the model such that it can acquire from the labeled examples. The desired result is that the machine model correctly recognizes and groups any unprocessed blood cells in a new image.

6. **Cell Counting & Classification**

After the model is trained and validated, it is a point in which it is used for counting and classifying blood cells in images. CNN and YOLO v8 nano concurrently comprehend this complex task by recognizing, identifying, and distinguishing between cells, counting blood cells for each type.

7. **GUI**

A GUI (Graphical User Interface) is developed to facilitate an intuitive interaction between the end-user and the system. The GUI is going to be for data inputting and outputing as well as providing the option to upload the new images for analysis, based on which it generates the results of the blood count.

8. **Implementation on Raspberry Pi 5**

The trained model and GUI are going to be in the Raspberry Pi 5 based device. The next stage in the development is to optimize the software code to run without any discrepancy in the Raspberry Pi's hardware by using only less amount of processing power, thereby executing the diagnostic command and producing the blood count results quicker and in real-time.

9. **Testing & Validation**

In the stage of validation, the system is fully tested using validation and test data set in an effort to guarantee that it does everything correctly. This step is of utmost importance to detect any flaws or bottlenecks by bringing up in real world environment.

You have been reading essays, a perfect way to improve your writing skills and explore different topics.

10. **<u>Data & Result Storage on Cloud</u>**

Lastly, the blood test results made on behalf of the patient as well as test images are synced to the cloud platform. It is possible to achieve this through the provision of easy access, scalability, and the integration with any other program, or database, and this is the best option of managing the data.

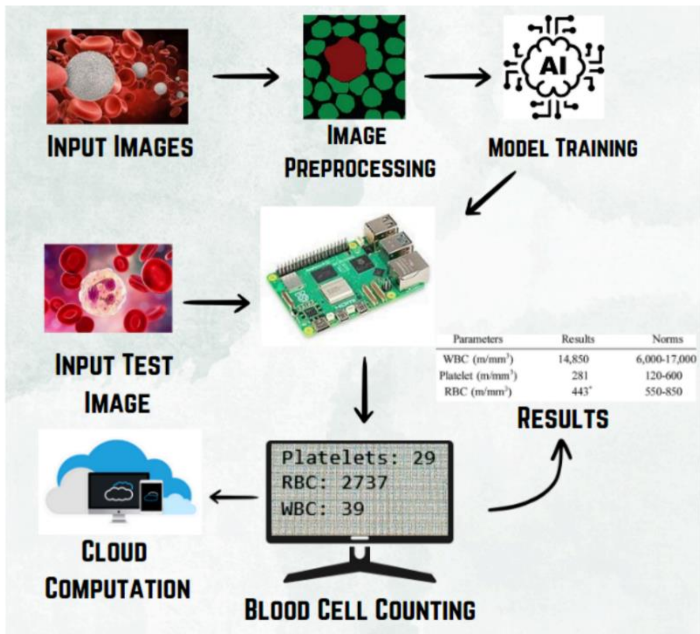


*Figure 11 Design Flow*

## 3.2. Hardware and Software part:

In the element of design a software and hardware are both important parts in order to make the project functional and efficient. The source of the software employed is Google Colab that proved to be the main development environment. Its key strength is the ability to perform CNNs and YOLO v8 nano models trainings that are focused on object detection and classification operation. This infrastructure is beneficial due to the power of Google Colab that hooks up to high-power computing resources that can accommodate GPUs and thus the training process is loaded up very fast. At first, image processing approaches such as histogram equalization, Gaussian filtering, and edge sharpening are used to improve quality of blood smear images before they are fed into the object detection and classification systems which will aid in the accurate diagnosing of blood diseases.

Moreover, the additional implementation has been completed for the user-friendly web application that will provide an easy-to-use platform for the user to interact with the system. The web application designed by us helps the users upload blood smear images and go through test tagging and result viewing all at one site. The Arduino Maker Zero is connect seamlessly to the Raspberry Pi 5 that is a local server and control unit as well. Raspberry and each model so that they can be deployed and used in an operation. It processes the enhanced images in real-time and displays the results through the web application interface.

Meanwhile, cloud computing is indistinguishably important because it furnishes the ability to scale storage and processing on demand. Data collected on both the image side and object recognition are stored to a cloud which provides safekeeping and ease of access. With this approach where local and cloud resources are both implemented, the speed and scalability of the project are optimized. Thus, the solution will excel in any medical and research application that it is used.

## 3.3. Software part:

One part of the project software is the modelling through Google Collab using the CNN as well as the YOLO V8 nano and the development of web application system.

### 3.3.1. Model Development

1. Google Collaboratory:

Google Colaboratory (Colab) is a free Jupyter Notebook environment provided by Google. It al lows. users to run Python code in the cloud without needing to install libraries or manage hard ware hardware. This

makes it ideal for intensive tasks like training deep learning models, which can be done especia lly quickly on GPUs.

**YOLOv8 Nano:**

You use YOLOv8 Nano, an evolution of Ultralytics' popular YOLO (one-

look) sample detector. YOLOv8 is known for its speed and accuracy. The Nano version is a lig htweight, highperformance model designed for limited spaces. This makes it a good choice for exporting equipment that requires less work. The resulting datasets:

(https://github.com/MahmudulAlam/Complete-Blood-Cell-Count-Dataset)

We have examined the data model, which contains visuals related to our work. The exact detail s depend on what you want. In our case, the data will include images of blood cells listed in thr ee categories: platelets, RBCs (red blood cells), and WBCs (white blood cells). The legend usu ally defines a box surrounding each cell in the chart.

2. Resizing the image:

We resized the images in the dataset before training. This is an application of deep learning to ensure that all images are the same size as input to the model. It can improve the safety and coordination of training.

**Grayscale conversion:**

We chose to train the model of the grayscale image. While color data can be useful for some tasks, gray scale images are sometimes sufficient. This reduces the size of the input data, has the ability to train quickly and reduce memory usage. Additionally, grayscale images may be useful for tasks where color may not be distinctive (e.g., blood cell morphology). Stains are dyes used to distinguish different types of cells by their color. However, changes in the staining process may cause color changes in the appearance of the blood. By converting the image to grayscale, you remove color information and force the model to rely on other visual factors (such as cell shape and texture) for detection. This may make the model more robust to changes in the blood sample staining process. Normalization process. This strategy aims to transform the image into a color space where different spots are similar, allowing the model to focus on morphological features.

3. Image preprocessing data analysis:

Image preprocessing is an important step in deep learning. It may include various techniques such as resizing, normalizing, cropping, and enhancing. Research on the effectiveness of different prioritization methods often depends on specific data and tasks. Some studies have shown that grayscale conversion is very effective for certain research purposes, especially when dealing with limited data or limited reso urces.

