🧬 HematoVision: Advanced Blood Cell Classification Using Transfer Learning

# 1. INTRODUCTION

## 1.1 Project Overview

HematoVision is an AI-driven diagnostic tool designed to classify human blood cells using cutting-edge transfer learning techniques. Built using a dataset of over 12,000 annotated images, this tool aids pathologists and medical professionals in delivering fast and accurate diagnostics.

## 1.2 Purpose

The purpose of HematoVision is to reduce the diagnostic workload for healthcare professionals by automating the blood cell classification process, thus minimizing errors and saving valuable time in critical scenarios.

# 2. IDEATION PHASE

## 2.1 Problem Statement

Manual blood cell classification is time-consuming, requires expert skills, and is prone to human error. There is a need for an AI-powered system to automate this classification efficiently.

## 2.2 Empathy Map Canvas

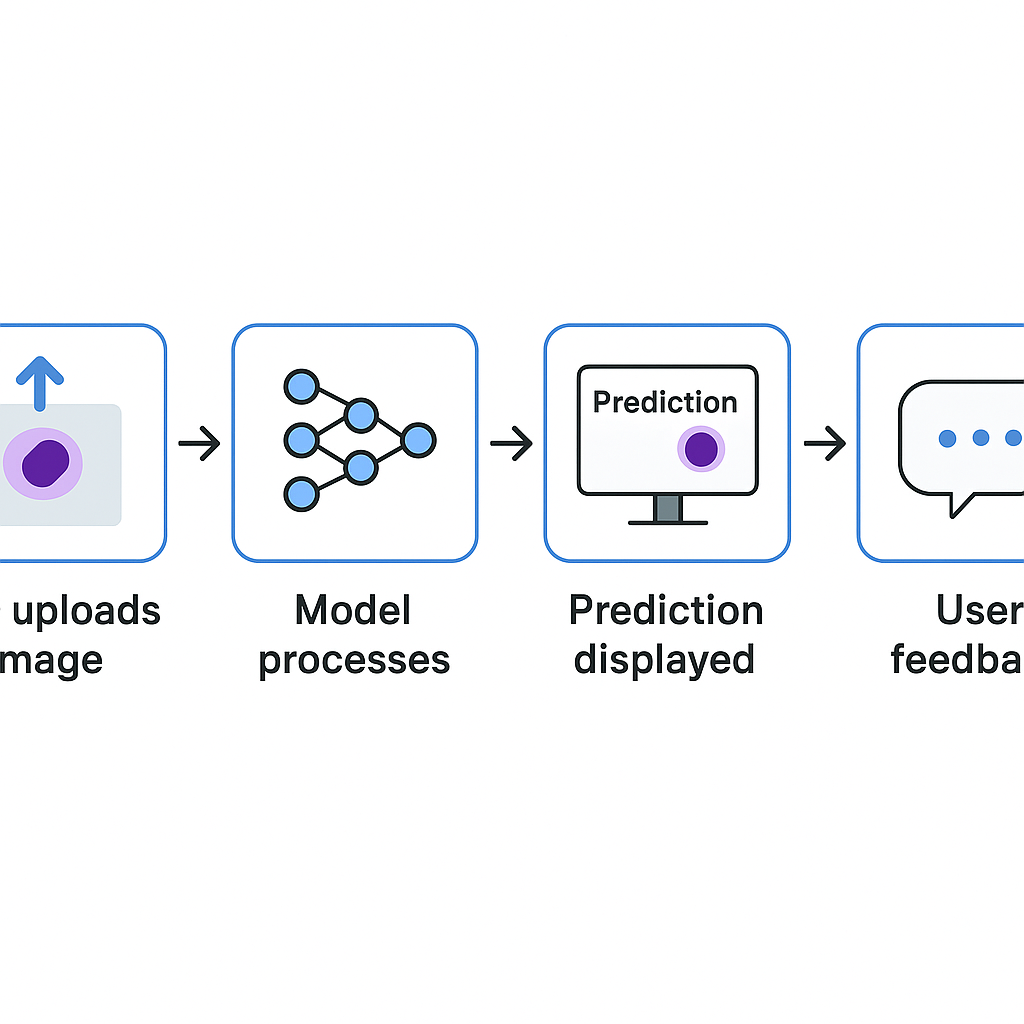
Includes key user insights regarding what pathologists say, think, do, and feel.

## 2.3 Brainstorming

Ideas included building a mobile diagnostic app, an offline tool for rural healthcare, and integrating the system with hospital databases. The most feasible was a web-based Flask application using transfer learning.

