HematoVision: Advanced Blood Cell Classification Using Transfer Learning

1. INTRODUCTION

1.1 Project Overview

HematoVision is a blood cell classification web application powered by deep learning and transfer learning. It uses a pre-trained convolutional neural network (CNN) to classify blood cells into distinct types such as eosinophils, lymphocytes, monocytes, and neutrophils, enhancing diagnostic precision in medical practice.

1.2 Purpose

To provide a scalable, efficient, and accurate tool for automated blood cell classification, supporting medical diagnostics, telemedicine, and medical education.

2. IDEATION PHASE

2.1 Problem Statement

Manual blood cell classification is time-consuming, expertise-dependent, and error-prone. Automating this process reduces diagnostic delay and improves reliability.

2.2 Empathy Map Canvas

Users: Pathologists, lab technicians, medical students, telemedicine providers
Needs: Fast, reliable blood cell classification
Pain Points: Time-intensive manual review, limited expert availability
Gains: Rapid diagnostics, scalable analysis, enhanced training

2.3 Brainstorming

Explored transfer learning with models like ResNet50, InceptionV3, and MobileNetV2. Designed scalable architecture for web integration and clinical deployment.

HematoVision: Advanced Blood Cell Classification Using Transfer Learning

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User uploads a blood cell image -> System preprocesses and classifies the image -> Displays predicted cell type with confidence score

3.2 Solution Requirement

- Pre-trained CNN model (e.g., ResNet50 or VGG16)
- Flask web framework
- Dataset of 12,000 annotated blood cell images
- Frontend with HTML/CSS/JavaScript

3.3 Data Flow Diagram

(Insert DFD illustration: Image Upload -> Preprocessing -> Model Prediction -> Display Output)

3.4 Technology Stack

- Python, Flask
- TensorFlow / Keras (for deep learning)
- HTML, CSS, JavaScript (frontend)

4. PROJECT DESIGN

4.1 Problem Solution Fit

By automating classification, HematoVision minimizes diagnostic workload and maximizes consistency and scalability.

4.2 Proposed Solution

HematoVision: Advanced Blood Cell Classification Using Transfer Learning

Create a web app that receives an image, processes it using a trained CNN via transfer learning, and outputs the blood cell type.

4.3 Solution Architecture

User -> Web App (Flask) -> Preprocessing -> CNN Model -> Prediction -> Output

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

- Week 1: Dataset cleaning, labeling, augmentation
- Week 2: Model development using transfer learning
- Week 3: Flask backend and UI integration
- Week 4: Testing, evaluation, and documentation

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Evaluated classification accuracy across validation/test sets; verified prediction time and performance under varied loads using different blood cell image sizes.

7. RESULTS

7.1 Output Screenshots

(Include classification results and prediction interface screenshots)

8. ADVANTAGES & DISADVANTAGES

Advantages

- High accuracy via transfer learning
- Reduces diagnostic time and manual error
- Easy integration into medical tools and platforms

Disadvantages

- Requires high-quality annotated data
- GPU dependency for training large models
- Limited to visual blood cell features (no biochemical markers)

9. CONCLUSION

HematoVision successfully demonstrates the potential of AI and transfer learning in automating and improving blood cell classification, supporting healthcare, telemedicine, and education.

10. FUTURE SCOPE

- Expand to include abnormal/malignant blood cells (e.g., leukemia detection)
- Integrate with hospital management systems
- Deploy as a mobile and cloud-based service
- Real-time microscopy image support

11. APPENDIX

- Source Code: app.py, preprocessing scripts, templates
- Dataset: 12,000 blood cell images (structured and annotated)
- GitHub & Demo Link: [Insert your GitHub link]
- Trained Model: hemato_model.h5