4. ADAMW Optimizer:

We learned to use the ADAMW (Adaptive Torque Estimation with Weight Decay) optimizer. ADAMW is a variant of the ADAM optimizer that fixes some limitations. It contains a heavy weight th at helps prevent overload and increases stability during exercise. Popular optimizers for deep learning s uch as SGD (Stochastic Gradient Descent), RMSprop (Root Mean Square Propeller) and Adam. Research shows that ADAMW has advantages over other optimization methods, such as faster integrati on and improved expandability compared the performance of various optimizers for object detection tasks and found that ADAMW achieved faster convergence and better accuracy compared to SGD and Adam. This suggests that ADAMW may be a good choice for training YOLOv8 Nano for blood analysis.

**Hyperparameter setting:**

Training value (start 0.001, end 0.0001): The training plan you use starts at 0.001 and gradually decreases to 0.0001 during training. This is a method that helps the model first learn coarse features and then refine them during training. Difficult patterns. However, it is important to monitor performance to avoid overfitting. The early stopping technique can be used to stop training if performance begins to deteriorate. A correct diagnosis. This is an essential starting point for blood tests where accuracy is important.

5. Model validation in Google Colab:

The specific functionality of model validation depends on the deep learning library you use to train YOLOv8 Nano. But in most cases, there will be a method provided by the model or training model that allows you to evaluate the effectiveness of the model on separate data. This evidence is not used during the study, but helps evaluate how well the model fits unseen data. Metrics such as mean precision (mAP) or return on valid data can be calculated. These measurements provide insight into the exact diagnosis of the model and the ability to find all related products. After testing our developed model, an overall accuracy of 94.8 % was achieved as a result of model validation.

6. Dual Model Approach:

In our project, two different models are being used to optimize detection and counting of blood cells by addressing specific challenges during the early testing stages. Initially, one model was used that could accurately predict White Blood Cells (WBCs). However, it was observed that this model which though effective in WBC identification often resulted in multiple detections for a single WBC due to the fact that these models were sensitive towards different shapes and sizes of these cells especially when using image processing techniques based on edge detection.

We resolved this by adding an additional model to increase the overall accuracy of our system. Now, the first model uses color images and employs sophisticated image processing techniques to discriminate various color characteristics thus making them more unique among other structures found in WBCs. This helps to solve over-detection problem significantly as the full color spectrum information can be utilized by the model which makes it especially useful for precise differentiation of WBC from other constituents present in blood.

Besides, we added another sub-model dedicated to red blood cells (RBC) and platelet recognition. A grayscale conversion is applied on these images simplifying them thereby concentrating only upon their shape characteristics while ignoring texture.

### 3.3.2. Web Application Development

**Introduction:**

The arrival of computer vision technology has changed many fields, such as health care, since it helps in providing robust means for analysis and Diagnosis. One of those is the identification and counting of blood cells which is crucial in medical diagnostics and research. The manual counting method using a microscope was commonly used to count blood cells but this technique consumed a lot of time and was prone to errors due to the human factors. Recently though, deep learning-based object detection models have emerged as a possible solution towards automating this process. This thesis describes a web application that I designed, implemented and evaluated with an aim of automating detection and enumeration of blood cells using YOLOv8 object detection model. It uses HTML, CSS and Flask framework together to provide an interface that is simple enough for users to upload images, process them through the YOLOv8 model, displaying results afterwards. By carefully considering each technology employed in building the app and also reviewing its functionalities well; this thesis therefore seeks to show how effective using deep learning is in analyzing blood cell on web-based platform. Also, as a docker image on some cloud platforms.

**System Architecture:**

The architecture system section is technical details of the web application. It outlines various components and gives details on how they timely behave with each other to create frontend and the backend. The HTML is one to mark out the user interface elements that represent the one changing the appearance of a website, while CSS is one for styling that provides a website with its form and usability. In Flask framework, this backend functionality is what becomes a bridge between front-end and back-end, and it serves to handle requests, route through pages, and integrate YOLOv8 model for the blood cell detection. This is possible by virtue of frontend and backend communicating with each other in real time, thus enabling people to load images that are later on might being processed by YOLOv8 model. After the analysis finished a green dots will mark the existence of cells on the image so that the user can immediately notice it.

**HTML:**

The standard modeling system of creating web pages and web applications is translated into Hyper Text Markup Language. It has an approach for managing the content and for specifying the structure and presentation of web pages. The form of the content, or rather the structure and syntax, allow web browsers to treat them according to create pages users can easily utilize.

**Key Features of HTML:**

**1)** Semantic Elements

**2)** Hyperlinks and Anchors

**3)** Image Embedding

**4)** Forms and Input Elements

**5)** Table Structure

**6)** Metadata and Document Structure

**Advanced Features and Best Practices:**

1) Accessibility

2) Responsive Design

3) HTML5 features

4) SEO Optimization

**CSS:**

CSS (Cascading Style Sheets) serves as a style sheets language to specify the presentation and layouts of HTML documents. It offers a way of styling HTML elements of which font color, typography, alignment, layout, and animation are among the elements that can be controlled. CSS, while being a tool commonly used for styling, allows developers to produce websites with graphically appealing and responsive designs that continue to convince readers' attention across browsers and devices.

**Key Features of CSS:**

1. Selectors and Declarations:
2. Cascading and Specificity
3. Box Model:

4. Flexbox and Grid Layouts
5. Media Queries
6. Transitions and Animations

**Advanced Features and Best Practices:**

1. Preprocessors
2. Vendor Prefixes and Polyfills
3. Performance Optimization
4. Modular CSS Architectures

**Flask:**

Flask is a Python web framework that allows creating web applications with its lightweight and flexible stack of libraries and framework tools. It is famous for its straightforwardness, the simplicity of use and the capability of extension, which makes these employers often chosen for projects of various sizes and complexity. Flask, on the other side, embraces the WSGI (Web Server Gateway Interface) standard, making it is possible to use Flask with the Wienkzeug and Gunicorn web servers.

**Key Features of Flask:**

1. Routing
2. Template Rendering
3. HTTP Request Handling
4. Context Management
5. Extension Ecosystem
6. Development Server

**Advanced Features and Best Practices:**

1. Blueprints
2. RESTful API Development
3. Database Integration
4. Security Considerations

**Azure App Services:**

The Azure App Service is a PaaS (platform-as-a-service) artifact developed by Microsoft Azure to help build, deploy, and scale applications and APIs. It makes the tangling of the underlying infrastructure superfluous so that developers can concentrate on the design and delivery of apps without being involved in the choosing of servers and settings, much less maintenance. Azure App Service in addition to the support for various programming languages, frames, and run-time environments, is a good option for all sorts of web application development.

