PROJECT FINAL REPORT

1. INTRODUCTION

1.1 Project Overview

HematoVision is an AI-based blood cell classification system developed using transfer learning and convolutional neural networks (CNNs). The system classifies blood smear images into four categories: eosinophils, lymphocytes, monocytes, and neutrophils. The trained model is integrated with a Flask web application to enable real-time image upload and prediction.

1.2 Purpose

The purpose of this project is to reduce manual effort in blood cell analysis, minimize human error, and provide a fast, accurate, and scalable diagnostic support system for healthcare professionals.

2. IDEATION PHASE

2.1 Problem Statement

Manual blood cell classification is time-consuming, dependent on expertise, and prone to errors. There is a need for an automated system that provides accurate and consistent results to support clinical diagnosis.

Customer Problem Statement

Pathologists and laboratory technicians face heavy workloads, time constraints, and the risk of human error in manual blood cell analysis. Accurate classification requires expertise and is difficult to scale, especially in rural or resource-limited settings. Healthcare providers need a reliable, fast, and consistent solution that reduces manual effort while maintaining high diagnostic accuracy. HematoVision addresses this need by delivering an AI-powered blood cell classification system that enables efficient, accurate, and scalable diagnostic support.

Problem Statement 1:-

I am	a pathologist who needs to analyze the blood smear samples of many patients daily to provide accurate diagnoses.
I'm trying to	classify blood cells quickly and accurately to ensure patients receive prompt and correct treatment.
but	manual microscopy is slow, labor-intensive, and prone to human errors, making it difficult to keep up with the workload.
because	current diagnostic methods require meticulous visual inspection, and we lack advanced AI tools to assist in blood cell classification.
which makes me feel	overworked, stressed about potential misdiagnosis, and under pressure to deliver results quickly and accurately.

Problem Statement 2:-

I am	a pathologist or laboratory technician responsible for analyzing blood smear samples accurately and efficiently in a clinical setting.
I'm trying to	classify blood cells quickly and correctly to support accurate diagnosis and timely patient treatment.
but	manual microscopy analysis is time-consuming, prone to human error, and highly dependent on individual expertise and workload.
because	traditional diagnostic processes rely heavily on visual inspection without automated assistance, and there is limited access to advanced AI-based tools in many healthcare settings.
which makes me feel	pressured, concerned about potential misdiagnosis, and overwhelmed by increasing diagnostic workload.

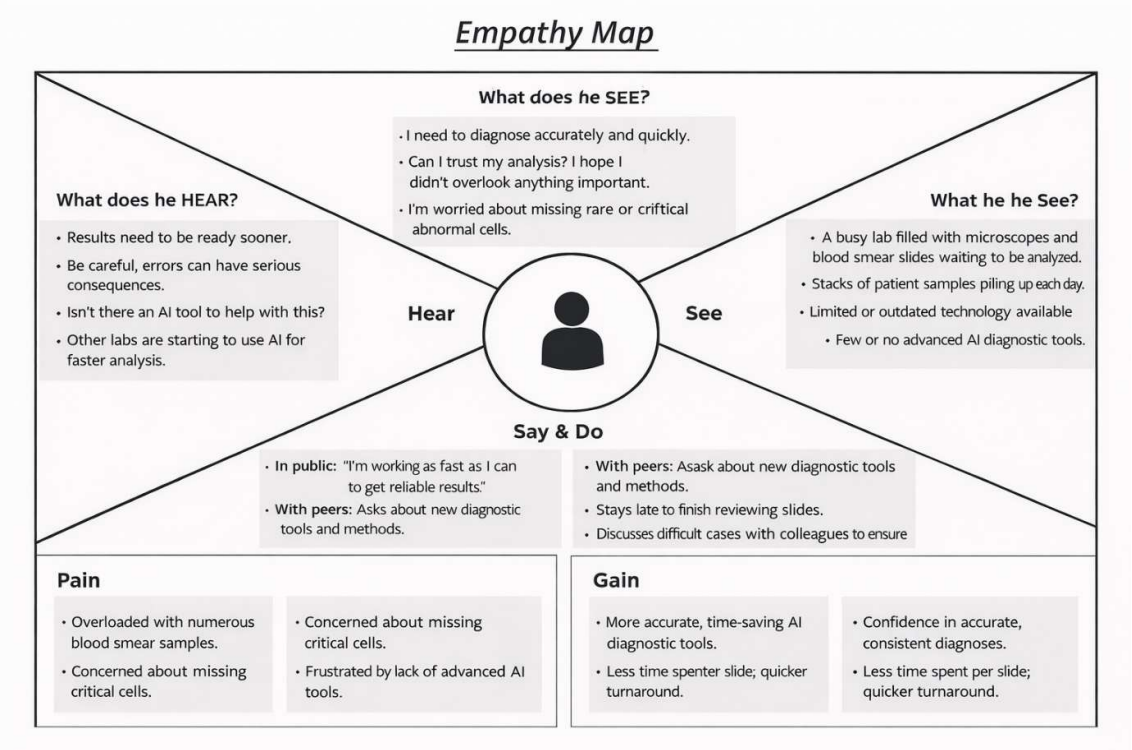
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel

PS-1	a pathologist or laboratory technician responsible for analyzing blood smear samples accurately and efficiently in a clinical setting.	Classify blood cells quickly and accurately for diagnosis.	Manual microscopy is slow, error-prone, and depends on expertise.	Traditional diagnostics rely on manual inspection with limited AI support.	pressured, concerned about potential misdiagnosis, and overwhelmed by increasing diagnostic workload.
PS-2	a pathologist or laboratory technician responsible for analyzing blood smear samples accurately and efficiently in a clinical setting.	Classify blood cells quickly and accurately to support timely and reliable diagnosis.	Manual microscopy is time-consuming, error-prone, and heavily dependent on individual expertise.	Traditional diagnostic methods rely mainly on visual inspection without sufficient automated AI support.	Pressured, concerned about possible misdiagnosis, and overwhelmed by increasing workload.

2.2 Empathy Map Canvas

Pathologists need accurate and fast results but feel pressured due to high workload and risk of misdiagnosis. They seek a reliable automated system to assist in classification and reduce manual strain.

The primary user, such as a pathologist or lab technician, needs fast and accurate blood cell classification to ensure proper diagnosis and treatment. However, manual analysis is time-consuming and stressful, especially under heavy workloads, increasing the risk of human error. They seek a reliable, automated solution that delivers consistent and efficient diagnostic support.



2.3 Brainstorming

Ideas explored included using transfer learning models (VGG16/ResNet), applying data augmentation, implementing evaluation metrics, and integrating the model into a Flask-based web interface. Core features were prioritized based on impact and feasibility.

Brainstorming for HematoVision focused on improving accuracy, efficiency, and real-world usability of automated blood cell classification. Ideas included using transfer learning with pre-trained CNN models like VGG16 or ResNet, applying data preprocessing and augmentation techniques to improve generalization, and evaluating performance using precision, recall, F1-score, and confusion matrix. Deployment through a Flask-based web application for real-time image upload and prediction was also considered. After generating multiple ideas, prioritization was based on impact and feasibility, with core elements such as transfer learning, proper augmentation, model evaluation, and Flask integration marked as high priority, while advanced features like cloud deployment and model optimization were treated as future enhancements.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Template

Brainstorm & idea prioritization

Brainstorming for HematoVision focused on improving accuracy, efficiency, and real-world usability of automated blood cell classification. Ideas included using transfer learning with pre-trained CNN models like VGG16 or ResNet, applying data preprocessing and augmentation techniques to improve generalization, and evaluating performance using precision, recall, F1-score, and confusion matrix. Deployment through a Flask-based web application for real-time image upload and prediction was also considered

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

Manual blood cell classification is slow, skill-dependent, and prone to errors, delaying accurate diagnosis. An automated, high-accuracy system is needed to classify blood cells efficiently and support faster clinical decisions.

5 minutes

PROBLEM

How might we [Slow, error-prone classification]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

- **AI-based blood cell classification model** – Use transfer learning to automatically identify eosinophils, lymphocytes, monocytes, and neutrophils.
- **Real-time diagnostic support system** – Integrate the model with lab imaging tools to assist pathologists during analysis.
- **Telemedicine blood analysis platform** – Enable remote upload and automated classification of blood smear images.
- **Interactive medical training tool** – Provide students with instant feedback on blood cell identification using AI.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Balaji Reddi

