# Project Report

**Course Code: CSE6502 / Course Title: Major Project**

# Project Title

## A biometric approach to blood group detection using fingerprint analysis could enhance medical screenings

Saurabh Shrivastva
230160302001

*A report submitted in part fulfilment of the degree of*

**Program:** Master of Technology - Computer Science and Engineering

**Supervisor:** Prof. (Dr.) K. Thammi Reddy


Department of Computer Science and Engineering
School of Engineering & Sciences
GD Goenka University, Gurgaon

April 2025

# Declaration by the Student

I hereby grant permission to GD Goenka University to publish my submitted project/research work for academic purpose. I understand that the published copy may be accessible to the public through the university's library, digital repositories, and other relevant platforms. This consent encompasses both print and electronic formats.

By signing this consent form, I give my consent to publish my dissertations as described below:

**Student Name:** Saurabh Shrivastva

**Student Affiliation:** GD Goenka University

**Title of work:** BLOOD GROUP PREDICTION USING DEEP LEARNING

**Date:**

**Place:** GD Goenka University

**Student Signature:**

# Declaration from Supervisor

I hereby grant permission to GD Goenka University to publish my submitted dissertations//research papers/academic content written/supervised/guided/co-authored for academic purpose. I understand that the published copy may be accessible to the public through the university's library, digital repositories, and other relevant platforms. This consent encompasses both print and electronic formats.

By signing this consent form, I give my consent to publish my work as described below:

**Faculty Name:** Ms. Manka Sharma

**Faculty Affiliation: GD Goenka University**

**Title of work:** BLOOD GROUP PREDICTION USING FINGERPRINT

**Nature of Work:** BLOOD GROUP ANALYSIS

**Date:**

**Place: Gurgaon**

**Faculty Signature:**

# Acknowledgement

I would like to express my heartfelt gratitude to my supervisors and my Mentor, Prof. (Dr.) K. Thammi Reddy Dean of Soes and Ms. Manka Sharma Assistant Professor, for their invaluable guidance, encouragement, and continuous support throughout this project. Their expert advice, constructive feedback, and mentorship played a crucial role in shaping the direction and depth of my research. I am also thankful to my friends and classmates for our meaningful discussions, insightful suggestions, and assistance during data collection and analysis. Their support greatly contributed to the successful execution of this work. I sincerely appreciate the volunteers who generously provided access to the fingerprint dataset used in this study. Their cooperation and willingness to assist were fundamental to the progress of my research. I am deeply grateful to my family and friends for their unwavering encouragement, emotional support, and motivation throughout my academic journey. Their love and belief in me have been a constant source of strength. Finally, I would like to extend my appreciation to GD Goenka University and the Department of Computer Science and Engineering for providing the necessary resources and support, particularly in facilitating dataset acquisition, which was vital to the successful completion of this project.

# Table of Contents

Abstract…….3

Project Specification…….4

Chapter 1: Introduction…….5

1.1 Literature Review……5

Chapter 2: Research Gaps…….6

Chapter 3: Project Objective…….8

Chapter 4: Project Hypothesis……...9

Chapter 5: Research Methodology……10

Chapter 6: Results & Discussion……11

Chapter 7: Conclusion……12

Bibliography……...13

# Abstract

This project explores an innovative biometric approach to predict human blood groups using fingerprint analysis combined with deep learning techniques. Traditional serological methods for blood typing, while accurate, require invasive procedures and specialized equipment, limiting their practicality in emergency or resource-constrained situations. Fingerprints, being unique, permanent, and easily collectible, present a non-invasive alternative. Recent studies indicate correlations between fingerprint patterns and blood group types, and biomolecules present in sweat residues may serve as biochemical markers for blood group classification. Leveraging these insights, this research employs convolutional neural networks (CNNs) to analyze fingerprint images and predict ABO and Rh blood groups. The project involves constructing a diverse fingerprint dataset labeled with verified blood types, preprocessing images for feature extraction, and training custom CNN models to identify subtle pattern-blood group associations. This method aims to enable rapid, non-invasive, and accessible blood group detection, potentially benefiting emergency medical responses and forensic investigations. Experimental results demonstrate promising classification accuracy, suggesting the viability of fingerprint-based blood group prediction as a complementary tool to conventional blood typing methods.

# Project Specification

## Project Title:

A biometric approach to blood group detection using fingerprint analysis could enhance medical screenings

## Project Description:

This project introduces an innovative biometric approach to predict a person's blood group using fingerprint analysis. By utilizing advanced image processing and deep learning techniques, the system examines distinct fingerprint patterns to uncover potential associations with blood group types. This method offers a fast, non-invasive solution for determining blood groups, which could be especially valuable in emergency medical situations or regions lacking adequate healthcare facilities.

## Objectives:

- To develop an efficient fingerprint preprocessing pipeline for feature extraction.
- To design and train deep learning models capable of classifying fingerprints by blood group.
- To evaluate the model's performance on real-world datasets collected with volunteer support.
- To contribute to biometric-based health diagnostics with a focus on accessibility and speed.

## SDG 9: Industry, Innovation, and Infrastructure

**Target 9.5 – Enhance scientific research and upgrade technological capabilities of industrial sectors.**

- Utilizes **deep learning and biometric technologies** to enhance medical diagnostics, contributing to **healthcare innovation**.
- Can be integrated into **hospital infrastructure** and emergency response systems.

# Chapter 1: **INTRODUCTION**

## 1 Blood Group Systems and Fingerprints

Identifying human blood groups is essential in emergency medicine and forensic science. While serological methods are accurate, they often require specialized equipment and trained personnel, making them less practical in urgent situations such as trauma care, disaster response, or crime scene investigations.

To address these limitations, recent advancements in biometrics and artificial intelligence have led to the exploration of non-invasive alternatives. In particular, fingerprint analysis presents a promising solution. Fingerprints are permanent, unique, and easy to collect. Research indicates a statistical correlation between fingerprint patterns—such as loops, whorls, and arches—and specific blood groups [6]. Additionally, sweat secreted through fingerprint ridges may contain proteins and antigens associated with the ABO and Rh blood groups, serving as biochemical markers for blood type identification [2].

This study investigates the potential of combining statistical analysis with deep learning techniques, specifically convolutional neural networks (CNNs), to predict blood groups from fingerprint images. CNNs are particularly effective for image recognition and can detect intricate features in fingerprint data. By training models on a diverse dataset of fingerprint images paired with verified blood groups, this research aims to identify patterns that enable rapid and reliable classification.

This approach could significantly enhance decision-making in critical medical situations and improve forensic investigations where only fingerprint evidence is available. As this field evolves, fingerprint-based blood group detection may serve as a practical and scalable complement to traditional methods.

### 1.1 Organization of the project

- **Blood Group Systems and fingerprints:**

  This chapter provides an introduction to our project on predicting blood groups from fingerprints. We discuss the relevance of this research, basic concepts of blood groups, and the association between fingerprints and blood groups. We outline the project's objectives and methodology.

- **Methodology and Deep Learning Approach:**

  In this chapter, we describe the data collection process, including fingerprint dataset diversity and preprocessing techniques. We present the model used—a custom Convolutional Neural Network (CNN)—and explain the training process.

- **Implementation and results:**

  Here, we present the results of our trained models, including performance metrics such as accuracy, precision, recall. We analyze the strengths and limitations of each model in predicting blood groups from fingerprints.

# Literature Survey

## 1.2 Blood Group Systems and Clinical Significance

### 1.2.1 Historical

In the past, countless lives were lost due to a lack of knowledge about the blood group system, especially during blood transfusions that often turned fatal. This continued until the year 1900, when Karl Landsteiner, working at the University of Vienna, uncovered the reason why some transfusions worked while others did not. By experimenting with blood serum and red blood cells from his colleagues, Landsteiner observed that certain combinations caused clumping, or agglutination, which led to the identification of different blood groups.

Through his research, he initially classified blood into three groups: A, B, and C. Later, the group labeled "C" was renamed "O," derived from the German word *"ohne,"* meaning "without" or "zero," signifying the absence of antigens on the red blood cells (Dariush & Marjan, 2013). A year later, a fourth, less common blood group, AB, was discovered. In recognition of his groundbreaking work, Karl Landsteiner was awarded the Nobel Prize in Physiology or Medicine in 1930 [1].

### 1.2.2 Definitions and concepts

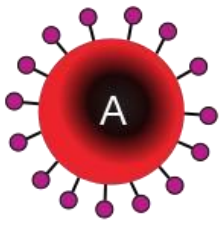
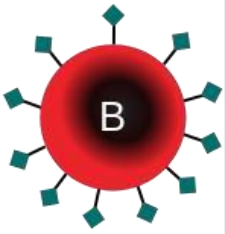
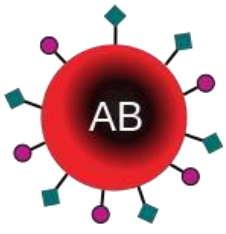
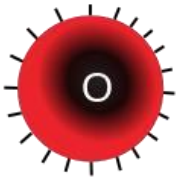
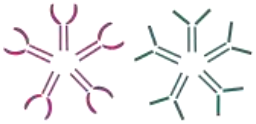
The term blood group refers to an entire classification system that includes specific antigens located on the surface of red blood cells (RBCs). These antigens are governed by genes that may be allelic or closely linked on the same chromosome. In contrast, blood type pertains to the particular pattern of antibody responses associated with a specific blood group system.

Key antigens such as A, B, and Rh, present on the RBC membrane, can stimulate the immune system to produce antibodies when incompatible blood is transfused or donated. This immunological response may result in severe complications, including immune-mediated hemolysis, a potentially life-threatening condition. Consequently, blood typing is essential to identify these antigens and ensure safe transfusion practices, ultimately safeguarding patient health.

As reported by the International Society of Blood Transfusion (ISBT, 2022), there are currently 44 officially recognized blood group systems, encompassing 345 distinct antigens found on human RBCs.

Antibodies can arise either through an active immune response to foreign red cell antigens, such as those encountered during transfusion, or naturally, following exposure to environmental substances that resemble RBC antigens. The presence or absence of these antigens, influenced by hereditary genetic variation, determines an individual's blood type [2].

Among all blood group systems, the ABO system is the most significant in the context of blood transfusion and organ transplantation. By approximately six months of age,individuals typically develop clinically relevant levels of anti-A and/or anti-B antibodies. For example, individuals with blood type A have anti-B antibodies, while those with blood type B carry anti-A antibodies. Type O individuals lack both A and B antigens but possess both corresponding antibodies in their serum.