**Key Features of Azure App Services:**

1. Web App Deployment
2. Scalability
3. High Availability
4. Managed Services
5. Integration with Azure Services
6. Development Tools and IDE Integration

**Advanced Features and Best Practices:**

1. Customization and Extensibility
2. Deployment Slots
3. Application Insights
4. Security and Compliance

**Docker:**

Among all, Docker is the one that most developers, engineers, and shipping firms alike to run the applications using containerization technology. It provides a platform-independent and portable solution for building microservices but into standard units called containers or images, which encapsulate those application's entire required dependencies and runtime environments in[only] one software package. Docker aids in deployment process, promotes scalability, raise resource utilization efficiency via encapsulation of programs from the infrastructure.

**Key Features of Docker:**

1. Portability
2. Isolation
3. Efficiency
4. Orchestration
5. Integration

**Use Cases and Applications of Docker:**

1. Application Deployment
2. Microservices Architecture
3. DevOps and CI/CD
4. Hybrid and Multi-cloud Deployments

## 3.4. You Only Look Once (YOLO) v8 Nano

YOLOv8 Nano is a variant of the YOLO (You Only Look Once) family that is built for real-time object detection and upholds the principle of detecting objects in a single inference pass through the entire image, contrasting with other detection systems that might process multiple parts of an image separately or require multiple passes.

### Working of YOLO v8 Nano

The model only needs to make one prediction per grid cell template (and it is just a few pixels per cell), which makes it possible to perform the same operation on large images. For each bounding box, there are predictions about how far it reaches from any intersecting point, its dimensions and how likely something resides in this particular space. Moreover, every cell gets divided into classes bearing in mind that there may be some item inside this box. This helps in both predicting boxes and classes thus making efficiency paramount for localizing and classifying objects by the model during evaluation

### YOLOv8 Nano Architecture

In particular, the YOLOv8 Nano architecture is designed to reduce the computational burden while maintaining an efficient detection capability. This is achieved through careful optimisation of the network structure:

1) **Backbone:** YOLOv8 Nano's backbone is responsible for feature extraction. It uses a series of convolutional layers but with reduced depth and number of channels comparing to larger YOLO models. To obtain features at less computational cost, lightweight layers such as depth-wise separable convolutions are used. In turn, these layers divide the convolution into two parts- a channel-

wise spatial convolution on each input channel, and a pointwise convolution that mixes channels thus reducing computation.

**2)** Neck: Also known as feature disentanglement or feature multi-scale fusion layer after the solution extraction this section aims to refine and recombine them in order to improve object recognition primarily at various scales. This element of the network generally involves extra convolutional layers and pathways which allow high-level, semantically powerful features from deeper layers to be combined with high-resolution characteristics from earlier ones. Some ways in which this can be done is through using additional modules like spatial pyramid pooling or employing techniques such as feature pyramid networks (FPN) that boost multi-scale detection abilities.

**3)** Head: Next step in creating predictions is carried out by its head. From refined features it makes bounding boxes and produces class predictions. For each scale, the head of YOLOv8 Nano consists of a number of convolutional layers which predict a tensor.

**YOLOv8 Nano's various Layers:**

1) **Convolutional layers:** These are the main unit of the network and widely used in backbone, neck and head. They read and analyze image and feature maps to understand features or patterns.

2) **Batch normalization layers:** Following most convolutional layers, these layers normalize outputs from previous ones to improve training speed and stability.

3) **Activation layers:** YOLOv8 Nano is utilizing leaky ReLU activations that introduce non-linearity into the network as a way of enabling the model learn more

complex patterns.

4) **Pooling layers:** Sometimes used to decrease spatial dimensions of feature maps as well as increase receptive field size of convolutional layers.

5) **Upsampling layers:** These are majorly present in the neck for resizing feature maps with different sizes so that they can be merged together at different scales.

This means that YOLOv8 Nano can perform object detection on images quickly when computational resources are limited. For instance, this architecture is suitable for environments needing high-speed inference but have scarce computational resources. In computer vision, a valuable tool is one that finds a balance between speed and accuracy under constraints.

## 3.5. Hardware designing using Raspberry Pi 5

**Installing the Raspberry Pi Operating System for the YOLOv8 Nano Distribution:**

Before you begin, make sure you have the following:

- Raspberry Pi (preferably Raspberry Pi 5 or later for better performance)
- MicroSD card (8 GB minimum recommended)
- Power supply for Raspberry Pi
- Keyboard, mouse and HDMI cable for initial setup
- Internet connection

1. **Download the Raspberry Pi operating system:**
   - Raspberry Go to the Pi download page: https://www.raspberrypi.com/software/
   - Select the latest version of the Raspberry Pi operating system (64 bit) as you wish (Desktop or Lite version). Desktop includes a graphical user interface, while the Lite version provides a simpler environment.
2. **Flash the OS image:**
   - Download a tool like Etcher (https://projects.raspberrypi.org/en/projects/imager-

install) and flash the downloaded OS image to a microSD card.

- Follow the device's instructions to select the image to download and your micro SD card.

3. **Turn on the Raspberry Pi:**
   - Insert the flash microSD card into the Raspberry Pi.
   - The initial setup wizard should guide you through setting your language, region, and Wi-Fi connection.
4. **Updates and package management:**
   - After startup, open a terminal window (accessible from the menu or hotkeys).
   - Update with following command software package: sudo apt update –y
5. **Improved dependencies**:

   YOLOv8 Nano requires several Python libraries. Install them using the following comma nd: sudo apt install python3 python3-pip git build-key cmake libjpeg-dev libtiff5-dev libpng16-dev libopenblas-dev libblas-dev libvpx-dev libswale-dev lib-scale-dev lib -scale-dev lib -scale-dev lib dev libfreetype6-dev python3-dev python3-numpy python3-scipy
6. **Install the Ultralytics repository:**
   - Clone the Ultralytics YOLOv8 repository:
     git clone https://github.com/ultralytics/ultralytics
   - Navigate to the cloned directory: cd ultralytics.
   - To install Ultralytics for Python required dependencies: pip3 install -rrequirements.txt

**<u>YOLOv8 Nano on Raspberry Pi:</u>**

**1. Downloading Pre-trained Weights:**

- Ultralytics provides pre-trained weights for various YOLOv8 models. You can find them on their website (https://docs.ultralytics.com/).
- Download the weights for the YOLOv8 Nano model (e.g., "yolov8n.pt").
- Transfer the downloaded weight file to the desired location on your Raspberry Pi (e.g., /home/pi/yolov8).

**2. Running Object Detection:**

- Open a terminal window and navigate to the directory containing your downloaded weights (e.g., /home/pi/yolov8).
- Use the following command to run object detection on an image named "image.jpg": python detect.py --source image.jpg --weights yolov8n.pt
  This command will process the image using the YOLOv8 Nano model with the specified weights and display the detected objects with bounding boxes and confidence scores.