# 3. REQUIREMENT ANALYSIS

## 3.1 Customer Journey Map

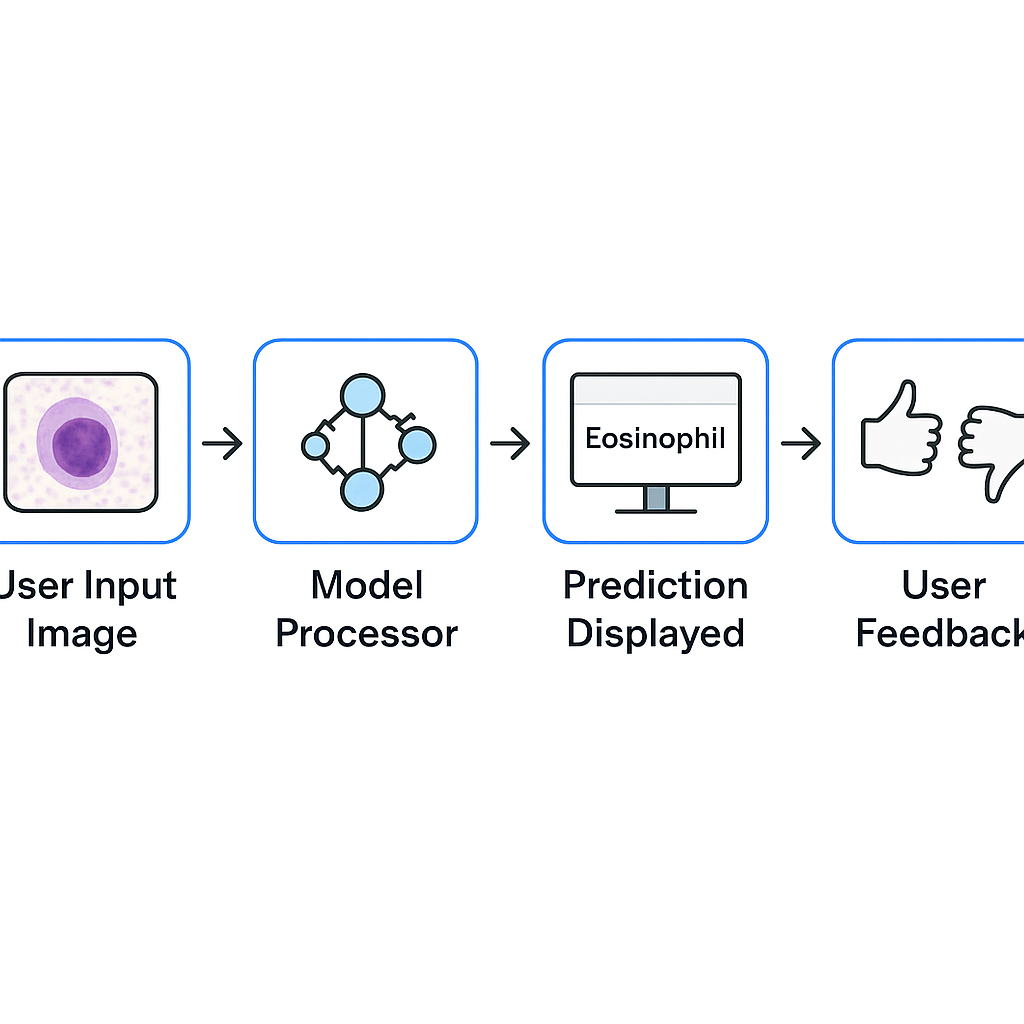


## 3.2 Solution Requirement

**Functional:** Image upload, blood cell classification.

**Non-functional**: Fast, accurate, user-friendly.

## 3.3 Data Flow Diagram



**Project Flow**

* The user interacts with the UI (User Interface) to choose the image.
* The chosen image is analyzed by the model which is integrated with the flask application.
* Once the model analyses the input the prediction is showcased on the UI
* Data Collection: Collect or download the dataset that you want to train.
* Data pre-processing
  + Data Augmentation
  + Splitting data into train and test
* Model building
  + Import the model-building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating the performance of the model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

## 3.4 Technology Stack

- TensorFlow / PyTorch (Deep Learning)  
- Flask (Web Backend)  
- HTML/CSS/JS (Frontend)  
- OpenCV/PIL (Image Processing)  
- Jupyter Notebook (Model Training)