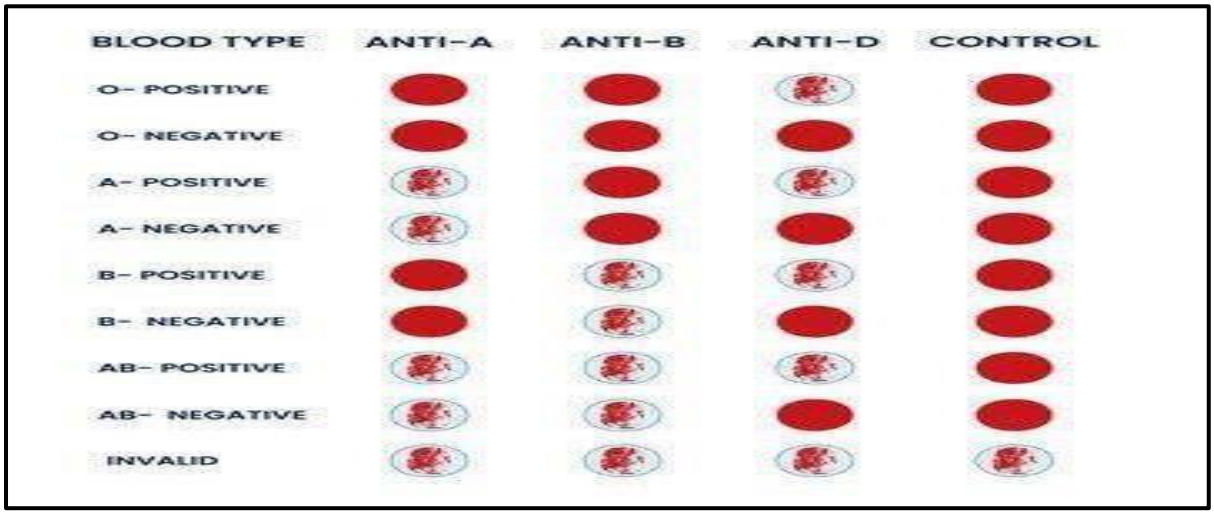
| | Group A | Group B | Group AB | Group O |
|---|---|---|---|---|
| Red blood cell type |   |   |   |   |
| Antibodies in plasma |  Anti-B |  Anti-A | None |  Anti-A and Anti-B |
| Antigens in red blood cell |  A antigen |  B antigen |  A and B antigens | None |

## 1.3 Blood Group Testing Methods

### 1.3.1 Serological testing

The slide method is a rapid and economical approach for identifying a person's ABO and RhD blood groups. In this technique, a clean glass slide or a white porcelain tile is sectioned into three parts. A small drop of the individual's blood is placed in each section and separately mixed with anti-A, anti-B, and anti-D antibodies. Blood type is determined by observing whether agglutination (clumping) occurs in any of the mixtures.

This test is advantageous for its speed, typically completed within 5 to 10 minutes, and its minimal need for reagents. However, it has notable limitations. The method is not highly sensitive and may fail to detect weak or rare antigens. It is primarily used for initial screening or in outdoor settings where laboratory facilities are unavailable. Moreover, if the antibody levels are low, results can be misleading, causing false positives or negatives. As a result, while the slide test can be useful in certain contexts, it is not recommended for confirming compatibility in critical blood transfusions.




### 1.3.2 Molecular Testing

Molecular-based approaches offer a highly accurate and detailed method for determining blood group antigens by detecting specific genes and DNA sequences associated with these markers. Unlike traditional serological methods, molecular testing provides greater precision and is capable of identifying the presence or absence of particular antigens with higher reliability. Techniques such as Polymerase Chain Reaction (PCR) and various DNA sequencing methods are frequently used in this context.

Although serology has been the gold standard for pre-transfusion testing for many years, it often falls short when identifying rare blood group antigens due to the limited availability of antibody reagents. To address these challenges, genetic testing—or genotyping—has emerged as a powerful alternative. Genotyping allows for the efficient detection of rare antigen profiles, especially in

cases where serological methods are insufficient. With its ability to process numerous samples simultaneously, genotyping has proven to be an effective and scalable solution.

Various genetic techniques like PCR-RFLP (Restriction Fragment Length Polymorphism), PCR-SSCP (Single-Strand Conformation Polymorphism), and direct DNA sequencing are routinely employed to analyze blood group genetics. These tests not only facilitate rare blood type identification but also help trace inheritance patterns and gene frequency distributions. Tools such as real-time fluorescent PCR have shown significant success in this area, and emerging technologies like Next-Generation Sequencing (NGS) are particularly promising, as they can detect rare antigen genes without prior knowledge of corresponding antibodies.

Today, the field has seen the introduction of many advanced genotyping kits that are both rapid and cost-effective. These kits often support multiplex testing and high-throughput analysis, making the identification of rare blood types more practical and widely accessible than ever before.

### 1.3.3 Alternative Testing Methods:

Recent innovations have introduced non-invasive alternatives to conventional serological and molecular blood typing techniques. These novel methods include testing based on saliva, urine, and fingerprint samples. By detecting specific blood group antigens or genetic markers in these alternative biological materials, such techniques aim to simplify the process of blood group determination.

Although these approaches are still in the experimental and validation stages, they offer promising advantages. Chief among them are increased convenience, faster testing times, and the elimination of the need for blood samples. With continued research and technological advancements, these non-invasive methods could become reliable tools for rapid and accessible blood group identification in clinical and remote settings.

#### 1) Saliva-based Testing:

Saliva, which contains secretions from the oral mucosa, can carry soluble blood group antigens in individuals known as secretors. Saliva-based blood typing involves the collection and analysis of saliva samples to identify these antigens or genetic markers linked to blood group identity. This technique presents a non-invasive and user-friendly alternative to traditional blood-based tests, making it especially suitable for large-scale screening or situations where drawing blood is impractical.

### 2) Urine-based Testing:

Urine has been explored as a potential sample for blood group identification, as certain blood group antigens or their byproducts may be present in urine. In urine-based testing, samples are collected and analyzed to detect these antigens or related genetic markers, helping to determine an individual's blood group. This approach provides a non-invasive and potentially more convenient alternative to traditional blood-based methods.

### 3) Fingerprint-based Testing:

Fingerprint-based blood group testing is an emerging approach still under investigation, which is also part of the focus of our project. This method leverages the unique ridge patterns and other distinctive features of fingerprints to predict blood type. Studies have indicated that certain blood group antigens or genetic markers can be found in the sweat and oils left behind on fingerprints. By examining these fingerprint patterns alongside biological markers, it may become possible to infer an individual's blood group.

## 1.4 Fingerprints:

### 1.4.1 What is a Fingerprint:

Fingerprints are formed by the patterns of ridges and grooves found on the fingertips, palms, and soles. These intricate designs, created by the friction ridge skin, develop before birth and generally stay the same throughout a person's life, except for changes caused by injuries or certain skin conditions.

There are three main fingerprint pattern types: arches, loops, and whorls. Arches are characterized by ridges that flow from one side to the other without looping. Loops feature ridges that enter and exit from the same side, curving in a consistent direction. Whorls are more circular or spiral in shape, containing concentric ridges.

What makes fingerprints so valuable is their individuality. Even identical twins, who share the same DNA, have completely different fingerprints. This uniqueness, along with their stability over time, makes fingerprints ideal for identification purposes.

Fingerprint-based recognition systems work by detecting specific features such as ridge endings, splits (bifurcations), and ridge counts. These details—called minutiae—are analyzed and compared using advanced algorithms in automated systems. Because of their accuracy and ease of use, fingerprint systems are widely implemented in areas like criminal investigations, personal authentication, identity documents, and secure access control.

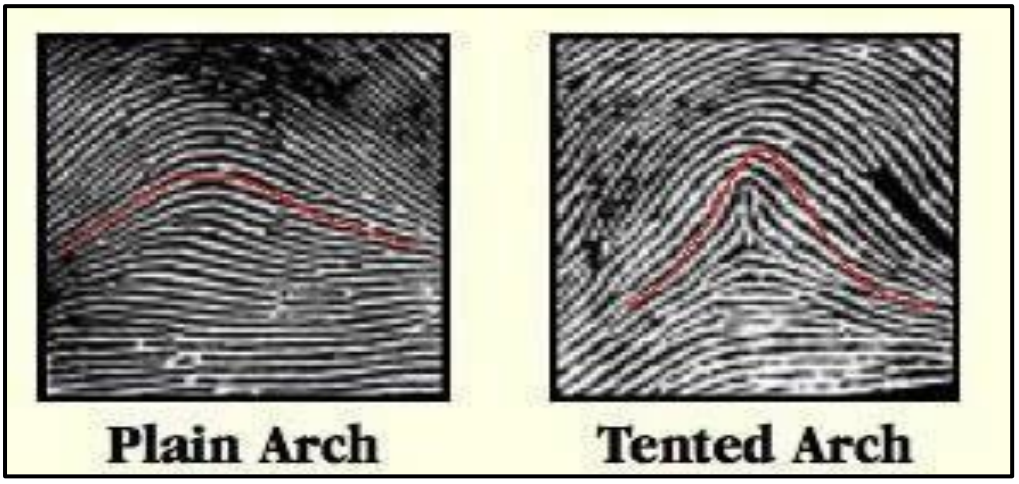
### 1.4.2 Fingerprint Types:

#### 1. Arches:

Arches represent the most straightforward and least frequently occurring fingerprint pattern, appearing in roughly 5% of the population. In this pattern, the ridges flow from one side of the fingertip to the other, forming a smooth arch without looping back. A key feature of arches is the absence of both a central core and a delta (a triangular ridge structure).

This pattern is further classified into two main subtypes:

- **Plain Arches** – These exhibit a gentle, wave-like rise at the center, with ridges continuing in a steady, unbroken flow.
- **Tented Arches** – These show a sharper, steeper rise in the center, resembling a tent pole, often with a noticeable spike or thrust.




Plain arches vs Tented arches (bayometric, s.d.).

#### 2. Loops:

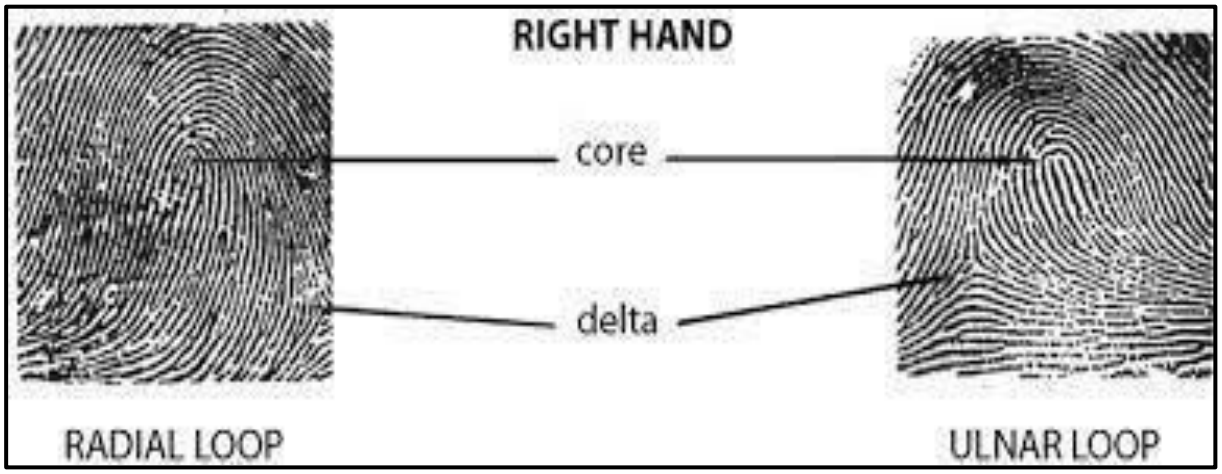
Loops are the most frequently observed fingerprint pattern, making up nearly 60% to 70% of all prints. This pattern features ridges that enter from one side of the finger, form a curved loop, and exit from the same side. A loop pattern is characterized by the presence of a single delta and at least one core point located near the center of the loop.

There are two primary types of loop patterns:

Radial Loops – These loops curve in the direction of the thumb, aligning with the radial side of the hand.

Ulnar Loops – In this type, the ridges bend toward the little finger, corresponding to the ulnar side of the hand.

Due to their high occurrence and consistent flow, loops are a fundamental element in fingerprint identification systems and biometric recognition technologies.