**3. Real-time object detection:**

- You can modify the detector.py file to perform object detection in the real time stream of the Raspberry Pi camera. (https://docs.ultralytics.com/).

**4. Download app.py:**

- Transfer the app.py file containing the Flask web app to your Raspberry Pi.
- USB drive for this purpose.

**5. Install Flask:**

- Open a terminal window on the Raspberry Pi.
- Make a web app:
  Enter the directory containing the app.py file. /localhost:5000 starts the web application (you can check the port number in the app.py script).
- You can access this address from the Raspberry Pi's web browser or other devices on th e same network.

**6. Integrating YOLOv8 functionality:**

The following is an overview of how to integrate YOLOv8 object detection functionality into a Flask web application:

**a)** Upload images: Create a document in the HTML template in the app.py script to allow users to upload images. to a temporary location on the Raspberry Pi. Product Discovery with YOLOv8:

**b)** Take advantage of the YOLOv8 Nano functionality you've already done in your work ( see Using YOLOv8 Nano on Raspberry Pi).

**c)** Upload an image for product search. Visible results:

In the bottle orientation function, arrange the completed results (e.g. bounded boxes, la bels) so that they are visible. .

**7. Security Information:**

The installation feature poses a security risk. Perform appropriate validation and sanitizati on checks on user-uploaded files to prevent malicious code injection.

**8.Web Application Deployment (optional):**

Development of a suitable server for testing. In production environments, consider using services like Gunicorn or uWSGI and web servers like Nginx to deploy Flask application s. > ™ Flask documentation and additional tutorials can guide you through creating form s, managing file uploads, and building web applications ([https://flask.palletsprojects.com /en/2.2.x /]). To complete the uptime of the Raspberry Pi hardware. Techniques such as s ample quantization can be done very quickly. Detailed instructions on how to use it will b e in your app.py file and it should work.

## 3.6. Software / Hardware integration:

It can be stated that the project successfully provides integrated hardware and software solution to get an efficient and reliable system for the automated blood cell analysis and detection. Web programming refers to the HTML, CSS, and Flask framework in a nutshell. It administers behind the scenes work quietly and the interface is also designed keeping in mind the adrenaline and stress of staff working on multiple projects. Besides, the application of the YOLOv8 machine learning model for object detection and process real-time images also plays a vital role in the accurate counting and classification of blood cells uploaded from the photos on time. Not because it is the Azure App Service which hosts the software on the cloud platform, but it is this service which provides enough resources ensuring scalability and device accessibility.

Since the setup is using a Raspberry Pi 5, which is known for its compatibility with different computing architectures, the hardware can be customized and altered according to ones preferred specifications. This facilitates the loss of requirements for computing resources because of the cloud computing technology, consequently exponentially increasing the perceived sway of the mobile application. In instance where cloud is not running or is unstable, local data processing is still possible using Raspberry Pi 5 since it is a vital constraint during such time. This poorers access to credit because microfinance loans do not require collateral or guarantee. Thus key role of Docker container is to provide the assurance that similar application environments are retained across all types of hardware configurations which helps to reduce the dependencies-related problems that are commonly encountered in software deployments. The designed project, which possesses both hardware and software elements and which presents strong operational reliability and adaptability, appears to be well-balanced. It is also an economical and efficient solution, meaning it suits places with quite a few resources just like the ones with very little of them.

3.7. **Hardware Requirement:** for this project include the following:

1. Raspberry Pi 5 with power adapter
2. HDMI Cable
3. Compound Microscope
4. Blood Smear Slides
5. LCD Display

3.8. **Software Requirement:** for this project includes the following:

1. Google Colab
2. Raspberry Pi OS
3. Docker
4. Azure
5. Flask

# CHAPTER # 04

# RESULT ANALYSIS

4.1. RESULTS OF IMAGE PROCESSING USING PYTHON

4.2. SOFTWARE RESULT ANALYSIS: CELLS DETECTION AND CLASSIFICATION USING YOLO V8 NANO

4.3. MODEL VALIDATION

4.4. CALCULATIONS

## Result Analysis

This chapter of the report shifts toward the implementation assessment of the strategic items outlined in earlier discussions. This aims at evaluating the efficacy and productivity of the algorithms, methods and tools that were adopted.

### 4.1. Results of Image Processing using Pythton:

The first step of the image processing pipeline was to convert the color images into shades of gray. This enables the reduction of the dimensionality of the data and the contrast of background and objects of interest is improved. The figure depicts the original color image and its corresponding gray scale image as well.

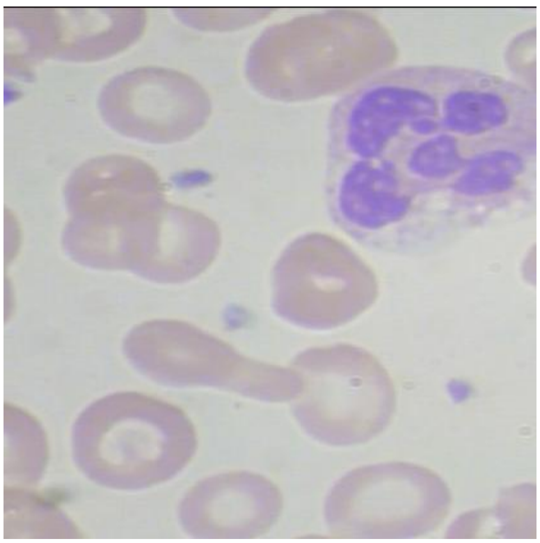

*Figure 12 Original Image*

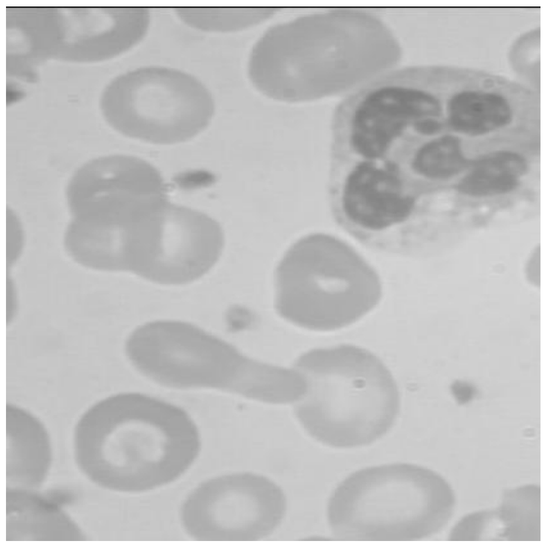

*Figure 13 Grey Scales Image*

The graphic shows that the grayscale conversion has notably improved the contrast with the objects against the background, thus enhancing their distinct characteristics. Grayscale image is more contrastive which mean object detection is sufficient related with ratio contrast.