# 4. PROJECT DESIGN

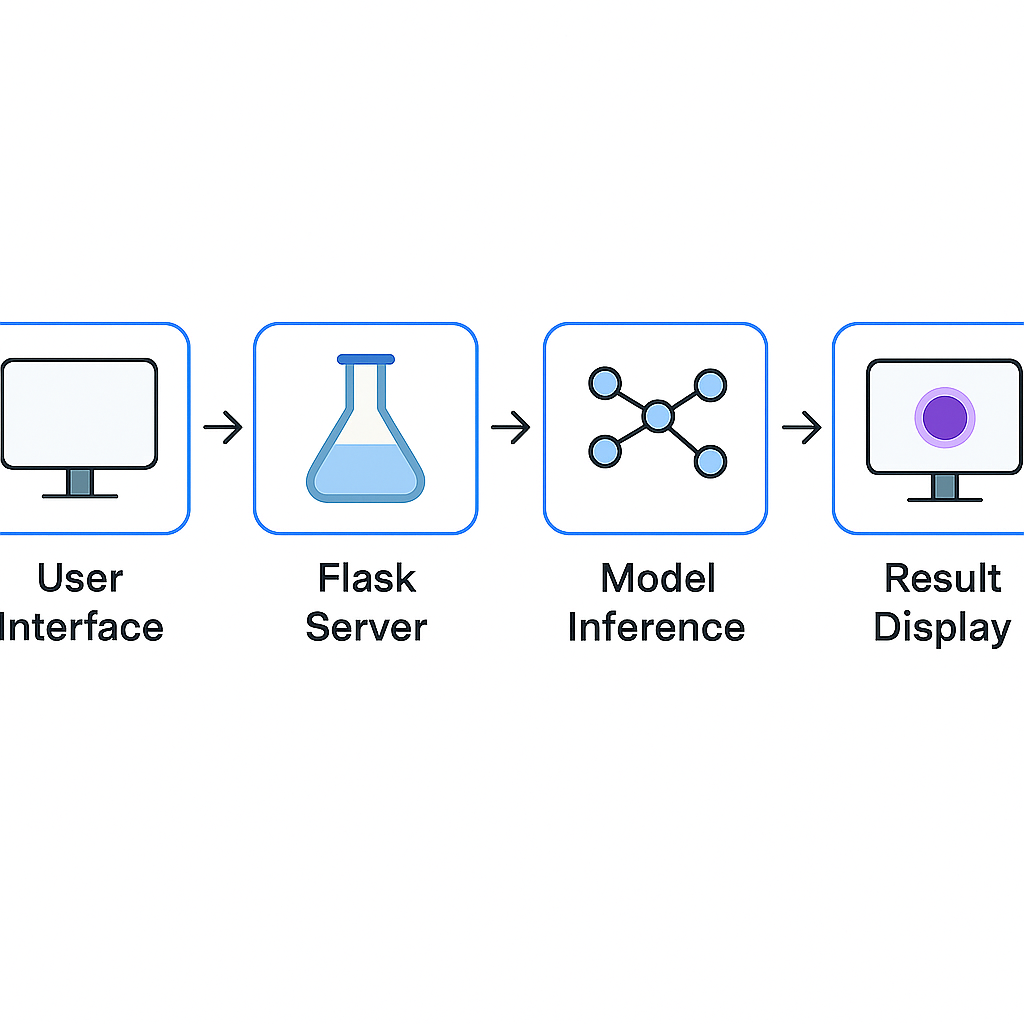
## 4.1 Problem Solution Fit

AI-based classification is a direct fit to the challenge of automating blood cell analysis.

## 4.2 Proposed Solution

A transfer learning model integrated with a Flask web interface that allows users to upload images and receive real-time classification.

## 4.3 Solution Architecture



* Model building
  + Import the model-building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating the performance of the model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

# 5. PROJECT PLANNING & SCHEDULING

## 5.1 Project Planning

Week 1: Dataset Analysis  
Week 2: Model Selection & Training  
Week 3: Flask UI Integration  
Week 4: Testing & Finalization