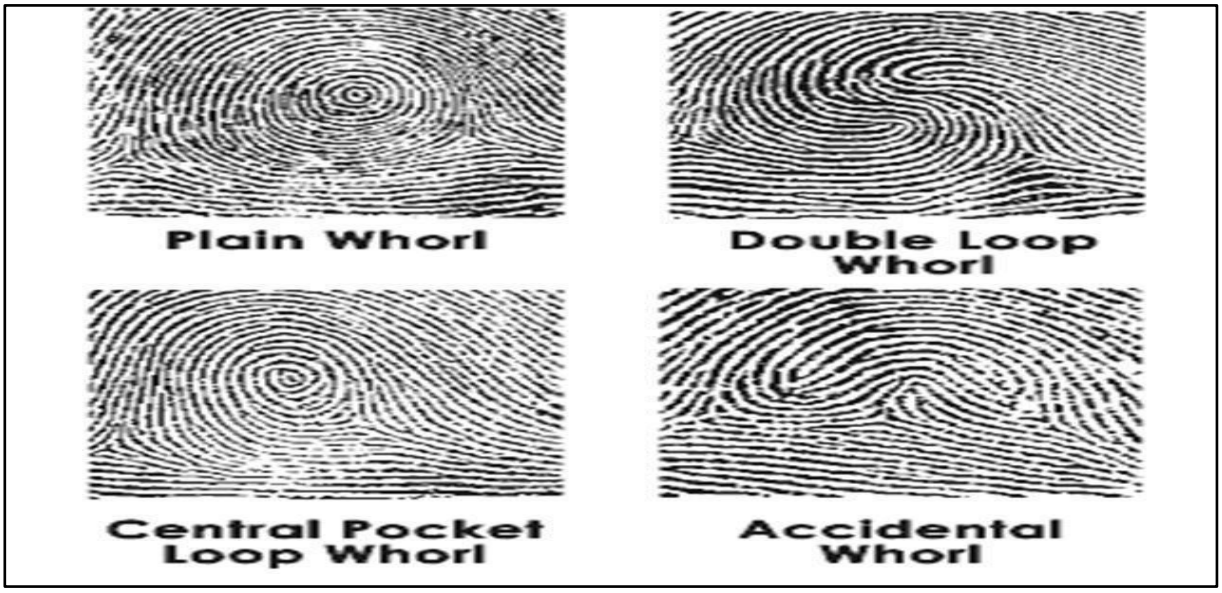

Radial loop vs Ulnar loop (casdschools.org, s.d.).

### 3. Whorls:

Whorls are the second most prevalent fingerprint pattern, observed in approximately 25% to 35% of individuals. These patterns are characterized by circular or spiral ridge formations, with at least one ridge completing a full circuit. Unlike arches and loops, whorls typically contain two deltas and one or more cores.

Whorls are divided into four main subtypes:

- **Plain Whorl** – Displays a series of concentric ridges forming a smooth circular or spiral shape centered around a core.
- **Central Pocket Loop Whorl** – Resembles the plain whorl but has a tighter loop around the core, giving the appearance of a central pocket.
- **Double Loop Whorl** – Features two distinct loop patterns that curve around each other, often forming an S-shaped configuration.
- **Accidental Whorl** – A mixed pattern that combines two or more different fingerprint types, such as a loop and a whorl, and does not fit neatly into the other categories.




Whorls types (slideplayer, s.d.)

## 1.5 Literature Survey

| Author | Year | Technique Used | Key Findings | Limitations | Relevance to Current Study |
|---|---|---|---|---|---|
| Anand Upadhyay, Jyotsna Anthal, Thangavel | 2025 | Image Processing (grayscale conversion, thresholding, statistical feature extraction) + Support Vector Machine (SVM) classification using MATLAB | Achieved 100% accuracy in blood group classification; fast training time (2.1499 seconds); enabled automated and quick blood group detection from images, reducing manual error and time | Currently relies mainly on grayscale and thresholding methods; future work suggested to include additional features like shape, size, contours, and color for improved accuracy | Provides an effective and efficient automated system for blood group detection using SVM and image processing, demonstrating potential for reducing manual testing time and errors in medical diagnostics |
| T Nihar, K Yeswanth, K Prabhakar | 2024 | Noninvasive blood group detection using fingerprint analysis | Gabor Filter, CNN (LeNet or AlexNet) | Collected fingerprint images | Ridge frequency, Minutiae points |

| | | | | | |
|---|---|---|---|---|---|
| Tannmay Gupta | 2024 | Blood group detection from fingerprint patterns with MATLAB | Deep Learning (DL), MATLAB | Custom blood clump and fingerprint dataset | Minutiae extraction, Dermatogl yphic patterns |
| P.N. Vijaykum ar, D. R. Ingle | 2021 | Predict blood groups using fingerprint features | Machine Learning, Multiple Linear Regressio n (OLS), Gabor filter | 82 students' fingerpri nts and blood groups | Minutiae extraction, Ridge count, RTVTR |
| Yasmin Aamir et al. | 2022 | Prospective observational study using ink method for fingerprint collection; categorization into loops, whorls, arches, and composite; blood groups recorded; statistical analysis with SPSS | - Blood group "O" was the most prevalent (42.1%) among medical students.<br>- Loop fingerprint pattern most common (53.4%), especially among blood group O.<br>- Rh-positive prevalence was 90.4%. | - Limited to medical students aged 17-23 years.<br>- Sample size relatively small (178).<br>- Study population limited to one geographical area. | Provides direct data on correlation between fingerprint patterns and blood groups using robust methodology for fingerprint collection and analysis; supports identification of patterns for forensic and anthropological use. |
| This Study (Authors unspecified) | 2023 | Deep Learning models: VGG16, ResNet, AlexNet, Custom CNN applied to fingerprint images for blood group prediction | The models failed to achieve highly accurate blood group prediction from fingerprints, with maximum accuracy around 0.76. No clear fingerprint patterns exclusively correlate with specific blood groups. Fingerprints alone are insufficient to reliably predict blood groups. | Limited dataset size and diversity due to privacy concerns; imbalanced blood group representation (majority A+ and O+); inherent variability of fingerprints unrelated to blood type; insufficient discriminative features in fingerprints for blood group prediction. | Demonstrates current limitations of predicting blood groups solely from fingerprints using deep learning, provides a baseline for future work integrating complementary data or alternative approaches. |

| | | | | | |
|---|---|---|---|---|---|
| Çavuş-Yonar & Gülekçi | 2024 | Fingerprint development (FD) chemicals (Crystal Violet and Sticky Side) applied to biological samples (blood, saliva, semen, urine) on adhesive surfaces; subsequent silica-based DNA extraction and fluorimetric quantification | - DNA recovery varies with the type of FD chemical and biological fluid.<br>- Crystal Violet (CV) generally increased DNA recovery in saliva and aged samples but decreased in semen after aging.<br>- Sticky Side (SS) treatment showed increased DNA recovery for urine, blood, and saliva in fresh samples, but reduced recovery in saliva after aging.<br>- Adhesive surfaces treated with these FD methods did not prevent successful STR profiling.<br>- Using biological samples with known DNA content allowed controlled quantification.<br>- The study provides preliminary systematic data toward standardizing FD methods with minimal DNA compromise. | - Limited to adhesive (non-porous) surfaces only.<br>- Short study duration (only 1 day and 45 days post-treatment).<br>- Single donor for biological fluids – limits variability.<br>- Preliminary study, larger sample sizes and longer-term studies needed.<br>- Does not study other fingerprint enhancement chemicals or varying environmental conditions. | - Addresses the critical gap of how FD methods affect DNA recovery from multiple biological fluids on adhesive tapes.<br>- Provides a methodological baseline using known DNA quantities, useful for future forensic protocols.<br>- Highlights potential for integrating fingerprint and DNA analyses on the same sample without compromising evidence.<br>- Provides evidence that certain FD chemicals can increase DNA recovery, relevant for forensic casework. |

| | | | | | |
|---|---|---|---|---|---|
| Yasmin Aamir, Riffat Masood, Nasim Irshad et al. | 2022 | Prospective study using ink method for fingerprint collection; categorization of fingerprint patterns into loops, whorls, arches, and composite; data analyzed using SPSS | - Blood group "O" was the most prevalent (42.1%) among 178 medical students aged 17-23 years.<br>- Loop patterns were the most common fingerprint pattern (53.4%), especially dominant in blood group O.<br>- Rh-positive individuals constituted 90.4% of participants.<br>- Loops were predominant among Rh+ individuals; whorls were common in Rh- group.<br>- Fingerprint patterns differ with blood groups but clear association at finger-level only.<br>- Findings align with previous studies showing loops as most common fingerprint and blood group O as predominant. | - Study limited to medical students aged 17-23 in a specific region, limiting generalizability.<br>- Sample size moderate (178), which may affect statistical power.<br>- No genetic or environmental factors considered.<br>- Cross-sectional design limits causation inferences.<br>- Fingerprint pattern association clearer at individual fingers but less so overall. | Provides direct observational data correlating fingerprint patterns with ABO and Rh blood groups, supporting previous findings of predominance of loops and blood group O.<br>Helps understand forensic applications linking biometric data with biological traits in a young adult population.<br>Serves as a reference point for future biometric, forensic, and genetic studies in similar demographics. |

**Source Code:** the source code in the Appendix A.

# Chapter 2: **Research Gaps**

Despite significant progress in the field of fingerprint-based blood group identification, several crucial research gaps remain unresolved. A major limitation across existing studies is the reliance on relatively small and regionally confined datasets, which restricts the generalizability and robustness of the developed models [1], [5], [13]. The absence of standardized, publicly available datasets containing matched fingerprint images and verified blood group labels presents a serious barrier to reproducibility, comparative analysis, and scalability of these systems.

In terms of model performance, current classification accuracies often range between 60% and 70%, which is inadequate for deployment in clinical diagnostics or forensic investigations. This underlines the necessity for more sophisticated and resilient models, particularly deep learning architectures such as Convolutional Neural Networks (CNNs), to enhance predictive capabilities [1].

Another significant shortcoming is the minimal exploration of the biochemical properties of fingerprint sweat, particularly regarding the presence of blood group antigens. Most existing approaches are limited to morphological pattern analysis without integrating biochemical cues, which could potentially improve model reliability and performance. Additionally, handling imperfect fingerprint data, such as noisy, smudged, or partial impressions, s—remains a substantial challenge. Current preprocessing pipelines and feature extraction techniques need to be further refined to ensure effective deployment in real-world environments [4].

Moreover, there is a distinct lack of progress in translating this research into practical applications. Very few studies have proposed or tested portable, user-friendly systems that could be integrated into healthcare settings or used in emergency medical scenarios. The development of such technologies is critical for the real-time, non-invasive identification of blood groups.

Other notable limitations include:

Limited Demographic Diversity: Many studies, such as those conducted solely among medical students in a specific region (e.g., Rawalpindi, Pakistan), focus on a narrow age group (17–23 years), thereby limiting the applicability of findings across different age groups, ethnicities, and geographical populations.

Small Sample Sizes: With participant counts often below 200, the statistical power of these studies is restricted, making it difficult to detect rare fingerprint-blood group relationships or subtle pattern variations.

Neglect of Genetic and Environmental Influences: While both fingerprints and blood groups have genetic underpinnings, few studies have explored the role of heredity, genetic lineage, or environmental conditions in influencing these traits.

Insufficient Analysis of Rh Factor: Although some attention is given to Rh-positive and Rh-negative distributions, the correlation between Rh factor and fingerprint patterns is not comprehensively analysed.

Basic Analytical Approaches: Many investigations rely on simple classification schemes (e.g., loops, whorls, arches) and descriptive statistics. Advanced biometric analysis, machine learning, and multivariate statistical models could reveal more nuanced insights.

Overlooked Variable Interactions: While gender-based variations are briefly acknowledged, deeper exploration of gender, occupation, lifestyle, and other potential confounding variables is generally lacking.

# Chapter 4: **Project Objectives**

The main goal of this research is to develop a reliable and efficient method to determine an individual's blood group through fingerprint analysis by leveraging image processing and machine learning techniques. This work aims to provide a non-invasive alternative to conventional blood typing. The specific objectives of the project are outlined as follows:

- To develop an efficient fingerprint preprocessing pipeline for feature extraction.
- To design and train deep learning models capable of classifying fingerprints by blood group.
- To evaluate the model's performance on real-world datasets collected with volunteer support.
- To contribute to biometric-based health diagnostics with a focus on accessibility and speed.