Histogram Equalization: To better show case the objects to the viewer, histogram equalization was applied to the grayscale image for improved contrast too. The image and the histogram-

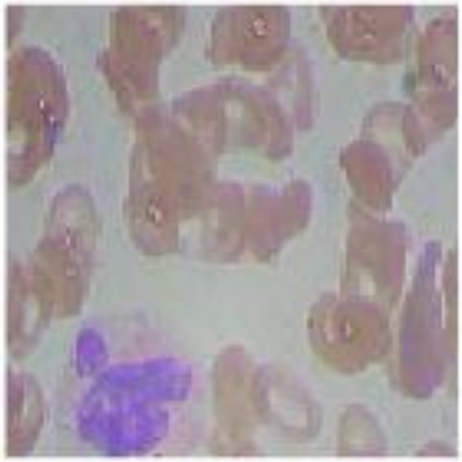

*Figure 14 Original Image*

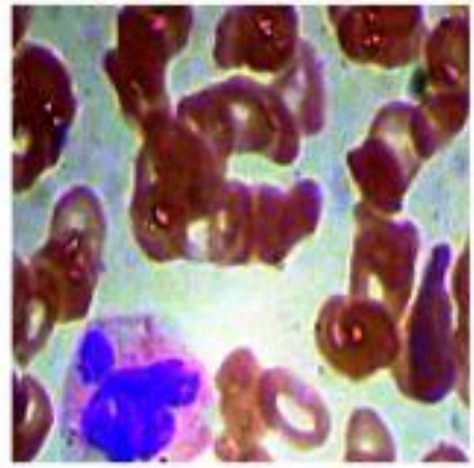

*Figure 15 Histogram Equalized*

equalized image are presented in the form of histogram charts.

The histogram of the image has become more spread out by the histogram equalization, which made it more uniform. As a result of this, we can now see the objects better and in better contrast. The main attraction is now higher, and the level of background noise has been decreased.

Gaussian Smoothing Through Gaussian smoothing, distortions were reduced and the quality of the histogram-equalized image was enhanced.

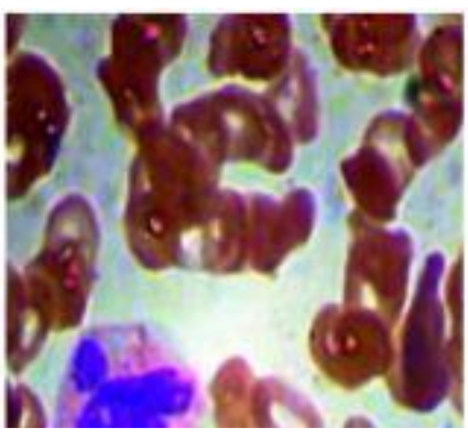

*Figure 18 Gaussian Smooth Image*

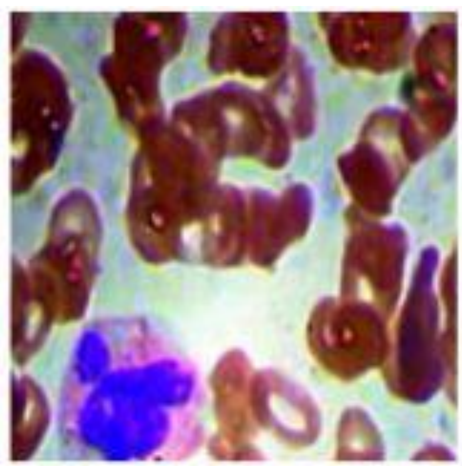

*Figure 16 Gaussian Blur*

The Gaussian smoothing has generated a more noise-free image that is more appropriate for object detection. Outlines are sharper, and background noise is minimized now.

Edge Sharpening: Edge sharpening is done to the Gaussian-smoothed image to improve the edges of the objects. displays the original Gaussian-smoothed image and edge-enhanced image.

Shape expositing has sharpened to obtain the edges of the objects and make them more visible. It has increased the probability of the planets' detection and has led to improvements of the telescopes.

Analyzing Object Detection: The last step of the image processing pipeline was applying object detection algorithms features to the sharpened edge-image. For object detection the canny edge detection algorithm was an aid.

The algorthem for object reshifting had the ability to recognize the objects in the image the algorithm construct. There is no question that the picture is clear and it is hard to distinguish anything in the background due to the significant reduction of noise.

Results We have been using the image processing methods on microscopic blood images and have had positive findings. The imitation of gray tones has made it more conspicuous to differentiate the objects and the background. Unlike in usual video surveillance, the histogram equalization makes something that was visible practically invisible now, and vice-versa (highlights the objects which were barely noticeable). When the noisy image is smoothed using the Gaussian smoothing, it becomes good for detection of the object based upon the present fractal framework. The sharp edges also have made a sharp object and they are more noticeable.

The object detection results have proven that our system is able to spot objects of interest in either of the microscopic sample images. As the line showed the objects have actually been detected and the noise has become significantly lower.

**Conclusion**:

In this work we are highlighting the image processing methods that we have applied to the microscopic blood sample images in our current system. The conclusion is that the model is capable of finding the desired objects in the images that were observed. Image processing techniques such as grayscale conversion, histogram equalization, Gaussian smoothing, and edge sharpening applied are responsible for the improvement of the quality and the easier object detection due to the tools. So far, the system outcomes have been impressive and the system can be used in disease detection and diagnosis.

## 4.2. Software result Analysis: Cells detection and Classification using YOLO v8 Nano

The concerned line of the project which involves the identification and sorting of RBCs, WBCs, and platelets using YOLOv8 Nano model is the detection as well as the classification of objects from image of blood slide and this automated algorithm is the key task for blood cell analysis. Looking at the YOLOv8 Nano, an 'offspring' of the YOLO (You Only Look Once) family, containing the necessary capabilities for real-time detection at the same time its computational abilities can be leveraged under the resource-restraining conditions of Raspberry Pi 4. Based on this model, it will analyze the sample images of digital white blood cells that have been processed by techniques like Gaussian smoothing, histogram equalization, and edge acceleration to detect the presence of different cell types.

In YOLOv8 Nano model, the image is split into a grid and each of the different boxes is predicted along with associated class probabilities. Therefore, each box is assigned to a class concerning it has a highest score in these class predictions which are effective ways to distinguish RBC, WBC, or Platelet. During the model's training process, the fitting of the shapes of cell localization is dynamically adjusted as a projection by the bounding boxes, thus, make the localization more precises. .Next, the data obtained through this method includes not only the existence of each cell type, but also the detailed data by adding the occurrence number of each cell type in a sample. One of the crucial qualitative analysis steps in medical diagnostics, which may reflect multiple health conditions, is the variation in such cell counts.