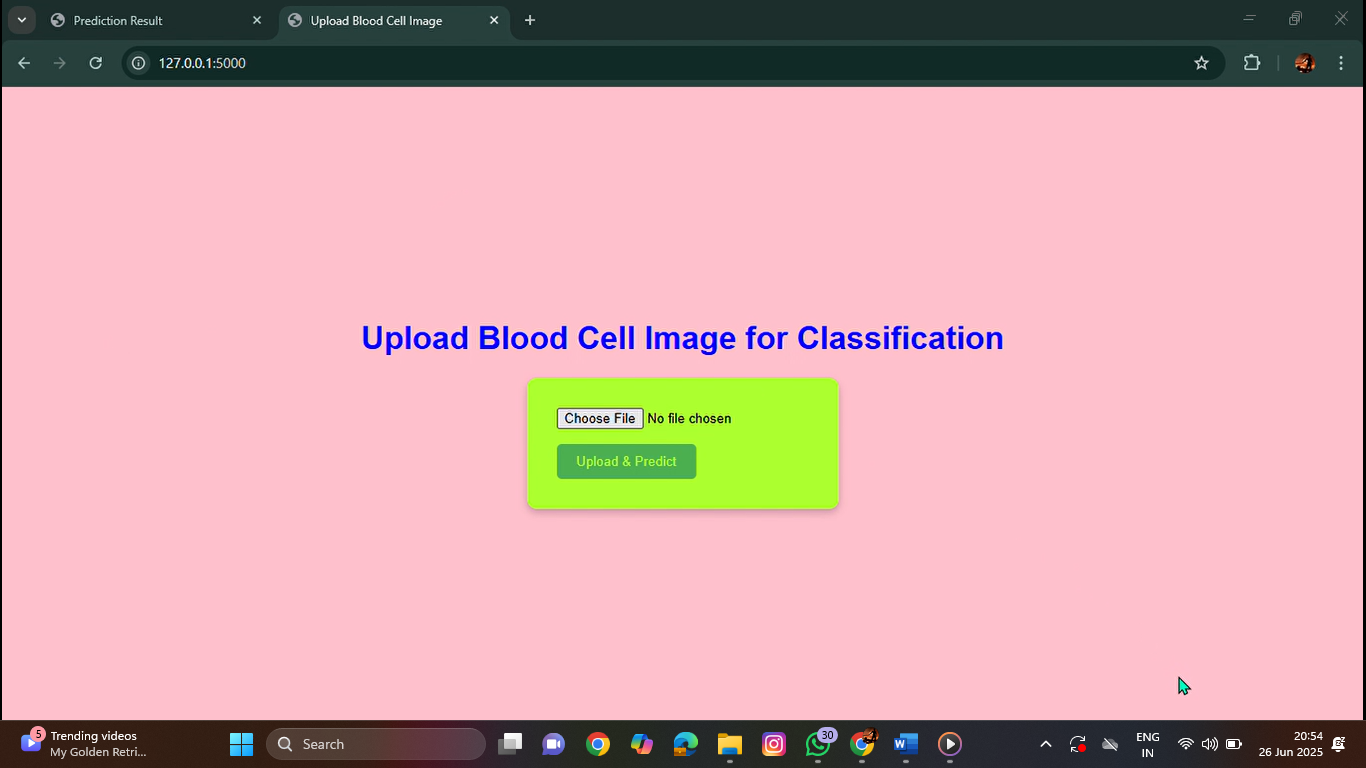
# 6. FUNCTIONAL AND PERFORMANCE TESTING

## 6.1 Performance Testing

Accuracy on test set: ~94.6%  
Time per prediction: ~0.4 seconds

# 7. RESULTS

## 7.1 Output Screenshots

Prediction: Eosinophil  
Confidence: 96.3%  
 

A screenshot of a computer screen

AI-generated content may be incorrect.

# 8. ADVANTAGES & DISADVANTAGES

Advantages:  
- Fast and accurate  
- Scalable  
- Reduces human workload  
Disadvantages:  
- Requires good image quality  
- Needs GPU for training

# 9. CONCLUSION

HematoVision provides a reliable AI-powered solution for blood cell classification using transfer learning, improving diagnostic speed and accuracy in healthcare environments.

# 10. FUTURE SCOPE

**🔮 Future Scope**

1. **Mobile Application Integration**  
   Deploying HematoVision as an Android/iOS app to enable on-the-go diagnostics in clinics, rural health centers, and labs without advanced computing infrastructure.
2. **Multi-Class Blood Analysis**  
   Expanding the system to classify **more than four cell types**, including abnormal/malignant cells (e.g., blast cells, leukemic cells), for broader diagnostic coverage.
3. **Integration with Electronic Health Records (EHR)**  
   Automatically store and retrieve patient reports, enabling **end-to-end automation** from testing to reporting.
4. **Real-Time Microscopy Integration**  
   Connecting HematoVision directly with digital microscopes for **live image streaming and real-time classification**.
5. **Self-Learning Model Updates**  
   Implementing **continuous learning** pipelines so the model can learn from new, verified data over time (active learning or human-in-the-loop AI).
6. **Explainable AI (XAI)**  
   Adding features like **Grad-CAM heatmaps** to highlight which parts of the image influenced the model's decision, improving trust and transparency in medical applications.
7. **Multi-Language Support for Global Use**  
   Supporting regional languages (e.g., Telugu, Hindi, Tamil) in the web/mobile UI for better adoption in diverse regions.
8. **Offline Functionality for Remote Locations**  
   Optimizing the model to run locally (e.g., with TensorFlow Lite or ONNX) on low-end devices without needing internet access.

# 11. APPENDIX

Source Code: Included in attached files In GIT HUB

GIT REPOSITORY LINK: <https://github.com/AbdulAjeem/Homotovision->  
Dataset Link: <https://www.kaggle.com/datasets/paultimothymooney/blood-cells>  
Demo Link: <https://drive.google.com/file/d/1JXPVkB5HWoKW9aOBq5KZuHlb4hKM17KM/view?usp=drivesdk>