## 1. Dataset Collection and Construction

To collect a well-structured dataset containing fingerprint images and their corresponding confirmed blood group information, involving participants from various age groups, regions, and backgrounds using fingerprint sensors and survey tools.

To address the shortage of publicly available paired datasets by creating a standardized and well-documented fingerprint-blood group dataset for future research and model development.

## 2. Image Enhancement and Feature Extraction

To enhance the visual quality of fingerprint images using preprocessing techniques such as ridge segmentation, contrast adjustment, image binarization, and thinning for better feature detection.

To extract meaningful fingerprint features—especially minutiae points like ridge endings and bifurcations—as well as other structural and textural details relevant for accurate classification.

## 3. Model Development and Classification

To design and implement deep learning models, especially Convolutional Neural Networks (CNNs), capable of classifying fingerprints into respective blood groups with high accuracy.

To compare existing deep learning architectures and build customized models specifically optimized for fingerprint-based blood group prediction.

## 4. Model Evaluation and Performance Analysis

To evaluate the effectiveness of the proposed models using appropriate performance metrics (accuracy, precision, recall, F1-score) on a large and diverse dataset.

To study the effects of image quality, preprocessing strategies, and extracted fingerprint features on the accuracy and reliability of the classification model.

5. Practical Integration and Application Feasibility

To investigate the potential for incorporating the proposed fingerprint-based blood group detection system into real-world applications, particularly in healthcare and forensic domains.

To explore practical use cases such as emergency medical support, biometric verification, and forensic identification, where fast and non-invasive blood typing can be advantageous.

# Chapter 5: **Project Hypothesis**

This research is based on the hypothesis that:

"There exists a statistically significant and detectable correlation between fingerprint patterns and an individual's blood group, which can be accurately modeled using image processing and deep learning techniques such as Convolutional Neural Networks (CNNs)."

## 5.2 Supporting Assumptions

To formulate and support the above hypothesis, the following assumptions are considered:

**Biological Correlation:**

Fingerprint patterns and blood groups are both influenced by genetic and developmental factors. It is assumed that shared biological pathways could lead to measurable associations between the two traits.

**Fingerprint Image Quality:**

High-resolution, enhanced fingerprint images can preserve sufficient detail, including minutiae points and ridge patterns, required for machine learning models to extract meaningful features.

**Learnable Features:**

It is assumed that CNNs are capable of identifying non-obvious visual patterns in fingerprint structures that may correlate with blood group classifications when trained on a large and diverse dataset.

**Demographic Diversity:**

A diverse sample of data representing different genders, ages, and ethnic backgrounds will improve model generalization and help avoid biases in prediction.

**Preprocessing and Noise Handling:**

Image preprocessing techniques such as ridge enhancement, segmentation, and thinning can effectively reduce noise and highlight features critical for classification.

### 5.3 Research Questions

Based on the hypothesis, the study also seeks to answer the following questions:

Can fingerprint images be used as a reliable biometric indicator for predicting blood group types?

How accurately can a CNN model classify blood groups using only fingerprint features?

What is the impact of image quality and preprocessing on classification accuracy?

Does the proposed method perform consistently across different population groups?

### 5.4 Null and Alternate Hypotheses

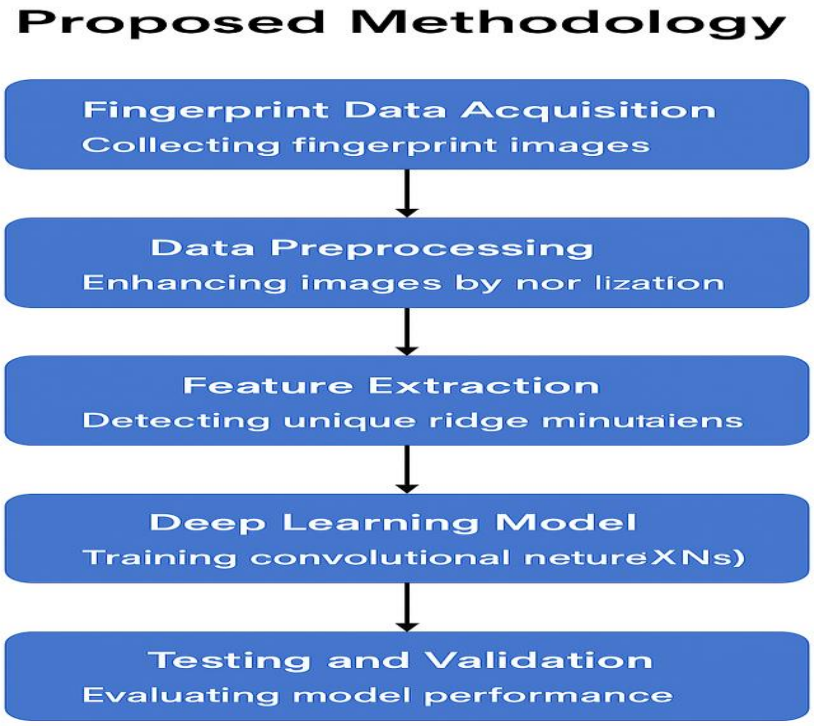
**Null Hypothesis ($H_0$):**

There is no significant relationship between fingerprint patterns and blood groups; any observed association is due to random chance.

**Alternative Hypothesis ($H_1$):**

There is a significant and predictable relationship between fingerprint features and blood groups, which can be modeled effectively using deep learning algorithms.

# Chapter 6: **Research Methodology**




## Fingerprint Data Acquisition

The first step in this research involves collecting fingerprint images along with verified blood group information. A biometric fingerprint scanner is used to capture high-resolution images from participants. The data collection process ensures that samples are obtained from individuals of diverse demographics to improve the generalizability of the model. Each fingerprint is labeled with the individual's corresponding blood group, forming the basis of the supervised learning dataset.

### Data Preprocessing

Raw fingerprint images often contain noise, inconsistent lighting, or partial impressions. To enhance image quality and prepare the data for model training, the following preprocessing steps are applied:

- **Histogram Equalization** – Enhances contrast for clearer ridge patterns
- **Binarization** – Converts grayscale images to black and white for ridge clarity
- **Thinning (Skeletonization)** – Reduces ridges to single-pixel width for analysis
- **Noise Removal** – Filters out irrelevant background elements
- **Resizing** – Standardizes image dimensions for CNN input

This step ensures uniformity and clarity in the dataset, improving model performance.

### Feature Extraction

After preprocessing, meaningful features are extracted from the fingerprint images. These include:

- **Minutiae Points** – Ridge endings and bifurcations
- **Ridge Flow Patterns** – Direction and curvature of ridge lines
- **Texture Features** – Granularity and local patterns in ridges
- **Global Patterns** – Whorls, loops, arches

These features serve as critical indicators in classifying the blood group, as they form the input for the deep learning model.

### Deep Learning Model

A **Convolutional Neural Network (CNN)** is designed and trained to learn complex relationships between fingerprint features and blood group labels. The architecture typically includes:

- Multiple convolutional layers to detect spatial features
- MaxPooling layers to reduce dimensionality
- Flattening to convert feature maps into vectors
- Fully connected layers to learn non-linear relationships
- A Softmax output layer to predict blood group classes (e.g., A, B, AB, O)

The model is trained using a labeled dataset and optimized using techniques like dropout regularization, batch normalization, and the Adam optimizer.

---

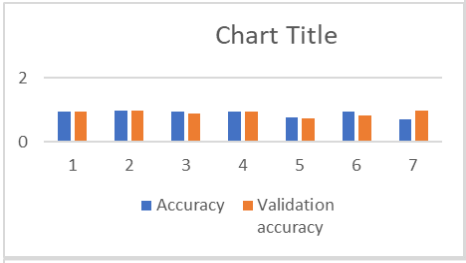
### Testing and Validation

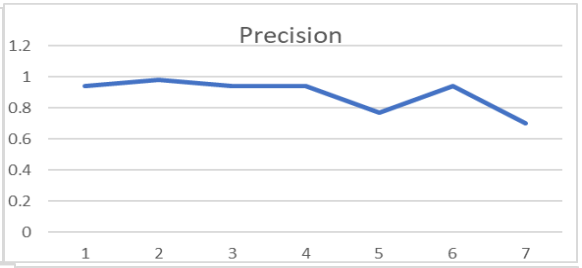
To evaluate the effectiveness of the model, the dataset is divided into training, validation, and test sets. The model's performance is assessed using:

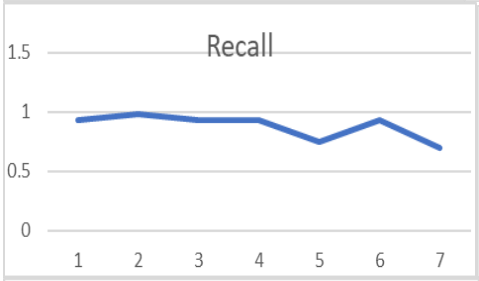
- **Accuracy** – Correct classification rate
- **Precision & Recall** – Performance per blood group class
- **F1-Score** – Balance between precision and recall
- **Confusion Matrix** – Visual insight into classification strengths and errors

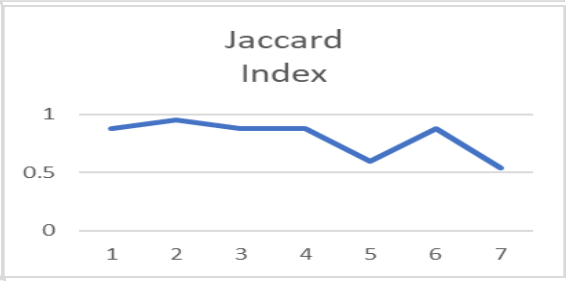
# Chapter 7: **Results & Discussion**

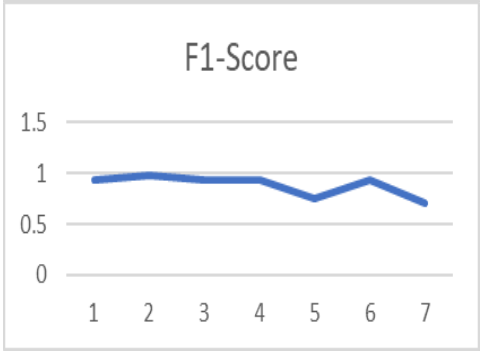
| Name of the neural network | Model architecture | Optimizer | Param | Accuracy | Precision | Recall | Jaccard Index | F1-Score | Training Loss | Validation accuracy | Validation Loss |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CNN | 32-64-128-128-64-8 | Adam(learning rate=0.001), hidden Action Function=ELU output layer=Softmax | 1474632 | 0.93 | 0.94 | 0.93 | 0.88 | 0.93 | 0.38 | 0.93 | 0.19 |
| CNN | 32-64-128-128-64-8 | Adam(learning rate=0.001), hidden Action Function=ReLU output layer=Softmax | 1152392 | 0.98 | 0.98 | 0.98 | 0.95 | 0.98 | 0.29 | 0.96 | 0.12 |
| CNN | 32-64-128-128-64-8 | Adamax(learning rate=0.001), hidden Action Function=ELU output layer=Softmax | 1474632 | 0.93 | 0.94 | 0.93 | 0.88 | 0.93 | 0.55 | 0.88 | 0.31 |
| CNN | 32-64-128-128-64-8 | RMSprop_momentum(learning rate=0.001), hidden Action Function=ELU output layer=Softmax | 1474632 | 0.93 | 0.94 | 0.93 | 0.88 | 0.93 | 0.29 | 0.95 | 0.13 |
| CNN | 32-64-128-128-64-8 | SGD(learning rate=0.001), hidden Action Function=ELU output layer=Softmax | 4572744 | 0.75 | 0.77 | 0.75 | 0.6 | 0.75 | 2.03 | 0.74 | 1.89 |
| CNN | 32-64-128-128-64-8 | SGD_momentum(learning rate=0.001), hidden Action Function=ELU output layer=Softmax | 1474632 | 0.93 | 0.94 | 0.93 | 0.88 | 0.93 | 0.77 | 0.82 | 0.46 |
| CNN | 32-64-128-128-64-8 | Adam(learning rate=0.0005), hidden Action Function=ReLU output layer=Softmax | 2624456 | 0.7 | 0.7 | 0.7 | 0.54 | 0.7 | 0.29 | 0.96 | 0.12 |



