AUTOMATED BLOOD GROUP DETECTION

Mini Project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

G. Santhoshi
Assistant Professor

By

DASARI ABHAYA MANASA (21R11A05C0)
NELLURU KUSUMA (21R11A05D8)

Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.
August-2024

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Accredited by NBA

CERTIFICATE

This is to certify that the B.Tech Mini Project report entitled **"AUTOMATED BLOOD GROUP DETECTION"** is a bonafide work done by **DASARI ABHAYA MANASA (21R11A05C0), NELLURU KUSUMA (21R11A05D8)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2023-2024.

Internal Guide	HOD - CSE
G.Santhoshi	**Dr A SreeLakshmi**
Assistant Professor	Professor

External Examiner

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA

DECLARATION BY THE CANDIDATE

We, **DASARI ABHAYA MANASA, NELLURU KUSUMA**, bearing Roll Nos. **21R11A05C0, 21R11A05D8**, hereby declare that the project report entitled **"AUTOMATED BLOOD GROUP DETECTION"** is done under the guidance of **Ms. G.Santhoshi**, **Assistant Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

DASARI ABHAYA MANASA(21R11A05C0)
NELLURU KUSUMA(21R11A05D8)
Department of CSE,
Geethanjali College of Engineering and Technology, Cheeryal.

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory. We would like to express our sincere gratitude to the following individuals for their invaluable contributions and support throughout this project. Firstly, we sincerely acknowledge our gratitude to **Sri. Ravinder Reddy Garu**, Chairman, Geethanjali Colllege of Engineering and Technology(GCET), for creating proactive environment for holistic development of students. We sincerely thank **Dr. Udaya Kumar Susarla**, Principal, GCET, for introducing multi skill development programs as part of curriculum. We would like to thank **Dr. A. Sree Lakshmi**, CSE Head Of Department, GCET, for constant support in completion of the project. A sincere thanks to our project guide, **Ms. G. Santhoshi**, Assisstant Professor, GCET, for regular follow ups and guiding us throughout the project period. We extend our appreciation towards our project coordinator, **Mr. S. Durga Prasad ,** Sr. Assistant Professor, GCET, for helping in successful completion of the internship.

We are indebted to all these individuals for their support, expertise, and guidance, without which this project would not have been possible.

Dasari Abhaya Manasa(21R11A05C0)

Nelluru Kusuma(21R11A05D8)

ABSTRACT

Determining blood type is essential including in emergency situation. Currently, these tests are performed manually by technicians, which can lead to human errors. Various systems have been developed to automate these tests, but none is able to perform the analysis in time for emergency situations. The project aims to address the need for efficient and accurate blood typing methods in healthcare settings. The project utilizes pre-acquired palm images obtained through sensors. These images serve as inputs for the proposed system. Initially, the images undergo preprocessing to enhance quality and remove noise, ensuring optimal analysis. Feature extraction methods are then applied to discern distinctive patterns associated with different blood groups from the palm images. Subsequently, machine learning algorithms such as support vector machines (SVM) Convolution Neural Networks (CNN) are trained on a dataset comprising labeled palm images and corresponding blood groups. The proposed system offers several advantages over manual blood typing methods, including speed, objectivity, and scalability. Moreover, its non-invasive nature reduces the risk of contamination and ensures patient safety. This mini project contributes to advancing healthcare technologies, with potential applications in blood transfusion services, clinical laboratories, and emergency medical care.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

1.1. ABOUT THE PROJECT

The "Automated Blood Group Detection" project presents a novel approach to determining an individual's blood group using fingerprint analysis. Traditional methods of blood group testing involve drawing blood and performing a series of laboratory tests, which, while accurate, can be time-consuming, invasive, and dependent on specialized equipment and trained personnel. In contrast, this project harnesses the power of machine learning to offer a non-invasive, quick, and efficient alternative that relies on the unique patterns found in a person’s fingerprints.

This project utilizes a dataset of fingerprint images, each associated with a known blood group, to train a machine learning model capable of predicting the blood group based on the fingerprint pattern. The underlying idea is rooted in the observation that certain dermatoglyphic patterns in fingerprints can be correlated with specific blood groups. By leveraging a Convolutional Neural Network (CNN), a type of deep learning model particularly well-suited for image recognition tasks, the system learns to identify these patterns and make accurate predictions. The impact of such a system could be profound, particularly in medical and emergency settings. Additionally, in remote areas or regions with limited access to healthcare facilities, where traditional blood testing might not be readily available, this fingerprint-based method could serve as a practical and accessible alternative. It could also reduce the need for invasive blood draws, making the process more comfortable for patients, particularly for those who are averse to needles.

The non-invasive nature of this method could open up new possibilities in preventive healthcare and routine medical check-ups, where quick and painless blood group detection could become a standard practice. The combination of machine learning and biometric analysis in this project not only demonstrates the potential of AI in healthcare but also lays the groundwork for future innovations that could transform the way we approach medical diagnostics.

1.2. OBJECTIVES

• Develop a Non-Invasive Blood Group Detection System: Create a system that determines an individual's blood group using fingerprint images, avoiding the need for blood sample collection.

• Utilize Machine Learning Techniques: Employ Convolutional Neural Networks (CNNs) to analyse fingerprint patterns for accurate blood group prediction.

• Enhance Speed of Blood Group Testing: Provide a rapid alternative to traditional blood group testing methods, crucial for emergency scenarios.

• Increase Accessibility of Blood Group Testing: Develop a method that can be used in remote or resource-limited areas where traditional blood testing facilities are unavailable.

• Improve Patient Comfort: Offer a non-invasive and needle-free method for blood group detection, reducing discomfort for patients.

• Ensure High Accuracy in Predictions: Train the model on a diverse and extensive dataset to achieve high accuracy in blood group prediction.

• Validate Model Performance: Test and validate the model's predictions using separate datasets to ensure reliability and robustness.

• Minimize Human Error: Automate the blood group detection process to reduce the potential for human error inherent in manual testing methods.

• Optimize the Model for Real-Time Applications: Develop the system to operate efficiently in real-time, allowing immediate blood group determination.

• Expand the Dataset for Broader Applicability: Continuously gather diverse fingerprint samples to improve the model's generalizability across different populations.

2. SYSTEM ANALYSIS

2.1. EXISTING SYSTEM

In the current healthcare environment, blood group detection is typically performed manually by laboratory technicians. This involves a process where blood samples are mixed with specific antibodies to observe reactions that indicate the blood group. Although automated systems exist, they are primarily designed for laboratory settings and may not provide results quickly enough for emergency situations. These systems generally require complex equipment and are dependent on the availability of blood samples, making them impractical in certain scenarios like remote locations or when rapid testing is required.

2.1.1. Drawbacks of the Existing System

1. Human Errors: Manual blood typing is prone to human errors, which can lead to incorrect blood group identification and potentially life-threatening situations during transfusions.
2. Time-Consuming: The current methods are often slow, especially when quick results are needed in emergency situations. This can delay critical medical decisions.
3. Dependence on Blood Samples: Existing systems require physical blood samples, making the process invasive and raising the risk of contamination and infection.
4. Limited Accessibility: Automated systems that are available are usually expensive and confined to well-equipped laboratories, limiting their availability in resource-constrained or remote areas.
5. Complexity: The equipment and procedures used in automated blood typing systems are complex, requiring specialized knowledge and maintenance, further limiting their scalability and use in different settings.
6. Non-scalable: The current systems do not scale well, particularly in settings requiring high-throughput testing or in large-scale medical emergencies.

2.2. PROPOSED SYSTEM

The proposed system for "Automated Blood Group Detection" leverages advanced machine learning techniques, specifically Convolutional Neural Networks (CNNs), to analyze fingerprint images and accurately predict an individual’s blood group. The system is designed to be non-invasive, eliminating the need for traditional blood sample collection and laboratory testing. By using a dataset of fingerprint images labeled with corresponding blood groups, the CNN model is trained to recognize patterns and features unique to each blood group. Once trained, the model can process new fingerprint images and provide an immediate prediction of the blood group. This system is intended to be user-friendly and efficient, making it suitable for deployment in various healthcare settings, including hospitals, clinics, and emergency response situations. The non-invasive nature of the system also makes it ideal for use in remote areas where access to conventional blood testing facilities may be limited. By integrating this technology into portable devices or biometric systems, the proposed system aims to provide a quick, reliable, and accessible method for blood group detection.

2.2.1. Advantages of proposed system

1. Non-Invasive: Eliminates the need for blood sample collection, making the process more comfortable for patients.
2. Quick and Efficient: Provides immediate blood group predictions, crucial in emergency situations.
3. Accessible: Can be used in remote or resource-limited areas without the need for specialized equipment.
4. Reduces Human Error: Automates the blood group detection process, minimizing errors associated with manual testing.
5. Portable: Potential for integration into mobile or biometric devices, allowing for on-the-go blood group detection.
6. Cost-Effective: Reduces the need for expensive laboratory tests and trained personnel.