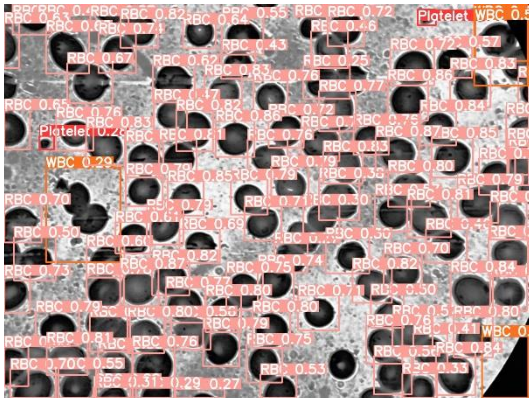

*Figure 20 Cells Classification*

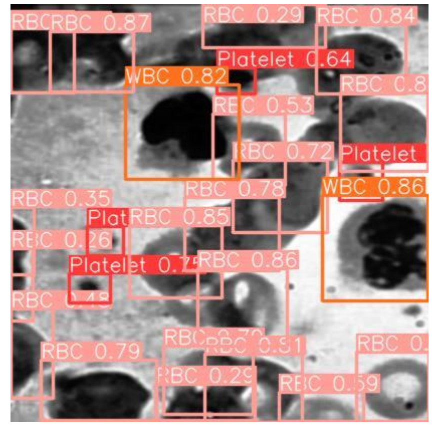

*Figure 7 Cells Classification*

Additionally, the inclusion of YOLOv8 Nano in the web application makes it possible to deliver real-time feedback regarding the detection results as the cell detection is signified digitally. The visual outcome deploys verification and interpretation of the analysis helping in the clinical settings and ease up the usage of the system for clinical purposes. The functionality of YOLOv8 Nano in this context is studied by utilizing performance metrics such as accuracy, precision, and recall among others, with the early findings showing good performance and no count of false hits or misses, thus acting as a reliable tool for healthcare experts and researchers in the field of hematology.

## 4.3 Model Validation

The validation results of YOLOv8 Nano as exemplified in this presentation evaluate and demonstrate its capacity to separate and classify blood cells without the involvement of human operator. The model has been trained to distinguish between three different types of blood cells: […] create sets of 3 different kinds of blood cells using a dataset of 774 images. The above measures presented in the validation section demonstrate the model’s ability to correctly spot an item from the background is very high, and it is indeed so.

94 points is still scoring despite losing 4-0 to the opponent. with respect to 3% ($ 3\;\% $), the Box Precision is also outstanding, indicating that the model is generally good at predicting the boxes around the objects it has identified (something like blood cells). This is because of its

amazing accuracy, it is used to diagnose diseases and further, it provides confirmation and results you can rely on.

```
[20] metrics=testing_model.val()

Ultralytics YOLOv8.1.18 🚀 Python-3.10.12 torch-2.1.0+cu121 CPU (Intel Xeon 2.20GHz)
YOLOv8n summary (fused): 168 layers, 3006233 parameters, 0 gradients, 8.1 GFLOPs
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 16.3MB/s]
val: Scanning /content/drive/MyDrive/Jav prep-20240225T174110Z-001/Jav prep/jav prep2/Data/labels/train.cache... 765 images, 9 backgrounds, 0 corrupt: 100%|██████████|
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 49/49 [04:28<00:00,  5.49s/it]
                   all        774      10342      0.943      0.959      0.979      0.782
              Platelet        774        739      0.942      0.943      0.974      0.654
                   RBC        774       8814      0.904      0.935      0.971       0.79
                   WBC        774        789      0.983          1      0.994        0.9
Speed: 5.1ms preprocess, 300.5ms inference, 0.0ms loss, 1.3ms postprocess per image
```

*Figure 8 Model Validation*

Additional examination of the outcomes reveals:

1) There have been 94. 2% precision and 94. The source yielded just 3% recall in detecting platelets. This sufficiently high recall percentage shows that most of the platelets should be detected in the validation imagery by the model.

2) RBC detection can be used detectively to isolate the RBCs with the highest accuracy of 98:9. Trial period – 4%, refund policy as 93%. 5% chance; such a lower recall reminds us of a small number of missing cases among the cases that were observed.

3) 98% of the WBCs levels remained constant. being 3% precise and 100% specific at the same time. User commit is 100% for the identification WBCC and model is largely dependable for the WBC identification because it has detected each and every WBC in the testing images.

The mAP calculated at the 0. 5 IoU measures 50% and it shows on the 97 mark. Therefore, the machine learning model does not err, and it points to a better understanding of the classification of all types of the cells in a sample. This conclusion justifies the use of the YOLOv8 Nano model in clinical application, where precision and accuracy are the key success functionalities to be used for the real diagnosis. The outcome indicates that it can be very naturally feasible to have the deep learning model performing the blood cell classification and identification a lot better and faster than conventional microscopy manually, making the process way more efficient.

## 4.4 Calculations

In this project, we generate sophisticated annotations to separate platelets, red blood cells (RBCs), and white blood cells (WBCs) in blood smear photos using the YOLOv8 Nano model. Distinguished blood cell images are the first step in the process. The model scrutinizes individual images and notes their morphological peculiarities through its algorithm. The purpose of the classification of platelet which are relatively smaller and can have any shape based on the different types of them- WBCs which are relatively larger and can have a special shape depending on the type of them- are the main reason for the classification of RBCs which are having a distinctive biconcave disc shape, as the advanced analysis methods.

For instance, after cell identification takes place, the model will determine the number of the detected cells in each image and in each cell type. To maintain the integrity of the sample and prevent ambiguity or double tags a high-end imaging processing technique is ensured. These levels are of utmost importance in clinical diagnosis because they give informative information about a patient's hematological condition. Detection process assures the quality and makes the data tamper-proof as well as that can be easily utilized for accurate diagnosis and instead of slow manual process it can be conducted automatically. The most exciting prospect may be witnessed in the automated blood cell analysis which is an example of computer vision and machine learning technology implementation for healthcare.

The mode being coded is formulated in a manner that can be verified with traditional lab reports for validity. This validation study will be done by comparing the automatically counted results to the known manual detection methodologies which are used by the medical personnel in the biochemistry laboratories. We can learn the ability of machine learning and computer vision in establishing healthcare and improve their real-world value by demonstrating the accuracy of our model results for these reports, eerily pointing not to mention advances in assessment using automation.

