CLEANTECH TRANSFORMING WASTE MANAGEMENT WITH TRANSFER LEARNING

SUBMITTED BY

DONDAPATI VENKATA SAI PRAHARSHA - 22BFA37034

ALAMURI JAHNAVI - 22BFA37005

BHOGI SIREESHA - 22BFA37019

AKULA NIVEDITHA - 22BFA37004

INTRODUCTION -

- The rapid advancements in artificial intelligence and machine learning have significantly transformed various domains, including healthcare. One critical area that demands automation and intelligent solutions is hematology — the study and analysis of blood. Manual examination of blood smear slides under a microscope is a
- routine yet laborious task performed by pathologists to detect abnormalities in white blood cells (WBCs). This traditional method requires expert knowledge, is time-consuming, and is susceptible to human error and fatigue, especially in high-volume diagnostic settings.

OBJECTIVE -

The primary objective of the **HematoVision** project is to develop an intelligent and automated system that can accurately classify blood cells using deep learning techniques, specifically through transfer learning. This project aims to improve the efficiency, accuracy, and accessibility of hematological diagnostics by integrating artificial intelligence with a simple and interactive web interface. The following are the detailed objectives of the project:

- **1. Automation of Blood Cell Classification**

To eliminate the dependency on manual observation by pathologists for blood smear analysis by automating the classification of white blood cells (WBCs) into four major types: eosinophils, lymphocytes, monocytes, and neutrophils. Automation aims to reduce diagnostic time and increase reliability and reproducibility.

- **2. Application of Transfer Learning for Accuracy and Efficiency**

To employ transfer learning, utilizing pre-trained Convolutional Neural Networks (CNNs) such as MobileNet or VGG16, thereby reducing the training time and computational resources required to build an effective model. This helps in improving the model's accuracy even with limited labeled medical data.

REQUIREMENTS USED -

- The success of any software-based project relies heavily on the proper understanding and documentation of system requirements. The HematoVision project, being an AI-enabled web-based diagnostic tool, has been designed with a clear set of functional and non-functional requirements that support its intended purpose — classifying blood cells accurately, efficiently, and in real-time.
- This section categorizes the requirements into Functional Requirements, which define what the system should do, and Non-Functional Requirements, which describe how the system should behave.

Category	Description
Programming	Python 3.x
Framework	Flask
ML Library	TensorFlow / Keras
UI	HTML5, CSS3, Bootstrap
IDE	Visual Studio Code / Jupyter
Dataset	12,000 blood cell images (BCCD)
Hardware	Minimum 4 GB RAM, i5 or higher CPU

TECHNICAL SPECIFICATIONS -

Tools and Libraries Used

- Flask: Used for building the web application backend
- TensorFlow and Keras: For developing and running the deep learning model
- PIL (Python Imaging Library): For image handling and preprocessing
- HTML5, CSS3, Bootstrap 5: Used to design the frontend interface
- OpenCV (optional): Can be used for additional image preprocessing

Project Folder Structure

- The project is structured into separate folders for easy maintenance:
- app.py: Main backend file written in Flask
- healthy_vs_rotten.h5: Trained deep learning model file
- templates/: Contains all HTML files (index.html, result.html, portfolio-details.html)
- static/uploads/: Folder for storing uploaded images temporarily

PROJECT IMPLEMENTATION -

10.1 Backend – Flask (Python)

- app.py is the central file.
- Flask is used to handle routing and backend logic.
- The model is loaded once at startup using load_model('healthy_vs_rotten.h5').
- Uploaded images are processed and passed to the model.
- The predicted result is passed to the result HTML template.

10.2 Frontend – HTML + Bootstrap

- All pages are written using HTML5.
- Bootstrap 5 is used to style the interface, making it responsive and modern.
- File input is provided on index.html.
- Result is displayed dynamically on result.html.

10.3 Image Preprocessing

- Images are resized to 224x224 pixels.
- They are converted into NumPy arrays.
- The data is normalized (pixel values scaled to [0,1]).
- Preprocessed data is reshaped to match model input.

APPLICATION SCENARIOS -

HematoVision is designed to address real-world challenges in both clinical and educational domains. The system can be applied across multiple scenarios where automated blood cell classification is beneficial. Below are the most impactful application areas:

12.1 Scenario 1: Automated Diagnostics in Hospitals

- HematoVision can be integrated into hospital diagnostic systems to automate the detection of white blood cell types. It reduces the dependency on manual classification and speeds up the diagnosis process, especially in pathology labs with high workloads. By automatically identifying eosinophils, lymphocytes, monocytes, and neutrophils, HematoVision assists doctors in identifying infections, immune disorders, or leukemia.

12.2 Scenario 2: Telemedicine and Remote Consultations

- In rural and remote areas, access to specialized diagnostic equipment or trained pathologists is limited. HematoVision can serve as a diagnostic assistant on telemedicine platforms. Healthcare workers can upload cell images remotely, and the system can classify them instantly, enabling remote doctors to provide accurate diagnoses without the need for physical lab visits.

RESULTS

The HematoVision system produces a simple and clear output based on the blood cell image uploaded by the user. The prediction results are presented instantly and are accurate (based on model training).

14.1 Output Format

- After a user uploads an image and clicks the Classify button, the result is displayed in a separate window with the following format:
- Classification Successful

Predicted Cell Type: Lymphocyte

- The predicted label (Lymphocyte, Monocyte, Neutrophil, or Eosinophil) is determined by the softmax output of the deep learning model.

14.2 Sample Test Results

- Uploaded Image 1: "Blood cell with round nucleus" → Prediction: Lymphocyte
- Uploaded Image 2: "Granulated large cell" → Prediction: Neutrophil
- These tests confirm that the pipeline works properly and that predictions are returned as expected (subject to model quality).

THANK YOU