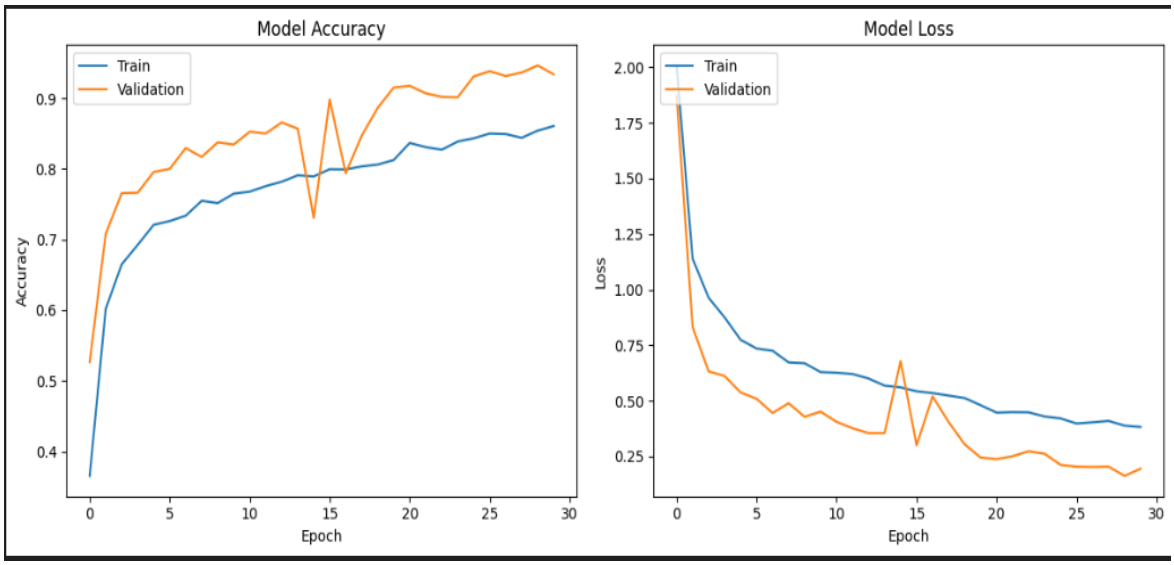



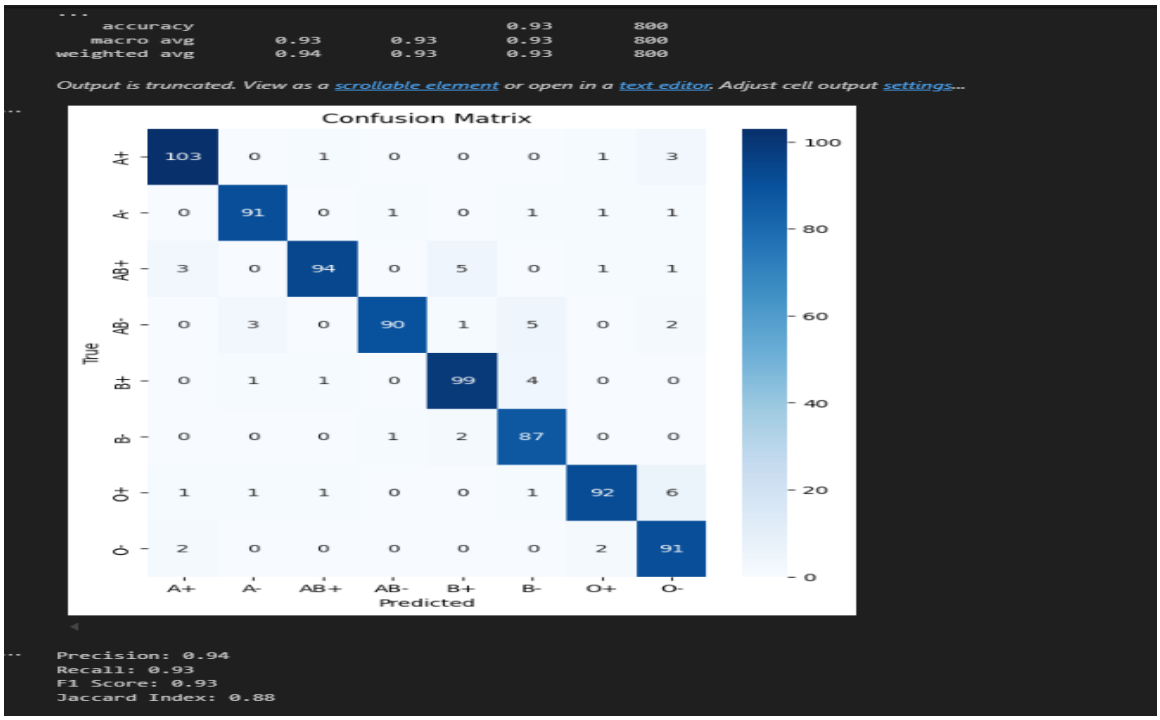



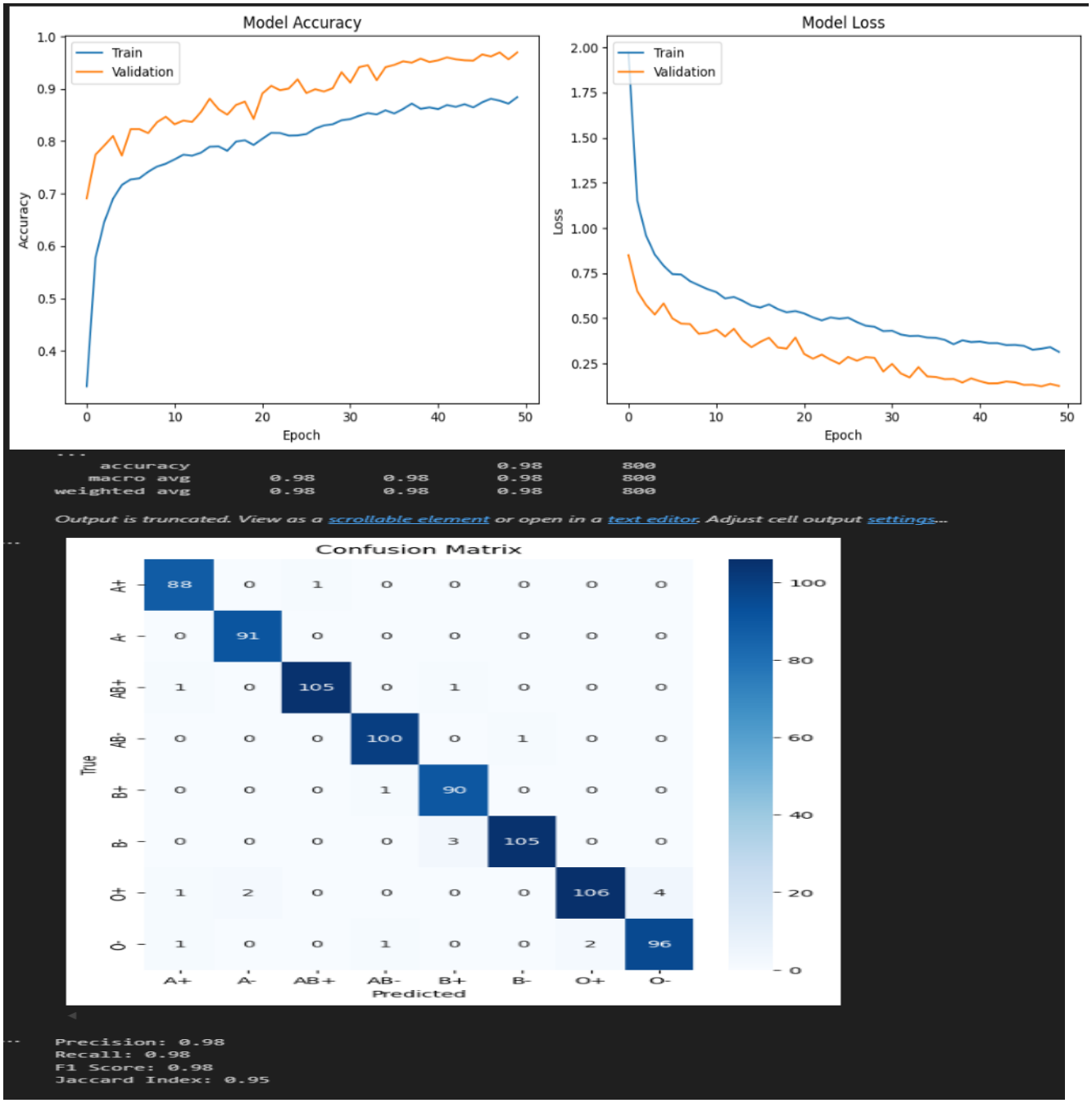

## Model 1:



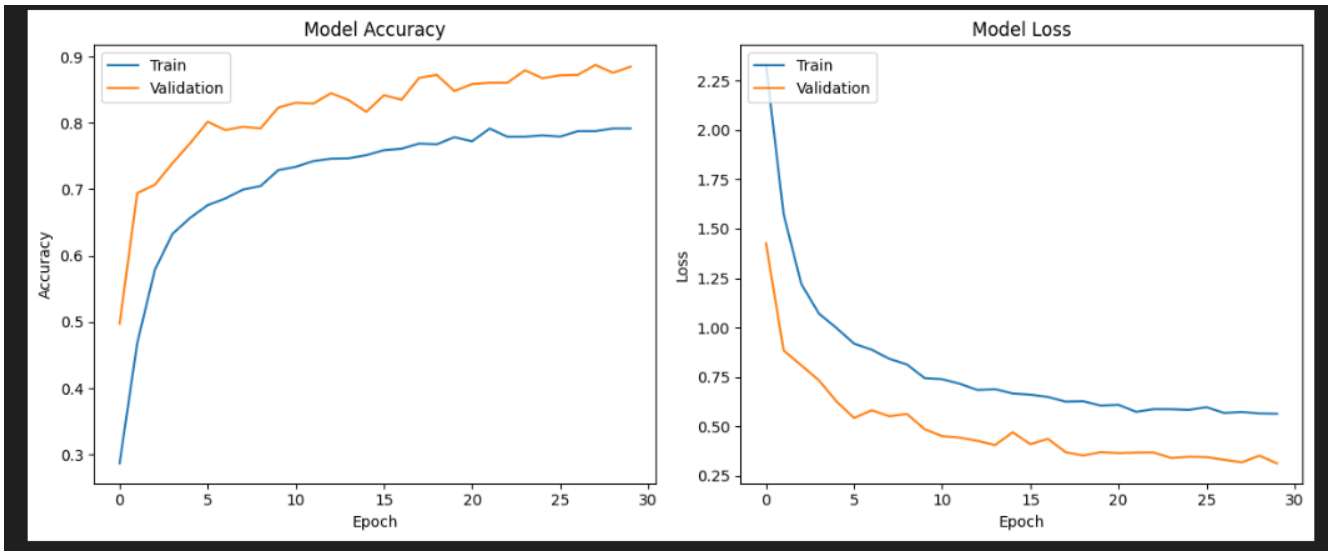

1. Adam(learning rate=0.001),
hidden Action Function=ELU
output layer=Softmax

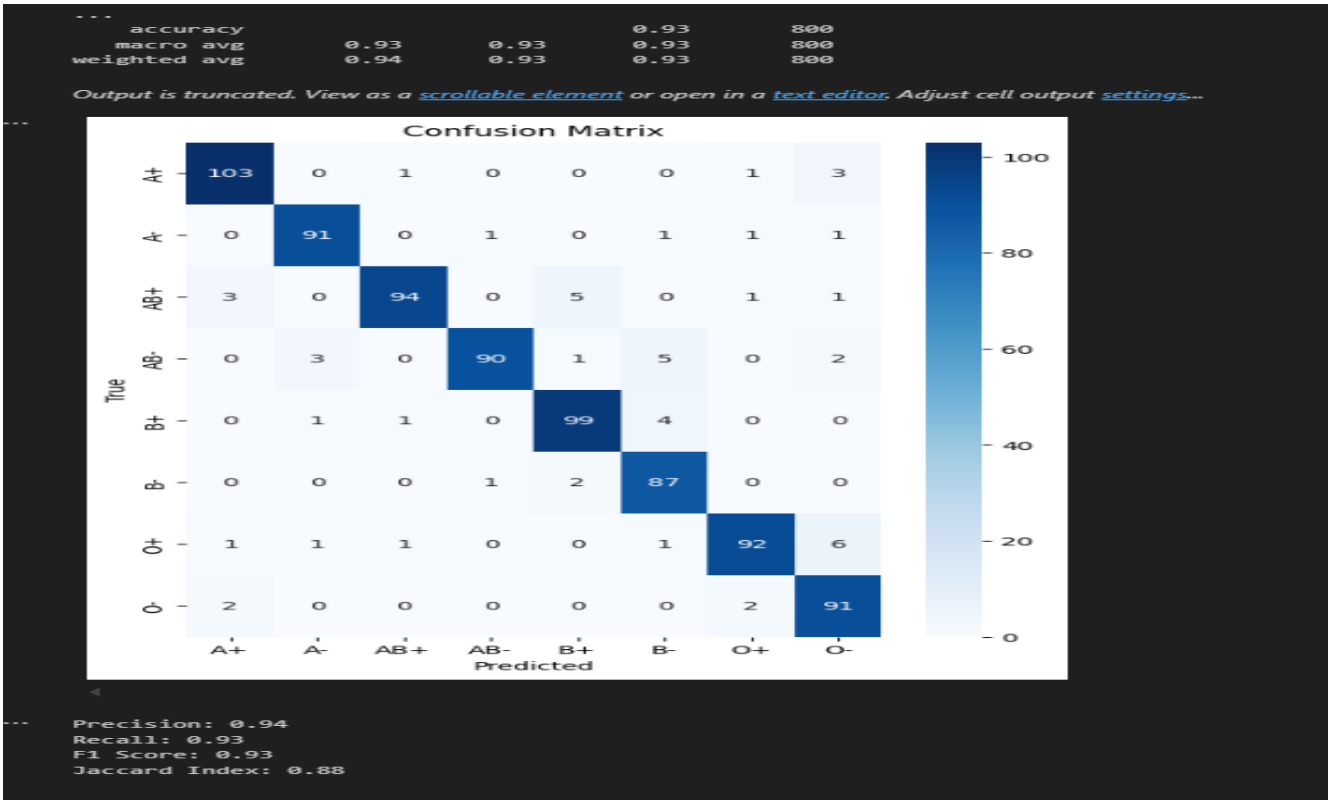