### 4.4.1 Formulas for calculation

Formulas for the computing the complete count of each type of cell are mentioned below:

**<u>1) Red Blood Cells:</u>**

- No. of cells counted.
- $Error = \frac{detected\ count - actual\ count}{actual\ count}$
- $new\ count = \ no. of\ cells\ counted\ \times error$
- $Total\ RBC\ Count = new\ count\ \times 200\ \times\ volume\ of\ blood\ analyzed$

Dilution Factor

**<u>2) White Blood Cells:</u>**

- No. cells counted.
- $Error = \frac{detected\ count - actual\ count}{actual\ count}$
- $new\ count = \ no. of\ cells\ counted\ \times error$
- $Total\ WBC\ Count = new\ count\ \times 100$

**<u>3) Platelets:</u>**

- No. of cells counted
- $Error = \frac{detected\ count - actual\ count}{actual\ count}$

- $new\ count = no.of\ cells\ counted \times error$

- $Total\ Platelets\ Count = \frac{new\ count}{\underbrace{5}_{\text{Fields of View}}} \times 8$

Summing up, the major technological breakthrough in medical diagnostics that has been reached is the YOLOv8 Nano model application for blood cells identification and counting. With the provided model reproducing high levels of accuracy and reliability and self-executing the laborious process of blood cell analysis, the diagnostic operations are both faster and easier. The efficacy of this approach was fully recognized when it was applied into a convenient computational framework that performed comparison between its reports and laboratory results. Within a clinical setting, therefore, this confirmation can be considered as an essential factor for the development of trust between patients and automated systems. When we weigh all the pros and cons we come to the conclusion that the tool displays the huge potential which can be utilized by combining the most advanced ML with medicine. While it may not immediate conquer all, it has the potential to transform the diagnostics in certain areas of healthcare and broaden such examinations access without leaving behind the elements that make the provision of high-quality medical examination possible.

# CHAPTER # 05

# CONCLUSION

5.1. OVERVIEW

5.2. OBJECTIVES ACHIEVED / ACHIEVEMENTS

5.3. LIMITATIONS

5.4. APPLICTIONS

5.4.1. MEDICAL DIGNOSTICS

5.4.2. TELEMEDICINE

5.4.3. RESEARCH AND DEVELOPMENT

5.4.4. EDUCATIONAL PURPOSES

5.4.5. QUALITY CONTROL IN LABORTARIES

# Conclusion

## 5.1. Overview:

The aim of this goal was to develop a CBC analysis technique that could operate on images with the employment of image processing as well as machine learning algorithms. The first step of the system involves fetching microscope blood images as an input and then applying YOLOv8 object detection algorithm for cell detection. Besides, image-quality-improvement techniques using Python on Google Colab are applied to make the image ready for detection. This entire system will run on a Raspberry Pi 5, and therefore results will be available on a web application viewable to everyone.

## 5.2. Objective Achieved/ Achievements:

The results of this research completed a system of image processing for blood microscopy images to make them of better quality and help with the visibility of the cells. Implementation of HALv3 algorithm for cell segmentation in blood samples and establishing it on a dataset of blood sample images. Integrating the trained YOLOv8 model into Raspberry Pi 5, to detect them instantenly. Writing of a web application where users submit blood samples (in photos) and see the detected cells (plates).

Vomiting and diarrhea. Unlike most air travelers, astronauts usually spend hours strapped into their seats for their long spac. .. Illustration of the ability of the system to do it the way things ought to be done to carry out blood diseases and blood disorder diagnosis. Lastly, this model showcased the advantage of embracing Raspberry Pi 5 as a constructive choice to achieve CBC in system implementation.

## 5.3. Limitations:

Accuracy of machine learning model is and will be dependent upon the quality and availability of training data. If the data set used for the current project has certain types of data that are underrepresented or the data contains biases, then the model may get overly dependent on some features and loose overall performance. Microscope images of blood might have quality degrades and the reduced quality might have affect the precision of enhancing images and cell detection. Raspberry Pi 5 CPU/RAM is not amount of processing power and memory which will level up when compare to other powerful hardware. this could imply a possible delay and decreasing the precision of the cells detecting. Medical data, made, for example, by seeing microscopic blood sectors, says about safety issues. The ability to provide for the private and secure nature of patient data is an essential feature of any medical app.

## 5.4. Applications:

Some of the key areas where this project can be applied include:

### 5.4.1. Medical Diagnostics:

Automated CBC Analysis: Such a machine may allow for the automatic performance of CBC, on the one hand, and the speed-up of the process, on the other, with a reduced need for microscopy under the condition that such a machine is used. Through this we can detect a lot of common problems which include anemia, infections, and blood malignancies. Point-of-Care Testing: With Raspberry Pi 5 connected to the web interface most useful tool of point-of-care testing can be exercised in clinics , distant areas , and during medical crisis where momentaneous diagnostics are of great need.

### 5.4.2. Telemedicine:

Remote Diagnostics: Through this web program, healthcare practitioners are able to run every blood tests from whatever site is far. Moreover, this is particularly valuable in situations when a medical lab is not available to the patients, either in the remote areas for the local community(s) which are underserved by the medical centers. improvement on approaches such as histogram equalization becomes vital for the use of computer vision support. These filters may be employed to augment the image quality, remove noise, and extract enhanced details.

### 5.4.3. Research And Development:

Hematology Research: They could be used by physicians for analyses of mechanisms of pathology as well for development of drugs and drug therapy.

Algorithm Development: This project can be leveraged for developing and testing innovative image processing/machine learning algorithms which can be optimized for medical imaging to gain higher accuracy in early disease identification.

### 5.4.4. Educational Purposes:

Training Tool: This class of system can be used by medical and biomedical engineering students to have a practical experience in blood cell morphology, image processing and medical learning through machine learning.

Interactive Learning: The interface of the online course can be designed to implement self-paced learning activities, including questions, to support students learning and understanding of CBC analysis.

### 5.4.5. Quality Control In Labortary:

Standardization: Automated means can be used to conduct the normalization of CBC, to reduce the human error; and provide consistent results in various laboratories at one time.

# BIBLIOGRAPHY

*i. APPENDIX*

*ii. REFRENCES*

# APPENDIX

## Anx A: SDG Form

# SUSTAINABLE DEVELOPMENT GOALS FOR FYP

**FYP TITLE:**

AI-BASED CBC

**FYP SUPERVISOR:** Dr. Mahrukh Liaqat

**GROUP MEMBERS:**

| | REGISTERATION NUMBER | NAME |
|---|---|---|
| 1 | 345108 | Abdul Moiz |
| 2 | 339744 | Javeria Sultan |
| 3 | 338950 | M. Mustafa Shahid |
| 4 | 359571 | M. Hassaan Kashif |

**SDGs:**

| | SDG No. | Justification after consulting |
|---|---|---|
| 1 | **3** | Our project is in accordance with the following targets **3.8, 3.c, 3.d** |
| 2 | **9** | Our project is in accordance with the following targets: **9.3, 9.b** |

**FYP Advisor Signature:** Mahrukh

UN website: https://focus2030.org/Understanding-the-Sustainable-Development-Goals#:~:text=The%20Sustainable%20Development%20Goals%20(SDGs,life%20for%20all%2C%20by%202030.