2.3. FEASIBILITY STUDY

2.3.1. Details

A feasibility study is crucial for assessing the viability of the "Automated Blood Group Detection" system. It evaluates various aspects such as technical, economic, operational, and legal factors to determine whether the project is practical and sustainable.

1. Technical Feasibility: The technical feasibility focuses on the availability of technology and expertise required to develop the system. The project leverages Convolutional Neural Networks (CNNs). With the current advancements in machine learning frameworks, developing the system is technically feasible. The project requires a dataset of fingerprint images, which can be sourced or generated, and sufficient storage and computational power to train the model.
2. Economic Feasibility: Economic feasibility assesses the financial viability of the project. The cost of developing the system includes expenses related to data collection, model training, software development, and deployment. Compared to traditional blood testing methods, the proposed system could be more cost-effective in the long run. Once developed, it can be scaled with minimal additional costs, making it economically sustainable.
3. Operational Feasibility: Operational feasibility examines how well the system will function within the current organizational environment. The proposed system is designed to be user-friendly and can be integrated into existing healthcare workflows with minimal disruption. It requires minimal training for healthcare providers, making it easy to adopt.
4. Schedule Feasibility: Schedule feasibility assesses whether the project can be completed within a reasonable timeframe. Developing the system involves several stages: data collection, model development, testing, and deployment. The availability of pre-existing frameworks and tools for machine learning can help accelerate the development process, making the schedule feasible.

2.3.2. Impact on environment

The "Automated Blood Group Detection" system is designed to have a minimal environmental impact, particularly when compared to traditional blood testing methods.

1. Reduction in Medical Waste: Traditional blood testing methods generate significant medical waste, including single-use needles, vials, and other disposable materials. The proposed system, being non-invasive and based on fingerprint analysis, eliminates the need for these materials, thereby reducing the volume of medical waste produced.
2. Lower Energy Consumption: The system primarily requires computational resources for training and running the machine learning model. While training deep learning models can be resource-intensive, once the model is developed, the energy required for processing individual fingerprint images is minimal.
3. Reduced Transportation Emissions: Traditional blood testing often requires transporting samples to centralized laboratories, which contributes to carbon emissions. The proposed system can be deployed locally in hospitals, clinics, and even mobile units, reducing the need for sample transportation and thus lowering the associated carbon footprint.
4. Decreased Use of Chemical Reagents: Conventional blood testing involves the use of various chemical reagents. The automated system avoids the use of such chemicals, reducing potential environmental hazards.

2.3.3. Safety

The safety of the system is a top priority, ensuring that the technology is reliable, secure, and free from risks that could affect users or patients.

1. Accuracy and Reliability: The safety of patients depends on the accuracy of the blood group predictions made by the system. To ensure this, the machine learning model is rigorously trained and tested using a diverse and representative dataset.
2. Non-Invasive Nature: One of the main safety benefits of the system is its non-invasive approach, which eliminates the risks associated with traditional blood collection methods, such as infections, needle-stick injuries, and discomfort.

3. This makes the system particularly safe for use in various populations, including children, the elderly, and those with needle phobias.
4. Operational Safety: The system is designed to be user-friendly, minimizing the chances of operational errors.
5. Risk of Misuse: To prevent misuse, the system is designed with safeguards that limit its use to authorized healthcare professionals.

2.3.4. Ethics

The "Automated Blood Group Detection" system, while innovative and beneficial, must be developed and deployed with careful attention to ethical considerations. Addressing these concerns ensures that the technology is used responsibly and in a manner that respects the rights and dignity of individuals.

1. Informed Consent: Individuals whose fingerprints are used for blood group detection must provide informed consent. Ensuring that consent is obtained in a clear and understandable manner is crucial to maintaining ethical standards.
2. Bias and Fairness: Machine learning models, including those used in this system, can inadvertently incorporate biases present in the training data. It's crucial to ensure that the dataset used to train the model is diverse and representative of all populations to prevent bias against any specific group. The system is designed and tested to ensure it works equally well across different demographic groups, avoiding disparities in accuracy or outcomes.
3. Impact on Employment: The introduction of automated systems in healthcare can raise concerns about job displacement. Ethical deployment of this technology should consider the impact on healthcare workers who currently perform traditional blood group testing. Retraining and upskilling opportunities should be provided to ensure that these workers can transition to new roles within the healthcare system.
4. Autonomy and Choice: Patients should have the autonomy to choose whether they want to use this automated system or opt for traditional blood group testing methods. Respecting patient autonomy means providing alternatives and ensuring that the use of the automated system is not mandatory or coercive.

2.3.5. Cost

The cost analysis for the system involves evaluating the various expenses associated with developing, deploying, and maintaining the system.

- Development Costs
- Testing and Validation Costs
- Deployment Costs
- Maintenance and Support Costs
- Operational Costs
- Training and User Education

2.3.6. Type

The "Automated Blood Group Detection" project is a web-based application designed to provide a convenient and non-invasive method for determining an individual's blood group. Hosted online, the system allows users to upload fingerprint images through a secure web interface. It leverages advanced machine learning models to analyze the fingerprints and predict the blood group in real-time. This web-based approach ensures broad accessibility, enabling healthcare providers and individuals to use the service from any internet-connected device, making it an efficient and scalable solution for blood group detection in various settings.

2.4. SCOPE OF THE PROJECT

The scope of the "Automated Blood Group Detection" project is comprehensive, aiming to transform the traditional approach to blood group testing through innovative web-based technology. At its core, the project focuses on developing a sophisticated web application that leverages Convolutional Neural Networks (CNNs) to analyze fingerprint images and predict an individual's blood group with high accuracy. Firstly, the development of the CNN model is central to the application. This includes the design and training of the model using a diverse dataset of fingerprint images labelled with corresponding blood groups. The model's performance will be rigorously tested to ensure its accuracy and reliability. The web application will serve as the interface through which users interact with the system. In terms of healthcare application, the project aims to integrate the system into various healthcare settings, such as hospitals, clinics, and emergency medical units. This integration will help streamline blood group testing processes, making them quicker and more comfortable for patients. By reducing the need for invasive procedures and lengthy lab tests, the system could significantly enhance patient care and operational efficiency in healthcare facilities. Data management and security are also critical aspects of the project's scope. Future developments may include integrating additional biometric features, such as facial recognition or iris scanning, to enhance the accuracy and versatility of the system. The project may explore the creation of mobile versions of the application to further increase accessibility and convenience for users.

2.5. SYSTEM CONFIGURATION

2.5.1. Hardware Requirements

1. Processor: Intel i5/i7 or AMD Ryzen 5/7 (8th Gen or later)
2. RAM: Minimum 8 GB (16 GB recommended for faster processing)
3. Storage: 500 GB SSD or higher (for storing datasets and model weights)
4. Camera/Sensor: High-resolution image sensor for capturing palm images
5. Peripheral Devices: Standard keyboard, mouse, and monitor

2.5.2. Software Requirements

1. Operating System: Windows 10/11, Linux (Ubuntu 20.04 or later), or macOS
2. Programming Language: Python 3.x
3. IDE/Editor: PyCharm, Jupyter Notebook, or VS Code
4. Libraries and Frameworks
   - OpenCV (for image preprocessing)
   - NumPy and Pandas (for data handling)
   - Scikit-learn (for implementing machine learning models)
   - TensorFlow/Keras or PyTorch (for deep learning if required)
   - Matplotlib and Seaborn (for data visualization)
5. Database: MySQL (for storing login details)

3. LITERATURE OVERVIEW

3.1. TITLE : “FINGERPRINT BASED BLOOD GROUP PREDICTION USING DEEP LEARNING”

This research emphasizes the stability and potential of biometric identification using fingerprint methods across various applications. The study introduces an efficient technique for blood group identification by leveraging the unique minutiae features present in fingerprints. These minutiae features serve as distinctive markers that can be utilized to predict an individual's blood group using different machine learning methods. Specifically, the proposed system employs Multiple Linear Regression with Ordinary Least Squares (OLS) to predict blood groups, achieving an accuracy rate of 62%. While this accuracy level represents a significant step forward in blood group identification through fingerprint analysis, the conclusion highlights avenues for future research and improvement. Firstly, the suggestion to increase the sample size in future studies is crucial. By expanding the dataset with more fingerprint samples and corresponding blood group data, researchers can enhance the reliability and generalizability of the predictive model. Secondly, the conclusion proposes considering additional fingerprint features that have not yet been incorporated into the analysis.

3.2. TITLE : “AN ASSOCIATION BETWEEN FINGERPRINT PATTERNS WITH BLOOD GROUP AND LIFESTYLE BASED DISEASES”

The fingerprints are having immense potential to have an effective method of identification. In this research, it investigates the problem of blood group identification and analysis of disease those arises with aging or disease called as lifestyle-based. With the literature review study, it is observed that fingers of an individual are having multiple unique patterns those are need to be extracted with computerized method with fingerprints image captured using digital device which allow to find known association of fingerprints patterns which may enhance the authenticity of the fingerprints in blood group identification and early indication of lifestyle-based diseases of an individual. The fingerprint used as a traditional, effective, and unique identification method of an individual. The analysis and classification of community based on age, blood group, fingerprint patterns and lifestyle diseases help to tackle any pandemic in future like COVID-19 in which mankind may suffer a lot having lifestyle-based diseases like hypertension, type 2-diabetes.

3.3. TITLE : "DERMATOGLYPHICS AND THEIR RELATIONSHIP WITH BLOOD GROUP"

This study revealed the relation between distribution of, blood group, and gender. This study was carried out on 150 subjects; maximum of the subjects belonged to O blood group, i.e., 35% (53 subjects) were O+, followed by B+, i.e., 33% (50 subjects); A+, i.e., 18% (27 subjects); AB+ i.e., 8% (12 subjects); O−, i.e., 3% (4 subjects); B−, i.e., 2% (3 subjects); and A−, i.e., 1% (1 subject). The universal distribution of fingerprint pattern was of the order in individuals with A, B, AB, and O blood group, i.e., higher frequency for loops, moderate for whorls, and low for arches. In this study, whorl was maximum in A+, i.e., 44% (12 subjects); arch was maximum in A−, i.e., 100% (1 subject); arch was maximum in B+, i.e., 48% (24 subjects); all the three patterns (arch, loop, and whorl) were equally distributed in B−, i.e., 33% (3 subjects); loop was maximum in AB+, i.e., 58% (7 subjects); loop was maximum in O+, i.e., 45% (25 subjects); and arch was maximum in O−, i.e., 75% (3 subjects). In this study, there was a significant correlation between the different fingerprint patterns and the group of the blood among 150 subjects with a P value of 0.046*; (*indicates highly significant $P < 0.05$), which correlates with Mehta, Kshirsagar et al., Bharadwaj et al., and with Rastogi and Pillai.

3.4. TITLE : "BLOOD GROUP DETERMINATION USING FINGERPRINT"

The presence of recurring patterns common to certain blood groups has been studied from various sources. Fingerprint matching algorithms help extract feature necessary for building a deep learning model. Deep learning methods are used in the field of dactylography for reconstruction of fingerprints, latent fingerprint matching and fingerprint classification. Such implementations further the case of utilizing deep learning approaches to associate finger prints with blood groups. Convolutional neural networks can give flattened representations of images that can be used as input for fully connected neural networks that classify the input images. Since the concept of mapping fingerprints is novel and has no standardized approaches, preconfigured CNN architectures like AlexNet, and LeNet-5 are used to initially classify the input fingerprint data. The performance of each of these models are evaluated to understand the optimal features and based on evaluations a specific CNN is built for the application.

3.5. TITLE : "A NOVEL APPROACH TO PREDICT BLOOD GROUP USING FINGERPRINT MAP READING"

The biometric identification using fingerprint method shows immense stability and hence potential to be considered for various applications. The present research proposed efficient technique of blood group identification. A fingerprint having multiple unique minutiae features which are used to predict blood group using different machine learning methods. Proposed system predicts blood group using Multiple Linear Regression with Ordinary Least Squares (OLS) with 62% accuracy. In future further study should be carried out by increasing the sample size to get more accurate result and consider additional fingerprint feature those are not considered yet.

4. SYSTEM DESIGN

4.1. SYSTEM ARCHITECTURE

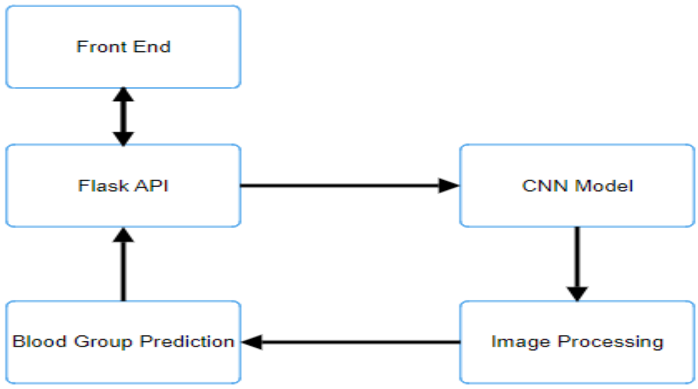

Fig: 4.1.1 System Architecture

The system architecture of this project is based on a multi-layered design. At the top layer, the User Interface provides a web interface for users to interact with the system. User inputs are validated, and the fingerprint image is forwarded to the Machine Learning Module. The CNN processes the fingerprint and predicts the blood group by analyzing the image's unique features. The prediction is then sent back to the user interface for display. Simultaneously, the Data Storage Module handles the management of user data. During registration and login, it stores and verifies user credentials using a database.

4.1.1. Module Description

4.1.1.1. User Interface (UI) Module

The User Interface Module is designed to provide a seamless and user-friendly experience for interacting with this system. It ensures intuitive navigation and easy access to various features, allowing users to effortlessly submit fingerprint images, receive predictions, and view their results. When a user first interacts with the system, they are greeted by the home page. For security and personalized access, users can log in through the login page, where they are required to enter their credentials. New users are directed to the signup page, where they can create an account by providing essential information. The core functionality of the system is found in the predict page, where users can submit a fingerprint image for analysis. Once an image is submitted, the system processes it through the trained CNN model to predict the user's blood group. After prediction, users are taken to the result page, which displays the detected blood group.

4.1.1.2. Machine Learning Module

The Machine Learning Module is the core of this system, responsible for processing fingerprint images and predicting the associated blood group. This module utilizes a Convolutional Neural Network (CNN), to automatically extract hierarchical features from input images. The training process involved feeding the CNN multiple fingerprint samples, allowing the model to learn intricate patterns and features that differentiate blood groups. The CNN was trained using a labeled dataset. When a user submits a new fingerprint image through the user interface, the Machine Learning Module preprocesses the image. The image is then passed through the trained CNN, which predicts the blood group by evaluating the learned features from the input fingerprint. The final blood group prediction is then passed to the user interface module for display.

4.1.1.3. Data Storage Module

The Data Storage Module is responsible for managing all the data-related operations in this system, particularly for storing and retrieving user information during registration and login. When a user registers through the system, the Data Storage Module ensures that the user's personal details, including their username, email, and password, are securely stored in the database. During the login process, this module plays a crucial role in verifying the user’s credentials. When a user submits their login details, the system retrieves the corresponding stored credentials from the database and compares them. If the password matches the stored hash for the given username or email, the user is granted access to the system. If the credentials do not match, the system prompts the user with an error message and denies access.

As a team, we worked collaboratively to develop the "Automated Blood Group Detection" system, leveraging our combined skills in machine learning, software development, and database management. We divided tasks based on our expertise, with focusing on the development of the user interface and on building the machine learning model and ensuring secure data handling. Our efforts involved extensive research, data collection, and model training using fingerprint images, allowing us to create an efficient system capable of predicting blood groups.

Our team contributions included building the User Interface Module, which provides a smooth and intuitive experience for users, and implementing the Machine Learning Module using a CNN for fingerprint analysis. We also developed a robust Data Storage Module to handle user information securely, ensuring that all interactions with the system are protected and efficient.

4.2. UML DIAGRAMS

4.2.1. Use Case Diagram

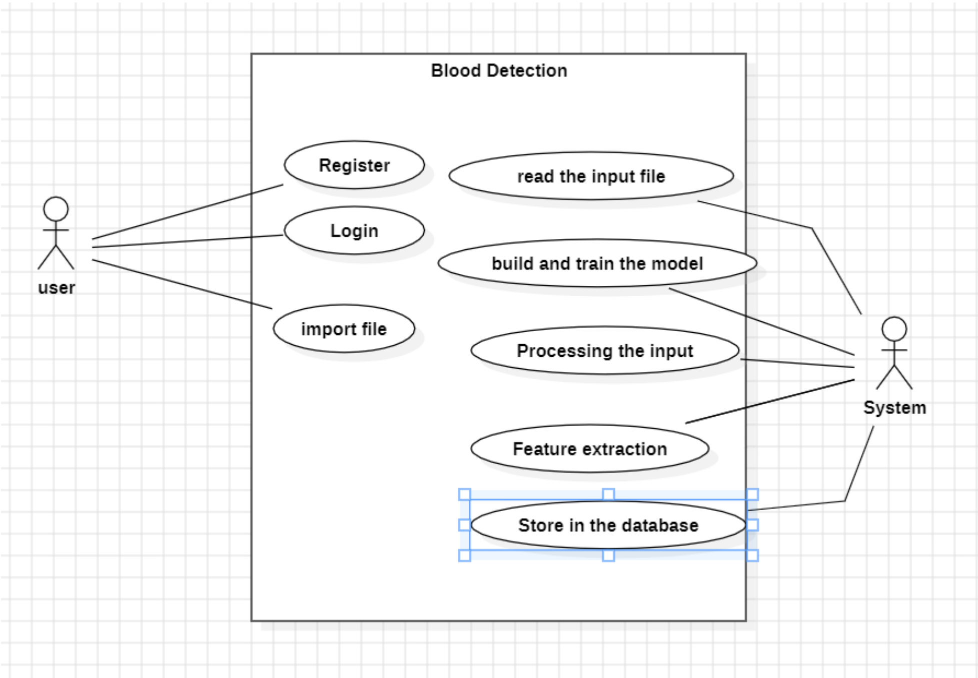

Fig 4.2.1.1. Use Case Diagram

4.2.2. Sequence Diagram

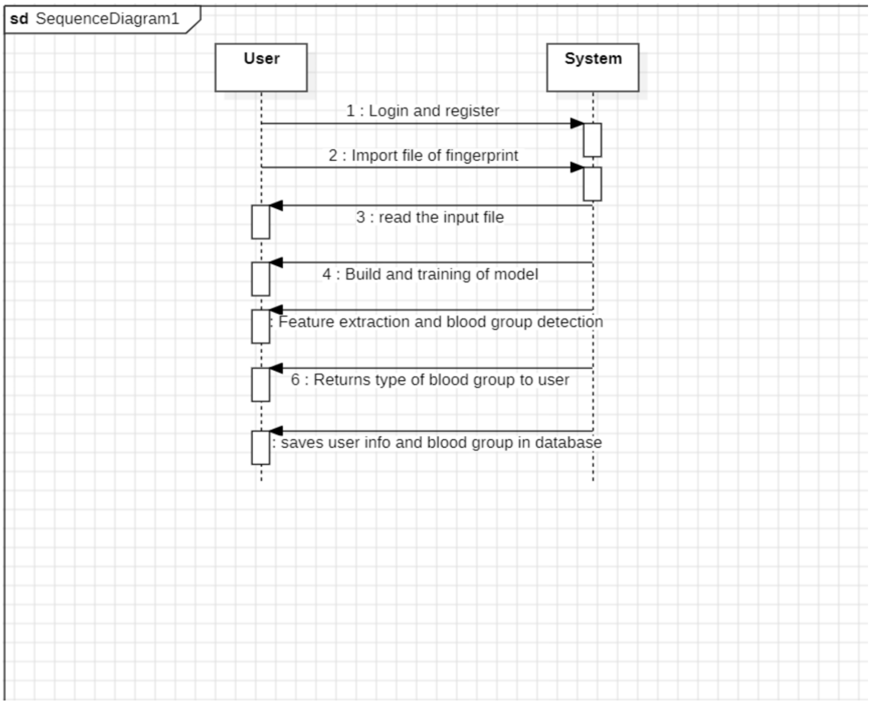

Fig 4.2.2.1. Sequence diagram

4.2.3. Activity Diagram

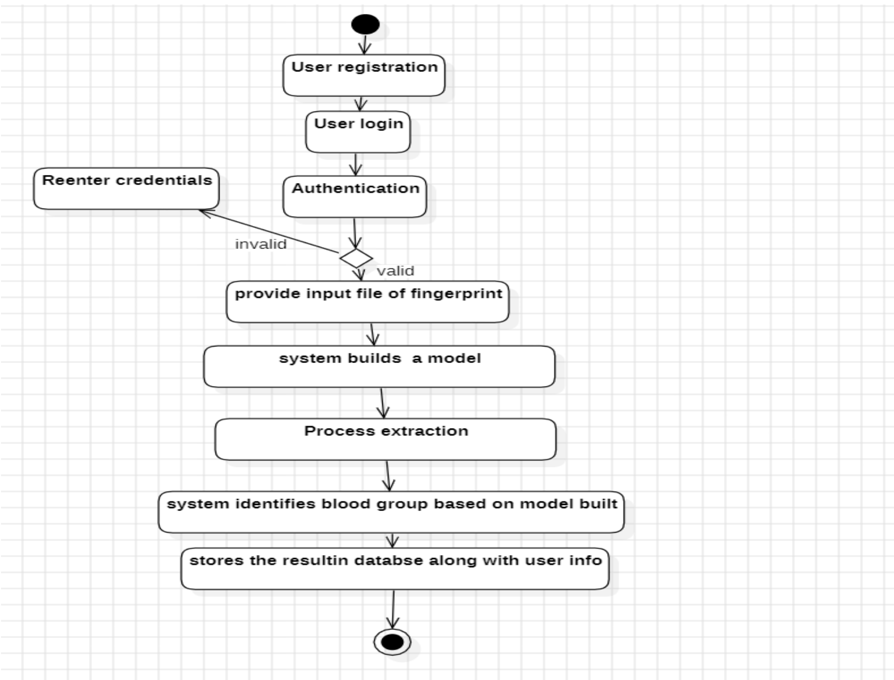

Fig 4.2.3.1. Activity diagram

4.3. SYSTEM DESIGN

4.3.1. Modular Design

4.3.1.1. User Interface (UI) Module

- Provides the front-end interface for users to interact with the system.
- Allows users to upload fingerprint images, view blood group predictions, and manage their accounts.
- Home Page: Introduction to the application and its features.
- Upload Page: Form for uploading fingerprint images.
- Result Page: Displays the predicted blood group.
- Authentication Pages: Login, Signup, and Profile management.
- Technologies: HTML, CSS, JavaScript

4.3.1.2. Machine Learning Module

- Manages the CNN model used for predicting blood groups from fingerprint images.
- Model Training Pipeline: Data preprocessing, model training, and validation.
- Inference Engine: Predicts the blood group based on the input fingerprint.
- Model Evaluation: Monitors the accuracy and performance of the model.
- Technologies: TensorFlow, Keras, PyTorch.

4.3.2. Dataset Design

The dataset for the Automated Blood Group Detection system is primarily focused on fingerprint images and their corresponding blood group labels.

- Fingerprint images labelled with the corresponding blood groups (A+, A-, B+, B-, AB+, AB-, O+, O-).
- This dataset design ensures that the system has a well-structured, secure, and scalable data foundation, supporting accurate and efficient blood group predictions.

5. IMPLEMENTATION

5.1. IMPLEMENTATION

The Automated Blood Group Detection System is a web-based application designed to predict a person's blood group using fingerprint images. The implementation involves several key components, including the collection of fingerprint data, the development and training of a machine learning model, and the deployment of the system as a user-friendly web interface.

5.1.1. Project Overview

The core idea of this project is to leverage the unique patterns present in human fingerprints to predict an individual's blood group. This approach aims to automate the traditional blood typing process, reducing the dependency on manual procedures, and minimizing the risk of human error. The web application allows users to upload a fingerprint image, which is then processed by a machine learning model to predict the blood group.

5.1.2. Dataset Collection

The project begins with the collection of a dataset containing fingerprint images labeled with corresponding blood groups. The dataset plays a crucial role in training the machine learning model. The dataset is sourced from public repositories that provide fingerprint images paired with blood group information. The dataset includes various blood groups (A+, A-, B+, B-, AB+, AB-, O+, O-).

5.1.3. Data Preprocessing

Fingerprint images are resized to a consistent dimension and normalized to ensure uniform input. This helps in reducing the computational load and improves the model's accuracy.

5.1.4. Algorithm Implementation

The heart of the system lies in the machine learning model that analyzes the fingerprint images and predicts the blood group. A Convolutional Neural Network (CNN) is chosen for this task due to its proven effectiveness in image recognition tasks. The CNN model is designed with multiple layers, including convolutional layers, pooling layers, and fully connected layers. Each layer is responsible for extracting different levels of features from the fingerprint images. The CNN model is trained on the preprocessed dataset. During training, the model learns to recognize the correlation between fingerprint patterns and blood group labels. The dataset is split into training and validation sets to monitor the model's performance and avoid overfitting. After training, the model's accuracy is evaluated using the validation set. Metrics like accuracy, precision, recall, and F1-score are used to assess the model's performance.

5.1.5. Web Application Development

The user interface is developed using HTML, CSS, and JavaScript. The frontend includes pages like the homepage, about page, team page, login and signup pages, and the main prediction page where users can upload their fingerprint images. The backend is built using a server-side technology such as Python with Flask. The backend handles user authentication, manages the uploaded images, and interacts with the machine learning model to generate predictions. The trained CNN model is deployed on the backend server. When a user uploads a fingerprint image, the image is sent to the model, which processes it and returns the predicted blood group.

5.2. SAMPLE CODE

5.2.1. Prediction Model

```
import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
from torch.utils.data import DataLoader, random_split
import os
from PIL import Image
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f'Using device: {device}')
# Define the transformation for the input image
data_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])])
# Load the dataset
data_dir = r'D:\Mini\archive1\dataset_blood_group'
image_dataset = datasets.ImageFolder(data_dir, transform=data_transform)
# Split the dataset into training and validation sets (80/20 split)
train_size = int(0.8 * len(image_dataset))
val_size = len(image_dataset) - train_size
train_dataset, val_dataset = random_split(image_dataset, [train_size, val_size])
# Optimize data loading
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True,
num_workers=8, pin_memory=True)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False, num_workers=8,
pin_memory=True)
dataloaders = {'train': train_loader, 'val': val_loader}
# Define a simpler CNN model
class SimpleCNN(nn.Module):
```

```
    def _init_(self):
        super(SimpleCNN, self)._init_()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.fc1 = nn.Linear(64 * 56 * 56, 512)
        self.fc2 = nn.Linear(512, 8)  # 8 classes for the 8 blood groups
    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 64 * 56 * 56)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
model = SimpleCNN().to(device)
# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)
# Training the model
num_epochs = 25
for epoch in range(num_epochs):
    print(f'\nEpoch {epoch+1}/{num_epochs}')
    for phase in ['train', 'val']:
        if phase == 'train':
            model.train()
            print("Training...")
        else:
            model.eval()
            print("Validating...")
        running_loss = 0.0
        running_corrects = 0
        for inputs, labels in dataloaders[phase]:
```

```
            inputs = inputs.to(device)
            labels = labels.to(device)
            optimizer.zero_grad()
            with torch.set_grad_enabled(phase == 'train'):
                outputs = model(inputs)
                _, preds = torch.max(outputs, 1)
                loss = criterion(outputs, labels)
                if phase == 'train':
                    loss.backward()
                    optimizer.step()
            running_loss += loss.item() * inputs.size(0)
            running_corrects += torch.sum(preds == labels.data)
            print(f"Batch done: Loss: {loss.item():.4f}")
        epoch_loss = running_loss / len(dataloaders[phase].dataset)
        epoch_acc = running_corrects.double() / len(dataloaders[phase].dataset)
        print(f'{phase.capitalize()} Loss: {epoch_loss:.4f} Acc: {epoch_acc:.4f}')
# Save the trained model
torch.save(model.state_dict(), 'fingerprint_blood_group_model.pkl')
```

5.2.2. Flask code

```
from flask import Flask, request, jsonify, render_template, redirect, url_for, flash,
session
import torch
from werkzeug.security import generate_password_hash, check_password_hash
import torchvision.transforms as transforms
app = Flask(_name_, template_folder='templates')
app.secret_key = 'your_secret_key'
UPLOAD_FOLDER = 'static/uploads/'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
# Path for saving uploads
if not os.path.exists(UPLOAD_FOLDER):
    os.makedirs(UPLOAD_FOLDER)
```

```
# Database connection
def get_db_connection():
    conn = sqlite3.connect('users.db')
    conn.row_factory = sqlite3.Row
    return conn
def init_db():
    with get_db_connection() as conn:
        conn.execute(
            'CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, fullname TEXT, email TEXT UNIQUE, username TEXT UNIQUE, password TEXT)')
        conn.commit()
init_db()
class SimpleCNN(torch.nn.Module):
    def _init_(self):
        super(SimpleCNN, self)._init_()
        self.conv1 = torch.nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.conv2 = torch.nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.fc1 = torch.nn.Linear(64 * 56 * 56, 512)
        self.fc2 = torch.nn.Linear(512, 8)  # Assuming 8 classes for blood groups
    def forward(self, x):
        x = self.pool(torch.nn.functional.relu(self.conv1(x)))
        x = self.pool(torch.nn.functional.relu(self.conv2(x)))
        x = x.view(-1, 64 * 56 * 56)
        x = torch.nn.functional.relu(self.fc1(x))
        x = self.fc2(x)
        return x
model = SimpleCNN()
model.load_state_dict(torch.load('fingerprint_blood_group_model.pth',
map_location=torch.device('cpu')))
model.eval()
# Define the transformation for the input image
```

```
data_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])])
# Predict blood group route
@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})
    file = request.files['file']
    if file.filename == '':
        return jsonify({'error': 'No selected file'})
    if file:
        # Save file to upload folder
        filename = file.filename
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)
        # Load image, apply transformations
        img = Image.open(file_path).convert('RGB')
        img_tensor = data_transform(img).unsqueeze(0)  # Add batch dimension
        # Perform prediction
        with torch.no_grad():
            prediction = model(img_tensor)
        blood_groups = ['A+', 'A-', 'AB+', 'AB-', 'B+', 'B-', 'O+', 'O-']
        predicted_group_name = blood_groups[predicted_blood_group]
        # Redirect to result page with the predicted blood group
        return render_template('result.html', blood_group=predicted_group_name)
@app.route('/')
def home():
    return render_template('hospitalhomepage.html')
@app.route('/about')
def about():
```

```
    return render_template('about.html')
@app.route('/team')
def team():
    return render_template('team.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        conn = get_db_connection()
        cursor = conn.cursor()
        cursor.execute('SELECT * FROM users WHERE username = ?', (username,))
        user = cursor.fetchone()
        conn.close()
        if user and check_password_hash(user['password'], password):
            session['user_id'] = user['id']
            flash('Login successful!', 'success')
            return redirect(url_for('predict_blood_group'))
        else:
            flash('Invalid username or password!', 'error')
    return render_template('login.html')
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        fullname = request.form['fullname']
        email = request.form['email']
        username = request.form['username']
        password = request.form['password']
        confirmpassword = request.form['confirmpassword']
        if password != confirmpassword:
            flash('Passwords do not match!', 'error')
            return redirect(url_for('signup'))
```

```
    hashed_password = generate_password_hash(password,
method='pbkdf2:sha256')
    conn = get_db_connection()
    cursor = conn.cursor()
    try:
      cursor.execute('INSERT INTO users (fullname, email, username, password)
VALUES (?, ?, ?, ?)',
              (fullname, email, username, hashed_password))
      conn.commit()
      flash('Account created successfully!', 'success')
      return redirect(url_for('login'))
    except sqlite3.IntegrityError:
      flash('Username or email already exists!', 'error')
    finally:
      conn.close()
  return render_template('signup.html')
@app.route('/logout')
def logout():
  session.pop('user_id', None)
  flash('You have been logged out.', 'success')
  return redirect(url_for('login'))
@app.route('/predict_blood_group')
def predict_blood_group():
  if 'user_id' not in session:
    flash('Please log in to access this page.', 'error')
    return redirect(url_for('login'))
  return render_template('login2.html')
if _name_ == '_main_':
  app.run(debug=True)
```

FOR FULL CODES : https://github.com/abhayamanasa/Automated-Blood-Group-Detection

6. TESTING

6.1. TESTING

6.1.1. Unit Testing

Unit Testing involves testing individual components or modules of the system to ensure that each part functions correctly in isolation.

6.1.1.1. Model Prediction

The Convolutional Neural Network (CNN) model used for predicting blood groups from fingerprint images is tested with various inputs to verify its accuracy.

- Correctly classified fingerprint images from the training dataset.
- Fingerprint images not present in the training dataset to check generalization.
- Edge cases, such as low-quality images to test the model's robustness.

6.1.1.2. Frontend Components

Each user interface element, such as buttons, forms, and image upload features, is tested to ensure they function as expected.

- The upload button triggers the image upload process.
- Input fields for login and signup pages validate user entries correctly.
- The homepage and other static pages load properly without errors.

6.1.1.3. Backend Functions

The backend logic, including user authentication, image processing, and database interactions, is tested.

- Validation of user login and registration processes.
- Handling of image upload and storage in the server.
- Interaction between the backend and the CNN model for prediction.

6.1.2. Integration Testing

Integration Testing focuses on verifying that different components of the system work together seamlessly.

6.1.2.1. Model Integration with Backend

The integration between the CNN model and the backend server is tested to ensure that uploaded images are correctly processed and predictions are returned without issues.

- Checking the flow from image upload to prediction display.
- Ensuring that the backend correctly handles model errors or timeouts.

6.1.2.2. Frontend and Backend Interaction

The communication between the frontend user interface and the backend server is tested.

- Verifying that user inputs on the frontend trigger the correct backend processes.
- Ensuring that the backend responses are properly rendered on the frontend.

6.1.3. System Testing

System Testing involves testing the entire system as a whole to verify that it meets the specified requirements.

6.1.3.1. End-to-End Testing

The complete workflow is tested, starting from user registration and login to image upload, prediction, and display of results. This ensures that the system performs all intended functions correctly when used by an end-user.

6.1.3.2. Performance Testing

The system's performance is tested under various conditions to assess its response time, load handling, and scalability.

- Testing the prediction response time to ensure it is within acceptable limits.
- Stress testing the system with multiple concurrent users to verify that it can handle high traffic without crashing or slowing down.

6.2. TEST CASES

Test Case ID	**Test Case Description**	**Test Steps**	**Expected Result**	**Test Result**
TC-01	Verify successful user registration	1. Navigate to the signup page. 2. Enter valid details. 3. Submit the form.	User account is created successfully, and the user is redirected to the login page.	Success
TC-02	Verify registration with existing username	1. Navigate to the signup page. 2. Enter an username already registered. 3. Submit the form.	The system should display an error message indicating the email is already in use.	Success
TC-03	Verify login with correct credentials	1. Navigate to the login page. 2. Enter valid credentials. 3. Submit the form.	The user is successfully logged in and redirected to the prediction page.	Success
TC-04	Verify login with incorrect credentials	1. Navigate to the login page. 2. Enter incorrect credentials.	The system displays an error message indicating incorrect username or password.	Success
TC-05	Verify successful image upload	1. Log in to the system. 2. Navigate to the upload page. 3. Upload a valid fingerprint image.	The image is successfully uploaded, and the system proceeds to process the image.	Success

TC-06	Verify correct blood group prediction for known samples	1. Upload a fingerprint image of a known blood group.	The system predicts the correct blood group.	Success
TC-07	Verify prediction with untrained fingerprint image	1. Upload a fingerprint image not present in the training dataset.	The system should still predict the correct blood group or handle the case appropriately.	Success
TC-08	Verify navigation between different pages	1. Log in to the system. 2. Navigate between homepage, about page, upload page, and other static pages.	All pages should load correctly, and navigation should be smooth.	Success
TC-09	Verify the functionality of the logout button	1. Log in to the system. 2. Click the logout button.	The user should be logged out and redirected to the login page.	Success

7. OUTPUT SCREENS

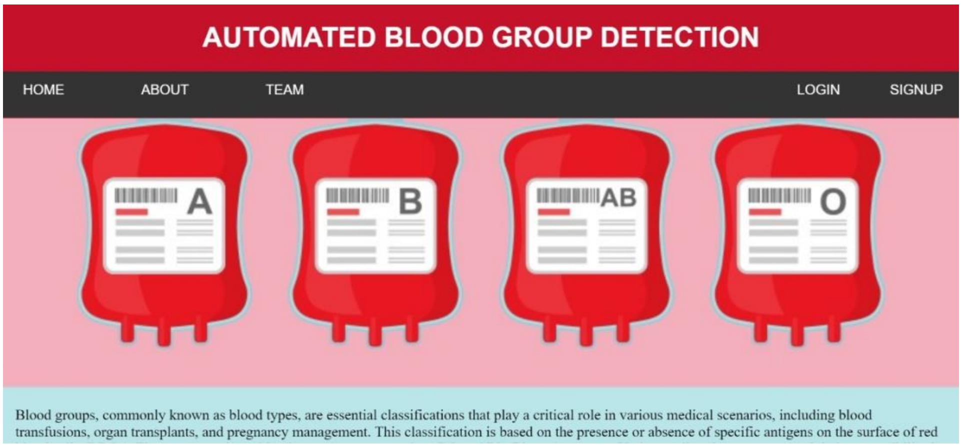

Fig: 7.1. Home page

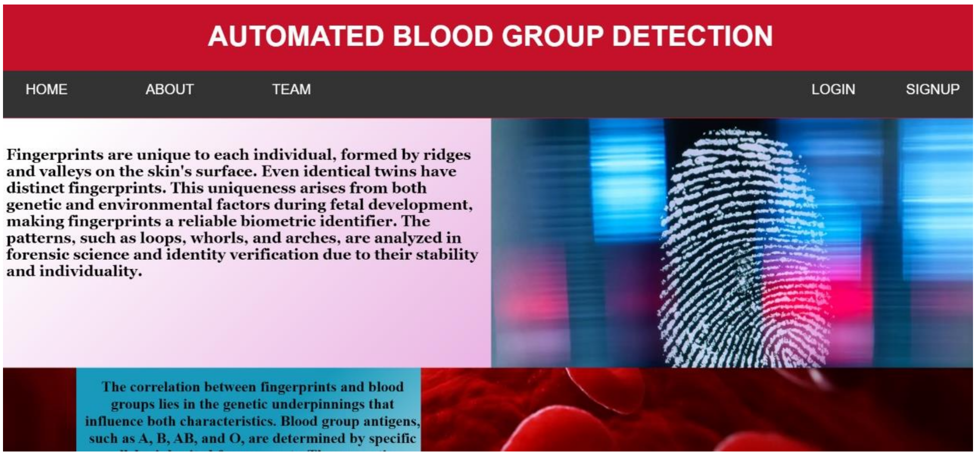

Fig: 7.2. About page 1

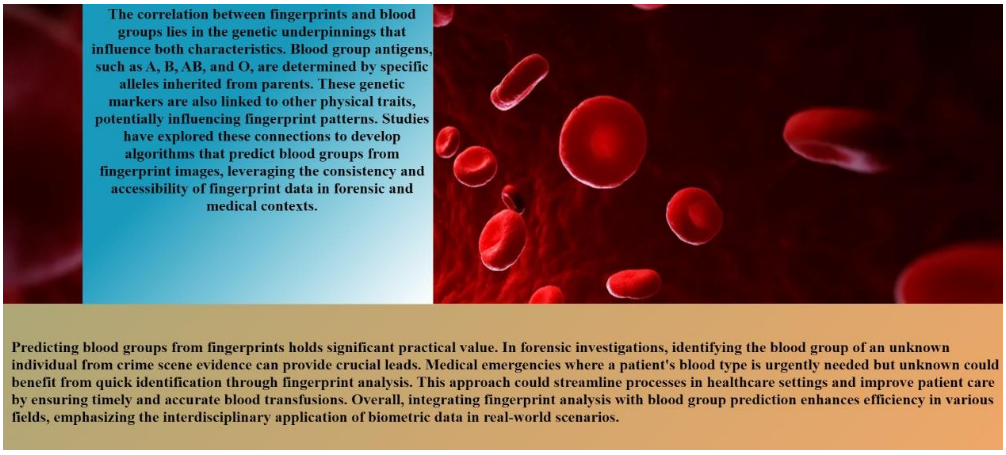

Fig: 7.3. About page 2

Fig: 7.4. Team page 1

Fig: 7.5. Team page 2

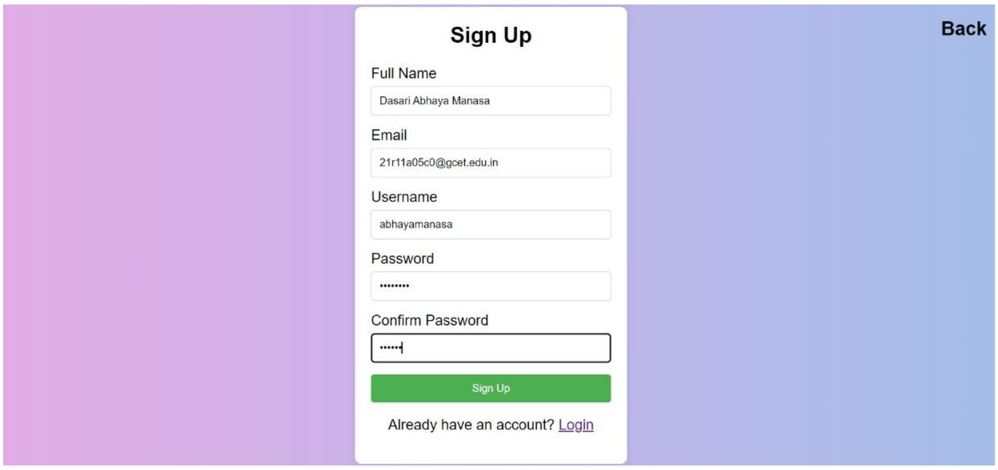

Fig: 7.6. Signup page while filling it

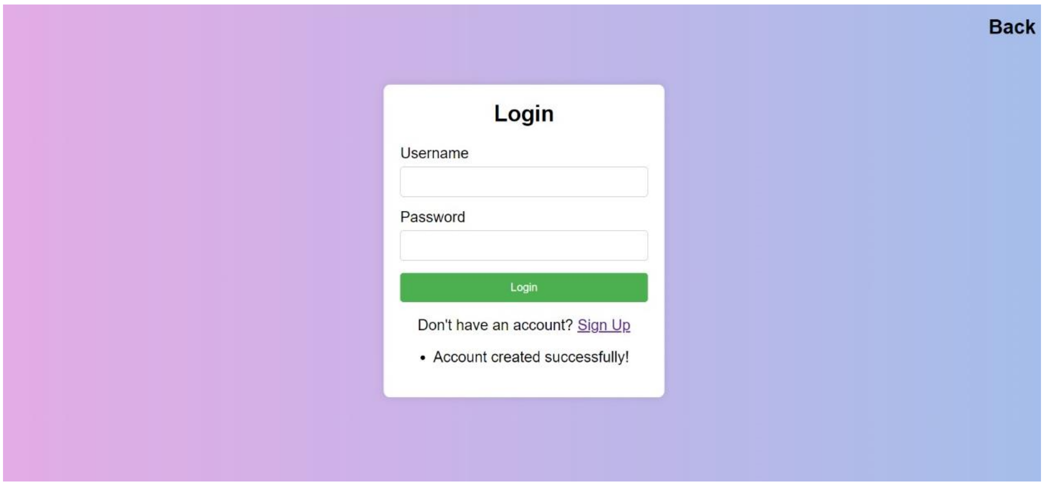

Fig: 7.7. Account successfully created and redirected to login page

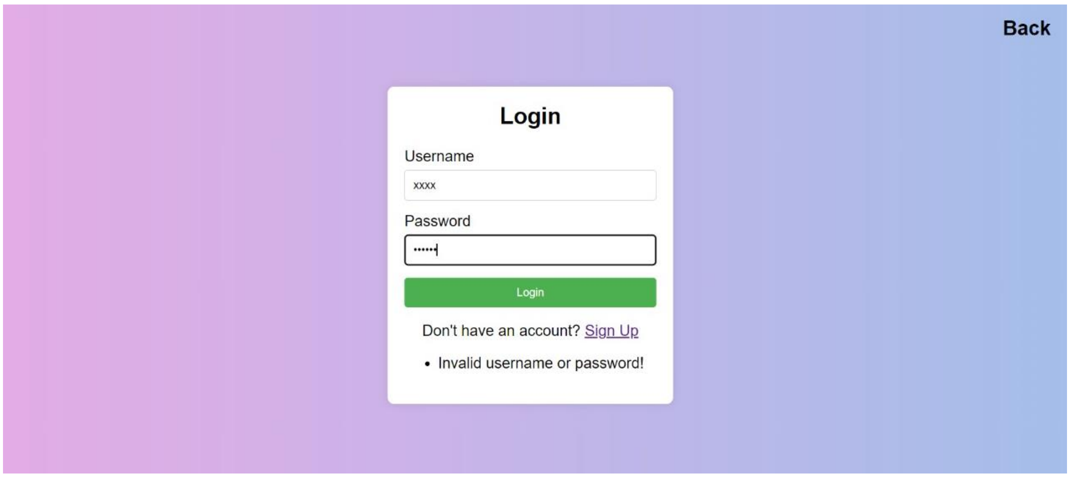

Fig: 7.8. Login Page when invalid credentials are entered

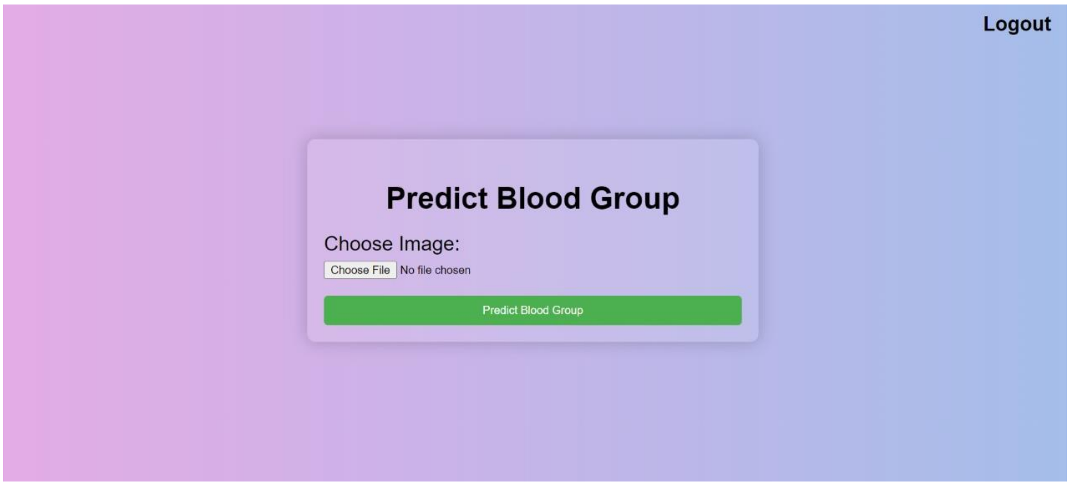

Fig: 7.9. Successful login redirects to prediction page

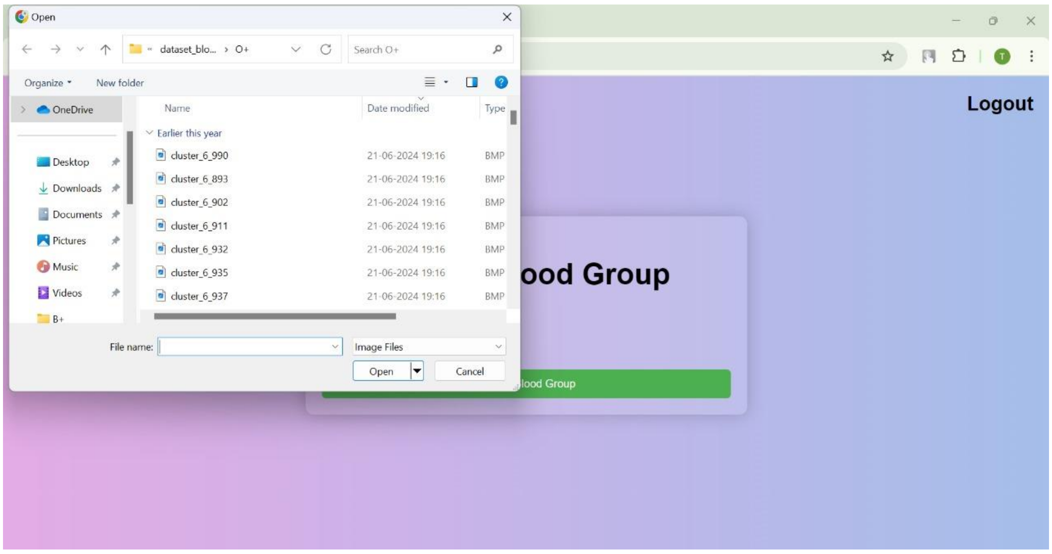

Fig: 7.10. Selecting fingerprint

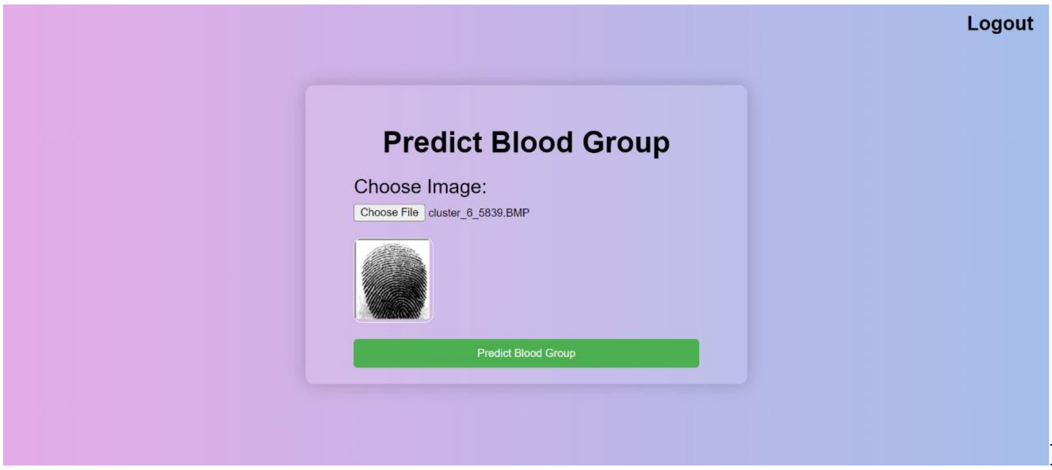

Fig: 7.11. Prediction page after image selection

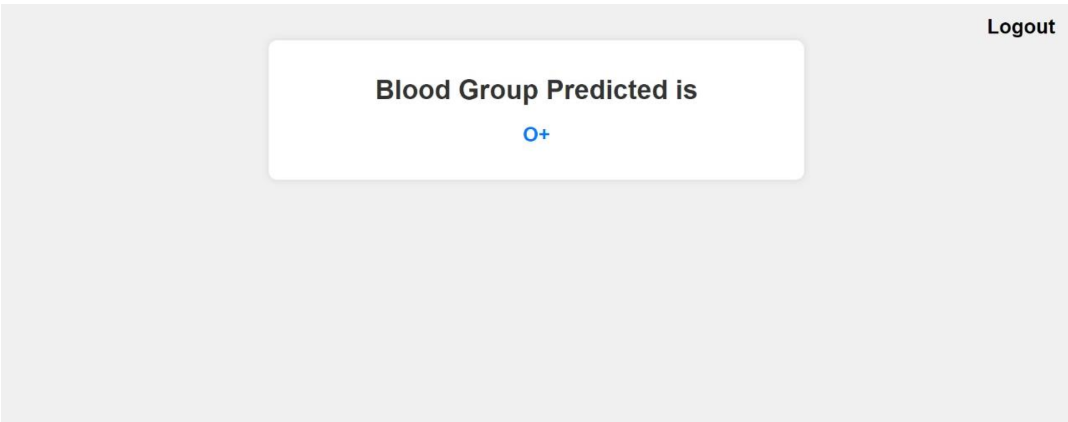

Fig: 7.12. Result page

8. CONCLUSION

8.1. CONCLUSION

The Automated Blood Group Detection System effectively combines machine learning with a user-friendly web interface to provide a reliable method for predicting blood groups based on fingerprint images. Throughout the development process, the system was rigorously tested to ensure accuracy, security, and usability. The CNN model trained on the fingerprint dataset demonstrated strong performance, correctly identifying blood groups in the majority of cases. The system's architecture, which includes clear modular components for image processing, blood group prediction, and user management, allows for future scalability and enhancements. Security measures have been implemented to protect user data and prevent unauthorized access. Overall, this project successfully showcases the potential of automated systems in healthcare diagnostics, providing a foundation for further development and application in real-world scenarios. The system's ability to quickly and accurately determine blood groups has significant implications for improving medical workflows and patient care.

8.2. FURTHER ENHANCEMENTS

The Automated Blood Group Detection System has the potential for several enhancements to improve its functionality, accuracy, and usability. One of the primary areas for improvement is the expansion of the dataset. Incorporating a larger and more diverse dataset with a wider variety of fingerprint images could help refine the CNN model, leading to more accurate predictions across different populations. Additionally, including data from various ethnic groups and age ranges could make the system more universally applicable. Another enhancement could involve integrating multi-modal biometric data. Combining fingerprint analysis with other biometric markers, such as iris scans or facial recognition, could increase the robustness of the system, particularly in cases where fingerprint quality is compromised. Improving the system’s user interface is another area of potential development. Implementing more intuitive navigation, real-time feedback during the image upload process, and a streamlined user experience could make the system more accessible to non-technical users, including healthcare providers in remote areas. Further, the system could be integrated with electronic health records (EHR) systems to allow seamless data sharing and real-time updates of a patient's blood group information, which could be crucial in emergency medical situations. The development of a mobile application version of the system would also enhance its accessibility, enabling on-the-go blood group detection. Finally, incorporating advanced security measures, such as biometric encryption and blockchain for secure data storage, could protect sensitive health information and ensure compliance with medical data regulations. These enhancements would not only improve the system's effectiveness but also expand its applicability in various medical and emergency settings.

9. BIBILIOGRAPHY

- https://ijarsct.co.in/Paper15393.pdf
- https://link.springer.com/article/10.1007/s10462-020-09891-w
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6555323/#:~:text=Loops%20were%20frequent%20and%20whorls,lowest%20in%20B%20blood%20group.
- https://www.matecconferences.org/articles/matecconf/pdf/2024/04/matecconf_icmed2024_01069.pdf
- https://www.researchgate.net/publication/351487579_A_Novel_Approach_to_Predict_Blood_Group_using_Fingerprint_Map_Reading
- https://www.kaggle.com/datasets/rajumavinmar/finger-print-based-blood-group-dataset?select=dataset_blood_group

10. APPENDICES

10.1. APPENDIX A : SOFTWARE USED

- Python: The core programming language used for building the machine learning model and implementing backend functionality.
- TensorFlow/Keras: Deep learning libraries used to construct and train the Convolutional Neural Network (CNN) model.
- OpenCV: A library utilized for image processing tasks, crucial for pre-processing fingerprint images before feeding them into the model.
- Flask: A lightweight web framework used for developing the web-based interface of the system, managing user interactions, and serving the model predictions.
- HTML/CSS/JavaScript: Technologies used for designing and developing the front-end interface, making the website interactive and responsive.
- MySQL: Databases used to store user information.

10.2. APPENDIX B : METHODOLOGIES USED

- Convolutional Neural Network (CNN): A deep learning model designed specifically for image recognition, used here to predict blood groups from fingerprint images.
- Image Preprocessing: Techniques like resizing, normalization, and augmentation were applied to the fingerprint images to improve model accuracy and generalization.
- Supervised Learning: The machine learning model was trained using a labeled dataset where each fingerprint image was tagged with the corresponding blood group.
- Web Development: The Flask framework was used to build the web application, while HTML/CSS/JavaScript were used to design the user interface.

10.3. APPENDIX C : DATASET DESIGN

- Fingerprint Images: The dataset comprised fingerprint images labeled with corresponding blood groups. Images were processed to ensure uniformity in size and format, enabling effective training of the CNN model.
- Training and Validation Split: The dataset was split into training and validation sets to train the model and evaluate its performance on unseen data.
- Data Augmentation: Techniques like rotation, zoom, and flip were applied to the images to increase the dataset size and improve the model's ability to generalize to new data.

10.4. APPENDIX D : SYSTEM ARCHITECTURE

- Client-Server Model: The web application follows a client-server architecture where the client (web browser) interacts with the server (Flask application) to submit fingerprint images and receive predictions.
- Model Integration: The CNN model is integrated into the server-side code, processing the images and returning the blood group prediction to the user.
- Database Management: User data is securely stored in a database, which the server accesses to manage and retrieve data as needed.

10.5. APPENDIX E : MODULAR DESIGN

- User Interface Module: Handles user interactions, including image uploads and displaying predictions. Built with HTML, CSS, and JavaScript.
- Image Processing Module: Preprocesses the uploaded fingerprint images, ensuring they are in the correct format for model prediction.
- Prediction Module: Utilizes the trained CNN model to predict the blood group based on the processed fingerprint image.
- Database Module: Manages the storage and retrieval of user data.

10.6. APPENDIX F : IMPLEMENTATION

- Data Collection: A dataset of fingerprint images labeled with corresponding blood groups was compiled and preprocessed.
- Model Training: The CNN model was trained on the preprocessed dataset, using supervised learning techniques to achieve high accuracy in blood group prediction.
- Web Development: A user-friendly web interface was developed, allowing users to upload fingerprint images and receive instant blood group predictions.
- Deployment: The complete system was deployed on a web server, making it accessible to users.

10.7. APPENDIX G : TESTING

- Test Cases: Various test cases were created to validate the functionality of the system, including image upload, prediction accuracy, and user interface responsiveness.
- Result Analysis: The system's predictions were compared with actual blood group data to assess accuracy, and adjustments were made to improve performance.

11. PLAGIARISM REPORT

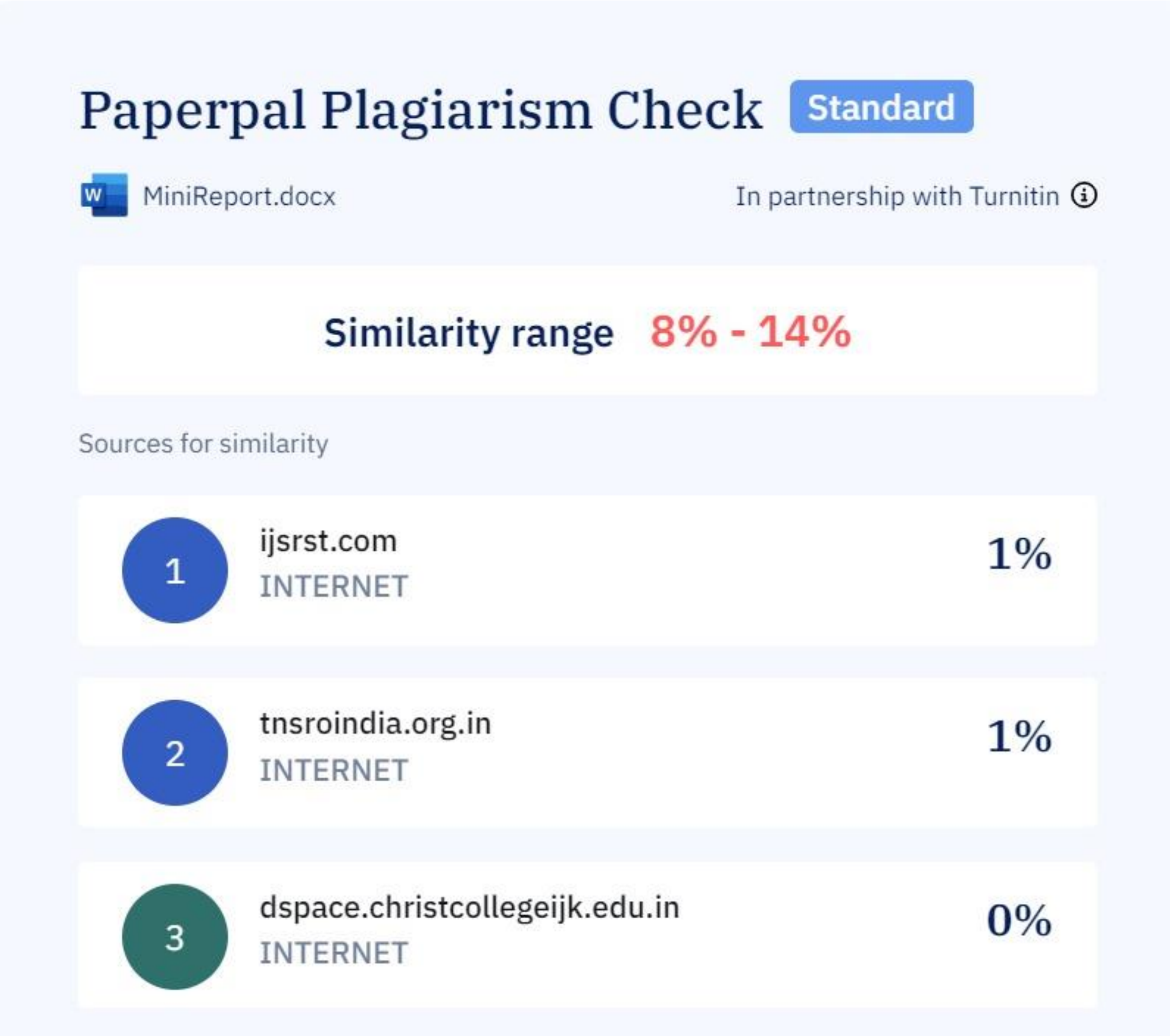

Fig: 11.1. Plagiarism report