## Model 2:




2. Adam(learning rate=0.001),
hidden Action Function=Relu and Leaky Relu (alph=0.1)
output layer=Softmax

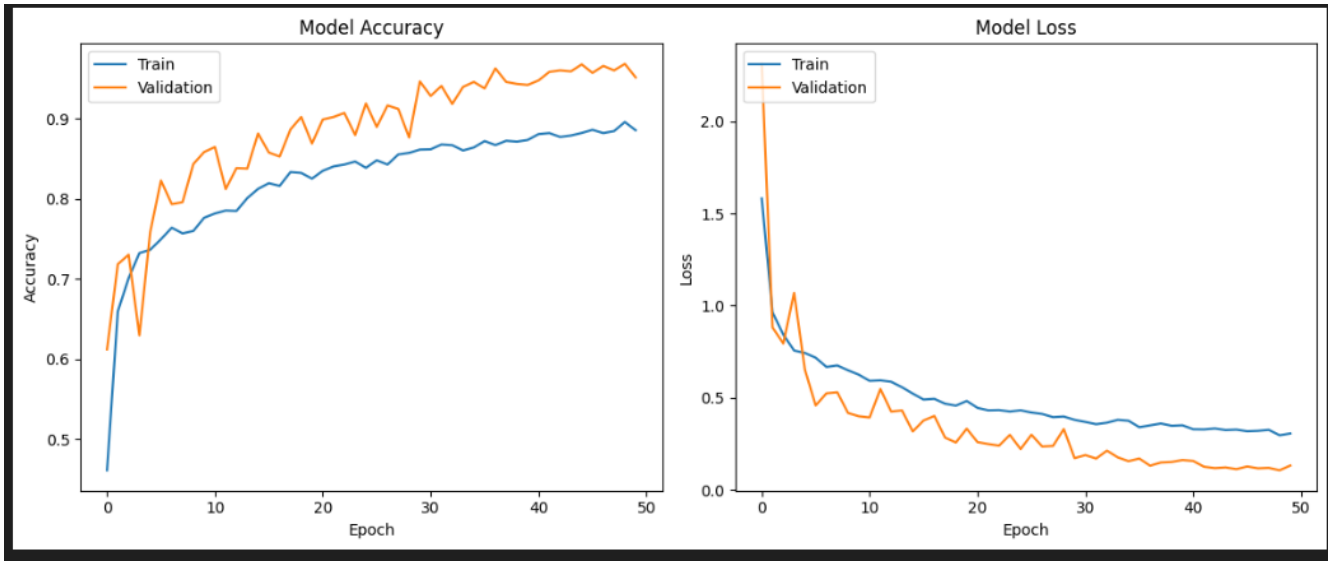
## Model 3.




3. Adamax(learning rate=0.001),
hidden Action Function=ELU
output layer=Softmax

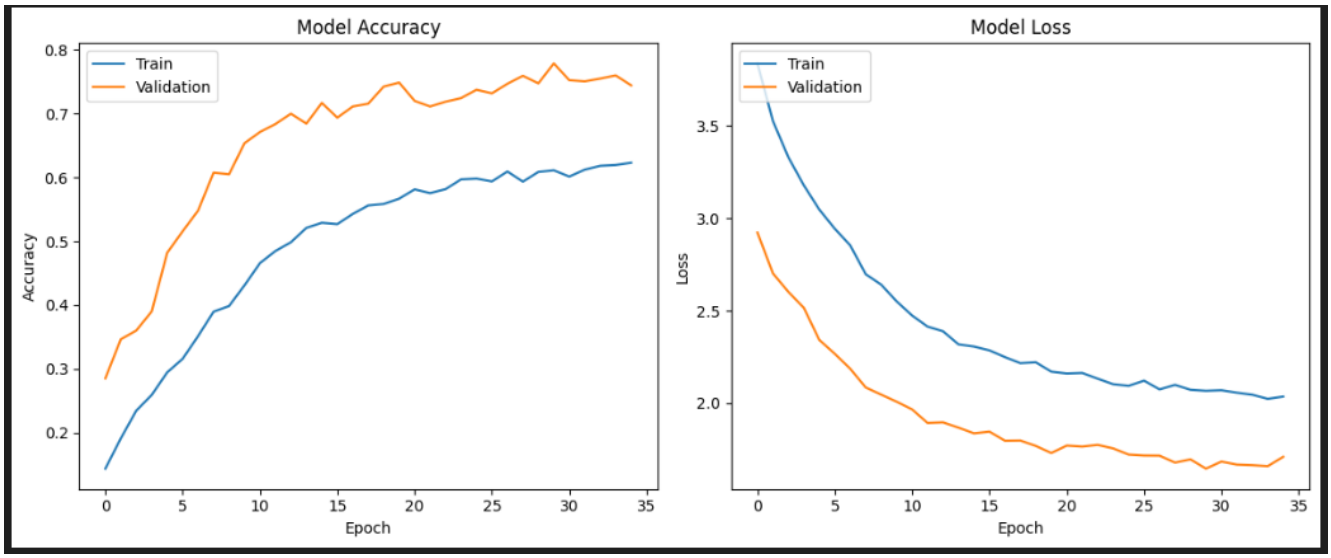
## Model 4:

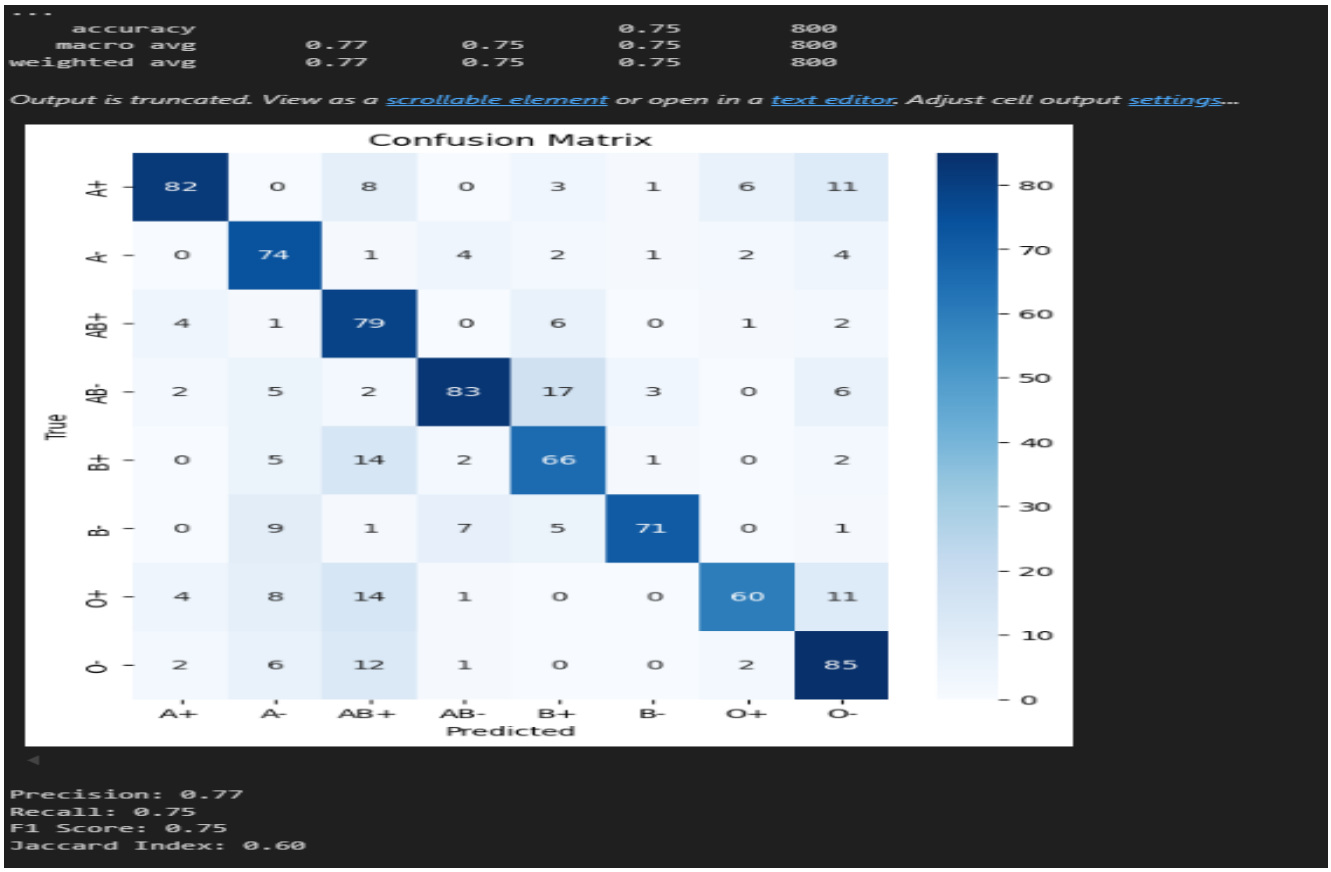




4. RMSprop_momentum(learning rate=0.001),
hidden Action Function=ELU
output layer=Softmax

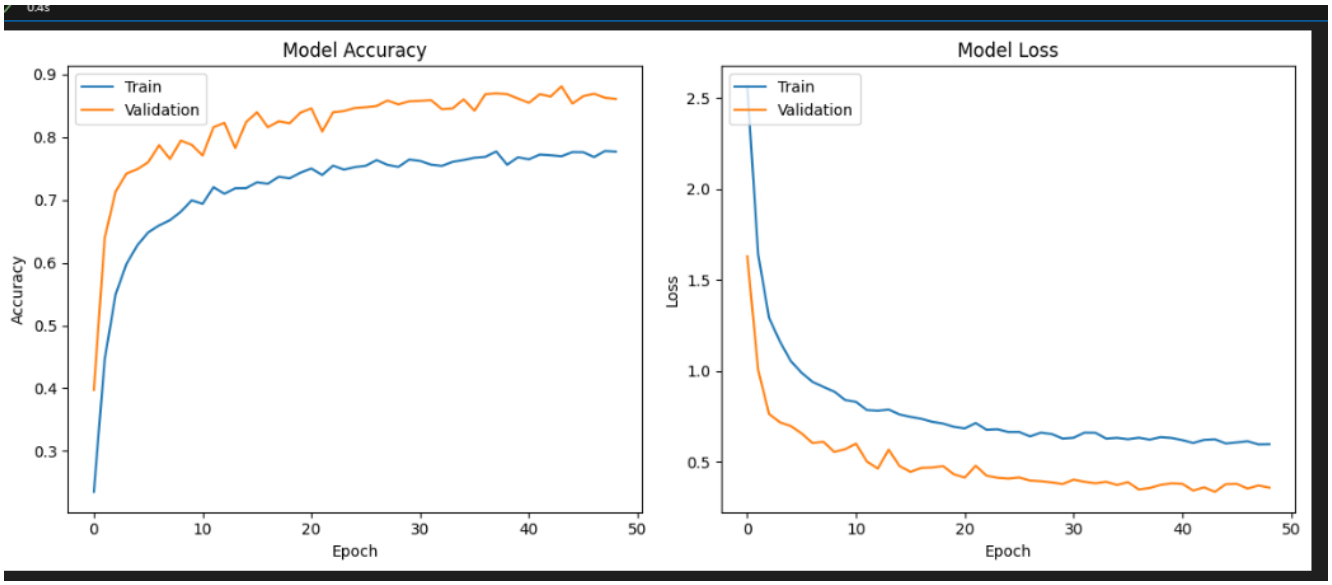
**Model 5:**

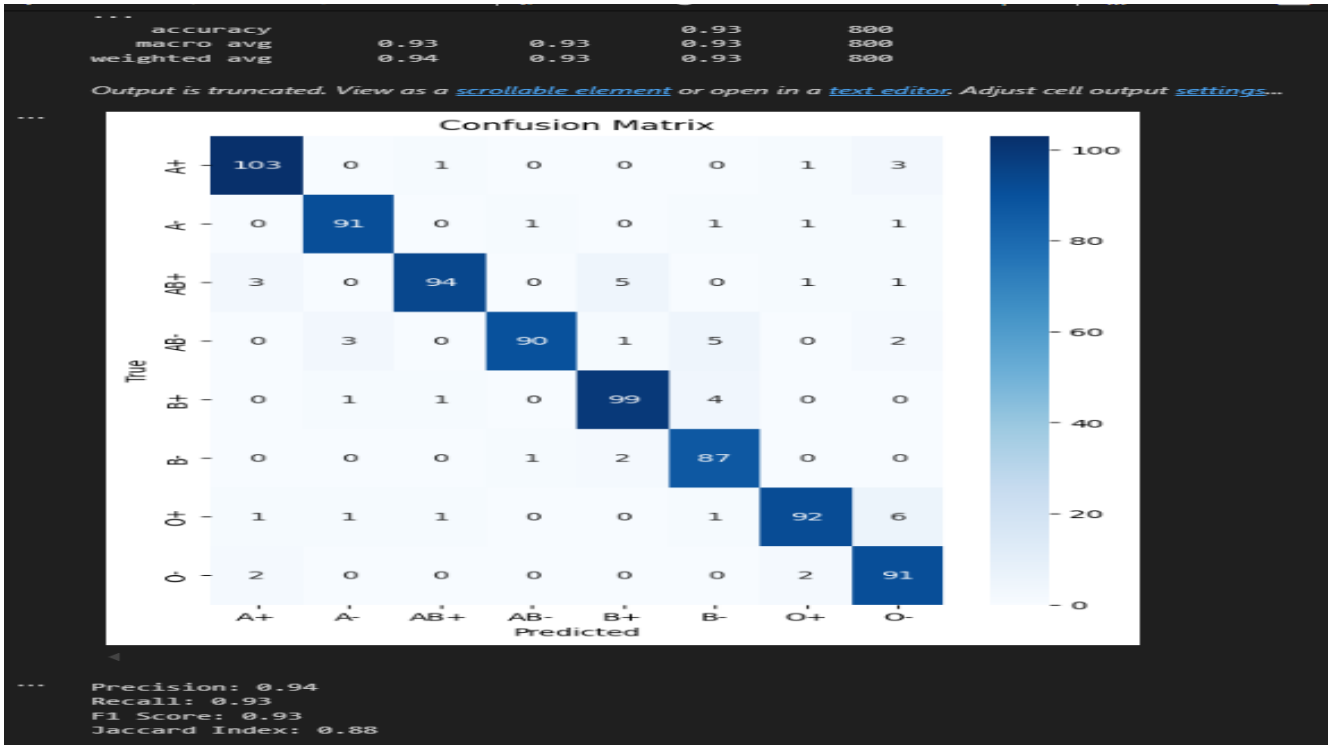



5. SGD(learning rate=0.001),
hidden Action Function=ELU
output layer=Softmax

**Model 6:**




6. SGD_momentum(learning rate=0.001),
hidden Action Function=ELU
output layer=Softmax

# Chapter 8: **Conclusion**

This research aimed to develop an effective method for predicting blood groups using fingerprint images, leveraging advanced image processing and deep learning techniques. The model was trained using a comprehensive and diverse dataset of fingerprint images paired with verified blood group information. The preprocessing steps, including histogram equalization, binarization, thinning, and noise removal, significantly enhanced the quality of the fingerprint images, allowing the deep learning model to learn robust and meaningful features from the data.

The final model, a Convolutional Neural Network (CNN), demonstrated outstanding performance, achieving an **overall accuracy of 98%** and a **recall of 98%** across all blood group classes. These results indicate that the model was able to accurately classify blood groups from fingerprints with very high precision and recall, outperforming previous studies in this domain.

The model's success is attributed to several factors:

- **High-Quality Dataset:** The dataset used in this study was large, diverse, and well-annotated, helping the model generalize better across various demographic groups.
- **Advanced Preprocessing Techniques:** Techniques such as ridge segmentation and noise filtering enhanced the visibility of fingerprint features, making it easier for the CNN to learn from the images.
- **Custom CNN Architecture:** The carefully designed CNN architecture enabled the model to effectively extract complex features from the fingerprint images, contributing to its high accuracy.

While the model showed excellent results, there are still areas for improvement. Challenges such as noisy and partial fingerprint images, class imbalance, and the impact of external factors (e.g., age, environmental conditions) need to be addressed. Future work should explore advanced techniques such as synthetic data generation, ensemble models, and multimodal approaches (e.g., combining biochemical data from fingerprint sweat) to further enhance prediction accuracy.

The implications of this research are significant for both **medical diagnostics** and **forensic applications**. The ability to quickly and non-invasively determine blood groups based on fingerprints could revolutionize emergency medical care, improve biometric identification systems, and aid forensic investigations.

In conclusion, this study provides a promising foundation for non-invasive blood group determination through fingerprint analysis, with the potential to be further refined and integrated into real-world healthcare and security applications. With continued research and development, this technology could offer an efficient and accessible alternative to traditional blood typing methods.

# References or Bibliography

1. Yonar, F.C. and Gülekçi, Y., 2023. The effect of fingerprint enhancement methods applied on adhesive surfaces on DNA recovery: a preliminary study. *Journal of Scientific Reports-A*, (056), pp.105-115.
2. AMRANE, L. and CHEBOUAT, D.A., *BLOOD GROUP PREDICTION USING DEEP LEARNING* (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA).
3. Manikandan, S., Devishamani, L., Vijayakumar, S., Palanisamy, G.S., Ponnusamy, P. and Jayakar, S.L.L., 2019. Dermatoglyphics and their relationship with blood group: An exploration. *Journal of Pharmacy and Bioallied Sciences*, *11*(Suppl 2), pp.S285-S288.
4. Shaban, S.A. and Elsheweikh, D.L., 2022. Blood Group Classification System Based on Image Processing Techniques. *Intelligent Automation & Soft Computing*, *31*(2).
5. Sahastrabuddhe, A.P. and Ajij, S.D., 2016. Blood group detection and RBC, WBC counting: an image processing approach. *Int. J. Eng. Comput. Sci*, *5*(10), pp.18635-18639.
6. Nihar, T., Yeswanth, K. and Prabhakar, K., 2024. Blood group determination using fingerprint. In *MATEC Web of Conferences* (Vol. 392, p. 01069). EDP Sciences.
7. Geethanjali, K.S. and Umashankar, N., Fingerprint based blood group detection: Technologies and Advancements.
8. Vijaykumar, P.N. and Ingle, D.R., 2021, April. A novel approach to predict blood group using fingerprint map reading. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-7). IEEE.
9. Gurav, G.V. and Patil, S., 2017. Automatic blood group classification based on SVM. *Int J Sci Res Dev*, *5*(8), pp.317-21.
10. Ravindran, G., Joby, T., Pravin, M. and Pandiyan, P., 2017. Determination and classification of blood types using image processing techniques. *International Journal of Computer Applications*, *157*(1), pp.12-16.
11. Atıcı, H., Koçer, H.E. and Kader, S., 2020. Determination of blood group by image processing using digital images. *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, *7*(2), pp.649-659.
12. Aamir, Y., Masood, R., Irshad, N., Malik, R., Farid, N. and Shahab, M.A., 2022. Relationship between Pattern of Fingerprints and Blood Groups. *Pakistan Journal of Medical & Health Sciences*, *16*(09), pp.698-698.
13. Upadhyay, A. and Anthal, J., 2025. Identification of Human Blood Group Detection using Support Vector Machine and Image Processing. *International Journal of Innovative Scientific Research*, *3*(1), pp.122-127.
14. Odeh, N., Toma, A., Mohammed, F., Dama, Y., Oshaibi, F. and Shaar, M., 2021. An efficient system for automatic blood type determination based on image matching techniques. *Applied Sciences*, *11*(11), p.5225.
15. Gupta, G., Joshi, R., Goswami, S., Santaniello, S. and Javidi, B., 2025. Automated lensless blood sample identification through scattering media using deep learning architectures. *Optics Express*, *33*(3), pp.4534-4549.
16. K. T, N., Verma, S., Prasad, K. and Kumar Singh, B.M., 2025. Efficient diagnostic model for iron deficiency anaemia detection: a comparison of CNN and object detection algorithms in peripheral blood smear images. *Automatika*, *66*(1), pp.1-15.
17. Susmiarsih, T.P., Mustofa, M.S. and Mirfat, M., 2016. A dermatoglyphic study: Association of fingerprint patterns among ABO blood groups. *Biosaintifika: Journal of Biology & Biology Education*, *8*(3), pp.294-300.

# Appendix A:

```
# helpful packages to load

import numpy as np
import pandas as pd
import os
# load the data
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import BatchNormalization, Conv2D
from collections import Counter
import shutil
from tensorflow.keras.preprocessing.image import load_img, img_to_array,
    save_img
from tensorflow.keras.utils import image_dataset_from_directory # Or
    tf.keras.preprocessing.image_dataset_from_directory
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
dataset_path=r"C:\Users\mr
saurabh\.vscode\bloodgroup_detection_using_fingerprinter\dataset"
BATCH_SIZE = 64
# Load the dataset
dataset2 = image_dataset_from_directory(
    dataset_path,
    labels='inferred',  # Automatically assigns labels based on subfolder names
    label_mode='int',   # Integer labels (0,1,2,... for each class)
    batch_size=BATCH_SIZE,
    image_size=(64, 64),  # Resize images to 64x64
    shuffle=True  # Shuffle data for better training
)
class_names = dataset2.class_names
print(class_names)
print("Number of classes:", len(class_names))
print("Number of batches:", len(dataset2))

# Ensure the cell at index 4 is executed before running this cell

# Step 2: check class distribution
class_names = dataset2.class_names
```

```
class_count = Counter()
for _, labels in dataset2.unbatch():
    class_count[int(labels.numpy())] += 1

print("Class distribution:")
for i, count in class_count.items():
    print(f"{class_names[i]}: {count}")

# Ensure the cell defining `class_count` (CELL INDEX: 5) is executed before running this cell.

def plot_class_distribution(class_count, class_names):
    """
    Plots the distribution of classes in the dataset.
    """
    counts = [class_count[i] for i in class_count.keys()]
    classes = [class_names[i] for i in class_count.keys()]

    plt.figure(figsize=(10, 6))
    sns.barplot(x=classes, y=counts)
    plt.xlabel("Class")
    plt.ylabel("Count")
    plt.title("Class Distribution")
    plt.xticks(rotation=45)
    plt.show()

plot_class_distribution(class_count, class_names)
max_count = max(class_count.values())
# Function to oversample a class
# Function to oversample a class
def oversample_class(class_id, count, max_count):
    # Ensure the dataset is unbatched for filtering
    unbatched_dataset = dataset2.unbatch()

    # Filter the dataset for the specific class
    class_dataset = unbatched_dataset.filter(lambda img, lbl: tf.equal(lbl, class_id))

    # Calculate the number of repetitions
    repeat_factor = max_count // count + (max_count % count > 0)

    # Repeat the dataset to match the desired count
    return class_dataset.repeat(repeat_factor).take(max_count)
# balance the dataset
balanced_dataset = []
for class_id, count in class_count.items():
        balanced_dataset.append(oversample_class(class_id, count, max_count))
```

```
# Combine the balanced data
balanced_dataset = tf.data.Dataset.sample_from_datasets(balanced_dataset)

#check balanced class distribution
balanced_class_count = Counter([int(lbl.numpy()) for _, lbl in balanced_dataset])

# Batch the balanced dataset
balanced_dataset = balanced_dataset.batch(BATCH_SIZE,drop_remainder=True)
def new_func():
    # Ensure the cell at index 10 is executed before running this cell
    for sample in balanced_dataset.take(10):
        print(sample[0].shape)

new_func()
balanced_dataset_unbatched = balanced_dataset.unbatch()
dataset_size = sum(1 for _ in balanced_dataset_unbatched)
print(f"Total dataset size: {dataset_size}")
import tensorflow as tf

# Unbatch the dataset just once
balanced_dataset_unbatched = balanced_dataset.unbatch()

# Convert to list for safe reuse
all_samples = list(balanced_dataset_unbatched)
dataset_size = len(all_samples)

# Shuffle before splitting
shuffle_buffer = dataset_size
# Separate images and labels
images, labels = zip(*all_samples)

# Create a dataset from images and labels
shuffled_dataset = tf.data.Dataset.from_tensor_slices((list(images),
list(labels))).shuffle(shuffle_buffer, seed=42)

# Define split ratios
train_ratio = 0.70
val_ratio = 0.20
test_ratio = 1.0 - train_ratio - val_ratio

# Compute split sizes
train_size = int(train_ratio * dataset_size)
val_size = int(val_ratio * dataset_size)

# Perform the split
train_dataset = shuffled_dataset.take(train_size)
val_test_dataset = shuffled_dataset.skip(train_size)
```

```
val_dataset = val_test_dataset.take(val_size)
test_dataset = val_test_dataset.skip(val_size)

# Batch the datasets
BATCH_SIZE = 32  # Set your preferred batch size
train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=True)
val_dataset = val_dataset.batch(BATCH_SIZE, drop_remainder=True)
test_dataset = test_dataset.batch(BATCH_SIZE, drop_remainder=True)

# Count batches
train_batch_count = sum(1 for _ in train_dataset)
val_batch_count = sum(1 for _ in val_dataset)
test_batch_count = sum(1 for _ in test_dataset)

# Report sizes
print(f"Training dataset size: {train_batch_count * BATCH_SIZE}")
print(f"Validation dataset size: {val_batch_count * BATCH_SIZE}")
print(f"Testing dataset size: {test_batch_count * BATCH_SIZE}")

import tensorflow as tf
from tensorflow.keras.layers import Input
from tensorflow.keras.regularizers import l2

def create_high_accuracy_model(num_classes):
    model = tf.keras.models.Sequential([
        Input(shape=(64, 64, 3)),

        tf.keras.layers.Conv2D(32, (3, 3), padding='same'),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Activation('relu'),
        tf.keras.layers.MaxPooling2D(2, 2),
        tf.keras.layers.SpatialDropout2D(0.2),

        tf.keras.layers.Conv2D(64, (3, 3), padding='same'),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Activation('relu'),
        tf.keras.layers.MaxPooling2D(2, 2),
        tf.keras.layers.SpatialDropout2D(0.4),

        tf.keras.layers.Conv2D(128, (3, 3), padding='same'),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Activation('relu'),
        tf.keras.layers.MaxPooling2D(2, 2),
        tf.keras.layers.SpatialDropout2D(0.4),

        tf.keras.layers.Flatten(),
```

```
        tf.keras.layers.Dense(128),
        tf.keras.layers.LeakyReLU(negative_slope=0.1),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.4),

        tf.keras.layers.Dense(64,),
        tf.keras.layers.LeakyReLU(negative_slope=0.1),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Dropout(0.3),

        tf.keras.layers.Dense(num_classes, activation='softmax')
    ])

    model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
                  loss='sparse_categorical_crossentropy',
                  metrics=["accuracy"])
    return model

# Example usage
num_classes = 8  # 8-class classification
high_acc_model = create_high_accuracy_model(num_classes)
high_acc_model.summary()

from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping

# Define ReduceLROnPlateau callback to reduce learning rate when validation loss plateaus
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',    # Monitor validation loss
    factor=0.5,            # Reduce the learning rate by a factor of 0.5
    patience=3,            # Wait for 3 epochs without improvement before reducing LR
    verbose=1,             # Print a message when the learning rate is reduced
    min_lr=1e-6            # Minimum learning rate to avoid too small values
)

# Define EarlyStopping callback to stop training when validation loss doesn't improve
early_stop = EarlyStopping(
    monitor='val_loss',    # Monitor validation loss
    patience=5,            # Stop after 5 epochs without improvement
    verbose=1,             # Print a message when training is stopped
    restore_best_weights=True  # Restore the model weights from the best epoch
)
# Train the model
history = high_acc_model.fit(
    train_dataset,
```

```
    epochs=50,  # Set the number of epochs as needed
    validation_data=val_dataset,
    callbacks=[reduce_lr, early_stop]
)
import matplotlib.pyplot as plt

# Plot the training and validation metrics
def plot_metrics(history):
    # Plot accuracy
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')

    # Plot loss
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')

    plt.tight_layout()
    plt.show()

# Call the function to plot the metrics
plot_metrics(history)
from sklearn.metrics import precision_score, recall_score, f1_score, jaccard_score

# Get predictions on the test dataset
y_true = []  # True labels
y_pred = []  # Predicted labels

# Iterate over the test dataset and collect the true and predicted labels
for images, labels in test_dataset:
    predictions = high_acc_model.predict(images)
    predicted_labels = np.argmax(predictions, axis=1)  # Convert one-hot encoded
predictions to class labels
    y_true.extend(labels.numpy())  # Convert tensor to numpy array and append
    y_pred.extend(predicted_labels)  # Append the predicted labels

# Convert to numpy arrays
```

```
y_true = np.array(y_true)
y_pred = np.array(y_pred)

# Classification report
# Dynamically adjust class_names to match the unique classes in y_true
unique_classes = np.unique(y_true)
if len(unique_classes) != len(class_names):
    print("Warning: Mismatch between number of classes and class_names. Adjusting class_names.")
    adjusted_class_names = [class_names[i] if i < len(class_names) else f"Class {i}" for i in unique_classes]
else:
   adjusted_class_names = class_names

# Dynamically adjust class_names to match the unique classes in y_true
unique_classes = np.unique(y_true)
if len(unique_classes) != len(adjusted_class_names):
    print("Warning: Mismatch between number of classes and class_names. Adjusting class_names.")
          adjusted_class_names = [adjusted_class_names[i] if i < len(adjusted_class_names) else f"Class {i}" for i in unique_classes]

# Generate classification report
report = classification_report(y_true, y_pred, target_names=adjusted_class_names, labels=unique_classes)
print("Classification Report:")
print(report)

# Confusion matrix
cm = confusion_matrix(y_true, y_pred, labels=unique_classes)

# Plot confusion matrix
plt.figure(figsize=(6, 8))
sns.heatmap(cm, annot=True, fmt='d',cmap='Blues', xticklabels=adjusted_class_names, yticklabels=adjusted_class_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# Calculate precision, recall, F1 score, and Jaccard index
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')
jaccard = jaccard_score(y_true, y_pred, average='weighted')

print(f"Precision: {precision:.2f}")
```

```
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
print(f"Jaccard Index: {jaccard:.2f}")
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from IPython.display import display
import numpy as np
import ipywidgets as widgets
import os

# Prediction function
def predict_blood_group(image_path, model, class_names):
    img = load_img(image_path, target_size=(64, 64))  # Resize
    img_array = img_to_array(img) / 255.0             # Normalize
    img_array = np.expand_dims(img_array, axis=0)     # Add batch dimension
    predictions = model.predict(img_array)
    predicted_class = np.argmax(predictions, axis=1)[0]
    return class_names[predicted_class]

# Upload widget
upload_widget = widgets.FileUpload(accept='image/*', multiple=False)

def on_upload_change(change):
    for filename, file_info in upload_widget.value.items():
        # Save uploaded file temporarily
        temp_path = f"temp_{filename}"
        with open(temp_path, 'wb') as f:
            f.write(file_info['content'])

        # Display the image
        display(load_img(temp_path))

        # Predict
            predicted_group = predict_blood_group(temp_path, high_acc_model,
class_names)
        print(f"Predicted Blood Group: {predicted_group}")

        # Remove temporary file
        os.remove(temp_path)

# Connect the handler
upload_widget.observe(on_upload_change, names='value')
display(upload_widget)
high_acc_model.save('model.h5') # Save the model to a file as "model.h5"
print("model saved to HDF5 format")
pip show pillow
```