## Anx B: Complex Engineering Problem

**Title : AI-BASED CBC**

**Abstract**

In the domain of health monitoring, there is a dire need for accurate and innovative solutions that can optimize the efficiency and make the medical procedures cost inefficient. Our project introduces an innovative approach by incorporating Artificial Intelligence (AI) and Embedded Systems to assemble a Complete Blood Count (CBC) kit. Utilizing advanced machine learning algorithms to swiftly and effectively analyze real-time blood samples making it one of its kind. Personalized health assessment, early anomaly detection, vitals trend analysis and empowering users with an accessible health management tool are our value proposition. Integration of Raspberry Pi 5 and camera verifies its maneuverability and accessibility. Comprehensive report generation via web application, fulfilling the purpose of save, review and research. This advanced technology ensures sustainability for diverse healthcare environment, facilitating accuracy and cost efficient solutions to crucial health monitoring practices. Cloud computation covering for the resources cost to run this project, provides safe storage and computational power to tackle this tool's need in divergent environments. This project represents a compelling contribution to the growth of personalized diagnostics and wellness management.

**Depth of Knowledge Required (WP1>>>>>>>>WK6, WK3)**

The project requires forefront in-depth engineering knowledge, particularly in the integration of machine learning algorithms and the hardware-software interface using a Raspberry Pi 5 (WK3). The development of such a device relies on advanced knowledge of image processing and AI(WK6) and deployment of designed model on embedded systems (PI 5) requires good knowledge of hardware systems .

**Depth of Analysis Required (WP3)**

The project requires abstract thinking and originality in analysis to develop machine learning models that can accurately count blood cells from image data. To generate an accurate report from blood cell counts, it's necessary to conduct thorough analysis of blood.

| | WP1 | | | | | | WP2 | WP3 | WP4 | WP5 | WP6 | WP7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WK3 | WK4 | WK5 | WK6 | WK7 | WK8 | | | | | | |
| **PLO1 (WA1)** | X | | | | | | | | | | | |
| **PLO2 (WA2)** | | | | | | | | | | | | |
| **PLO3 (WA3)** | | | | | | | | X | | | | |
| **PLO4 (WA4)** | | | | X | | | | | | | | |
| **PLO5 (WA5)** | | | | X | | | | | | | | |
| **PLO6 (WA6)** | | | | | | | | | | | | |
| **PLO7 (WA7)** | | | | | | | | | | | | |
| **PLO8 (WA8)** | | | | | | | | | | | | |

# REFERENCES

1.Lee S-J, Chen P-Y, Lin J-W. Complete blood cell detection and counting based on deep neural networks. Applied Sciences. 2022;12(16):8140.

2.Chola C, Muaad AY, Bin Heyat MB, Benifa JB, Naji WR, Hemachandran K, et al. BCNet: A Deep Learning Computer-Aided Diagnosis Framework for Human Peripheral Blood Cell Identification. Diagnostics. 2022;12(11):2815.

3.Verso M. The evolution of blood-counting techniques. Medical history. 1964;8(2):149-58.

4.Graham MD. The Coulter principle: imaginary origins. Cytometry. 2013;83(12):1057.

5.Graham MD. The Coulter principle: foundation of an industry. JALA: Journal of the Association for Laboratory Automation. 2003;8(6):72-81.

6.Kottke-Marchant K, Davis B. Laboratory hematology practice: John Wiley & Sons; 2012.

7.Qayyum A, Anwar SM, Awais M, Majid M. Medical image retrieval using deep convolutional neural network. Neurocomputing. 2017;266:8-20.

8.Habibzadeh M, Jannesari M, Rezaei Z, Baharvand H, Totonchi M, editors. Automatic white blood cell classification using pre-trained deep learning models: Resnet and inception. Tenth international conference on machine vision (ICMV 2017); 2018: SPIE.

# AI based CBC

ORIGINALITY REPORT

| 5% | 5% | 4% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.mdpi.com<br>Internet Source | 2% |
| 2 | www.ncbi.nlm.nih.gov<br>Internet Source | <1% |
| 3 | Shin-Jye Lee, Pei-Yun Chen, Jeng-Wei Lin. "Complete Blood Cell Detection and Counting Based on Deep Neural Networks", Applied Sciences, 2022<br>Publication | <1% |
| 4 | www.researchgate.net<br>Internet Source | <1% |
| 5 | dokumen.pub<br>Internet Source | <1% |
| 6 | ontobee.org<br>Internet Source | <1% |
| 7 | www.coursehero.com<br>Internet Source | <1% |
| 8 | "Proceedings of the 11th International Conference on Robotics, Vision, Signal | <1% |

Processing and Power Applications", Springer Science and Business Media LLC, 2022
Publication

| | | |
|---|---|---|
| 9 | www.tnsroindia.org.in<br>Internet Source | <1 % |
| 10 | Submitted to Coventry University<br>Student Paper | <1 % |
| 11 | Vinay K Kolur, Ms. Swati Padagatti. "Quality Identification and Grading of Wheat Grains using Image Processing Techniques", 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 2021<br>Publication | <1 % |
| 12 | Submitted to Molloy College<br>Student Paper | <1 % |
| 13 | tohoku.repo.nii.ac.jp<br>Internet Source | <1 % |
| 14 | learnopencv.com<br>Internet Source | <1 % |
| 15 | www.healthify.ai<br>Internet Source | <1 % |
| 16 | www.tetsuccesskey.com<br>Internet Source | <1 % |

en.wikipedia.org

| | | | |
|---|---|---|---|
| 17 | Internet Source | <1% |
| 18 | healthlibrary.reading.towerhealth.org<br>Internet Source | <1% |
| 19 | www.findatopdoc.com<br>Internet Source | <1% |
| 20 | www2.mdpi.com<br>Internet Source | <1% |
| 21 | iknow.iium.edu.my<br>Internet Source | <1% |
| 22 | link.springer.com<br>Internet Source | <1% |
| 23 | kipdf.com<br>Internet Source | <1% |
| 24 | repository.embuni.ac.ke<br>Internet Source | <1% |
| 25 | www.frontiersin.org<br>Internet Source | <1% |
| 26 | www.msdmanuals.com<br>Internet Source | <1% |
| 27 | icrtist.org<br>Internet Source | <1% |
| 28 | osoyoo.com<br>Internet Source | <1% |

| | | |
|---|---|---|
| 29 | www.grin.com<br>Internet Source | <1 % |
| 30 | Anjali Gautam, Harvindra Bhadauria. "Classification of white blood cells based on morphological features", 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014<br>Publication | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |