# HematoVision: AI-Powered Blood Cell Classification

# 1. INTRODUCTION

## 1.1 Project Overview

HematoVision is a deep learning project designed to accurately and efficiently classify blood cells from microscopic images. Utilizing a dataset of over 12,000 images, this project leverages transfer learning with the VGG16 architecture to distinguish between four major types of white blood cells: eosinophils, lymphocytes, monocytes, and neutrophils. The final output is a fully functional web application built with Flask, where users can upload a blood cell image and receive an instant classification from the trained AI model. This project serves as a demonstration for real-world applications in healthcare, such as automated diagnostics, remote medical consultations, and medical training tools.

# 2. IDEATION PHASE

This section includes identifying the problem, understanding user needs through empathy mapping, and generating ideas through brainstorming.

# 3. REQUIREMENT ANALYSIS

## 3.1 Technology Stack

Backend: Python  
Deep Learning Framework: TensorFlow, Keras  
Web Framework: Flask  
Data Manipulation: NumPy  
Utilities: Scikit-learn, Matplotlib, Seaborn, Pillow  
Environment: Anaconda

# 4. PROJECT DESIGN

## 4.1 Project Architecture

The application follows a standard client-server model:  
User (Client): Interacts with the web interface (index.html) to upload an image.  
Flask Server (Backend): The app.py script receives the image.  
AI Model: The server preprocesses the image and feeds it to the loaded Keras model (hematovision\_best\_model.h5).  
Prediction: The model returns a prediction (the classified cell type and confidence score).  
Result: The server renders the result.html page, displaying the prediction and the uploaded image back to the user.

# 5. PROJECT PLANNING & SCHEDULING

## 5.1 Setup and Installation

Prerequisites:  
- Anaconda (or Miniconda) installed on your system.  
  
Installation Steps:  
1. Clone the Repository (or create the project folder):  
 git clone [your-github-repo-url]  
 cd HematoVision\_Project  
2. Create and Activate the Conda Environment:  
 conda create --name hematovision python=3.9  
 conda activate hematovision  
3. Install Required Packages:  
 conda install -c anaconda numpy pandas scikit-learn matplotlib scipy seaborn tensorflow flask jupyter  
 pip install pillow

# 6. FUNCTIONAL AND PERFORMANCE TESTING

Performance and functionality were validated through a series of tests covering model inference speed, classification accuracy, and system robustness.

# 7. RESULTS

The application was successfully deployed locally. The user can upload an image and receive real-time predictions of white blood cell types.

# 8. ADVANTAGES & DISADVANTAGES

Advantages:  
- High accuracy classification using deep learning  
- Simple web interface for user interaction  
- Real-time results for fast diagnosis  
  
Disadvantages:  
- Limited to classification of only four white blood cell types  
- Requires local setup of Python and packages

# 9. CONCLUSION

HematoVision demonstrates the power of deep learning in medical diagnostics, offering a glimpse into AI’s potential to assist professionals and improve patient outcomes through efficient and accurate image classification.

# 10. FUTURE SCOPE

Future improvements include enhancing model accuracy with more diverse datasets, extending classification to red blood cells and platelets, and deploying the solution as a scalable cloud-based API for integration into existing healthcare systems.