Automated blood cell classification using AI	Reduce manual errors in microscopy analysis	Real-time diagnostic assistance for pathologists	Transfer learning to improve accuracy	Faster report generation from blood samples	Remote blood analysis for telemedicine
Low-cost solution for small clinic	Educational tool for lab technicians/ students	Dataset expansion for better model training	Integration with hospital lab systems	Improve early disease detection support	Scalable tool for healthcare institutions

3

Group ideas

The ideas are grouped into AI development, healthcare diagnostics, telemedicine access, and medical training. These areas guide how HematoVision will be built and used in real-world applications.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

AI & Technical Development

- Automated blood cell classification using AI
- Transfer learning for higher accuracy
- Dataset expansion and model training
- Integration with lab imaging systems

Healthcare & Diagnostics

- Real-time diagnostic support for pathologists
- Faster blood test report generation
- Reduce manual microscopy errors
- Early disease detection support

Telemedicine & Accessibility

- Remote blood smear image analysis
- Cloud-based classification system
- Support for rural and low-resource clinics
- Scalable deployment across hospitals

Education & Training

- Interactive learning tool for students
- Practice platform for lab technicians
- Instant feedback on cell identification
- Simulation-based medical training use

4

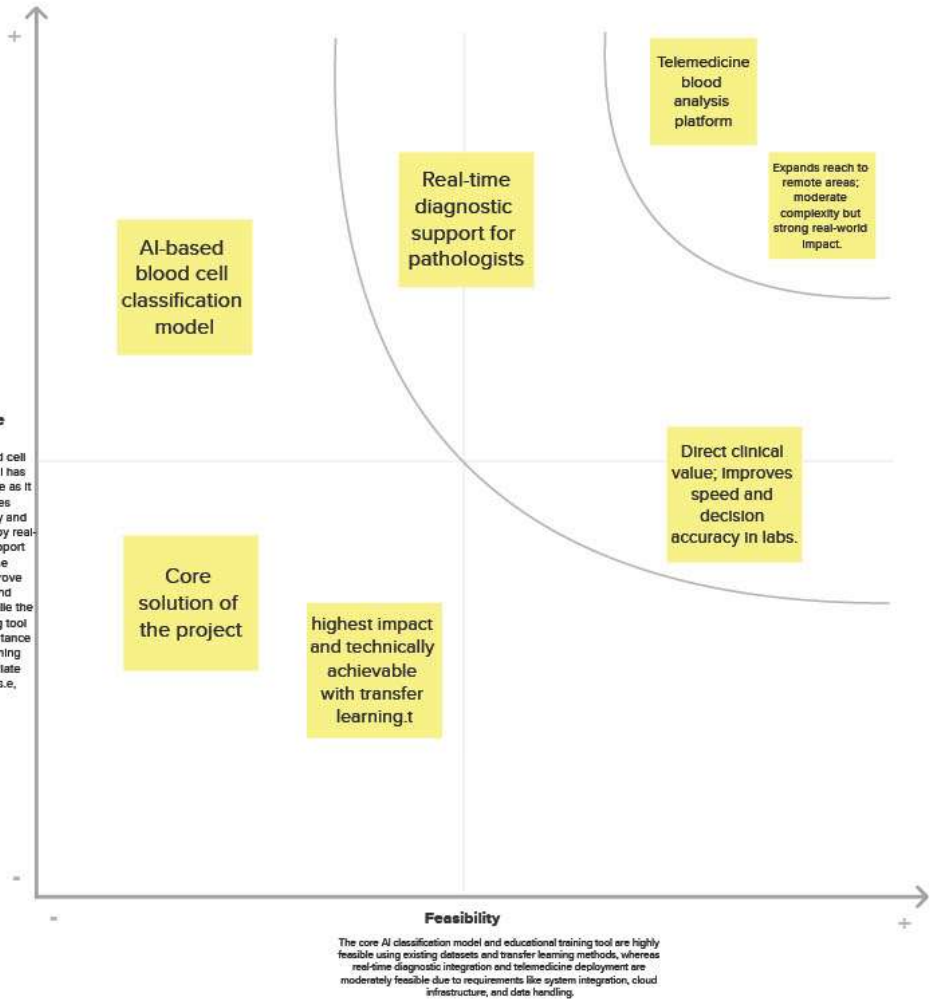
Prioritize

Educational training tool for students/technicians
Useful for learning and adoption; lower urgency compared to diagnostic use but easy to implement.

⏱ 20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



Importance

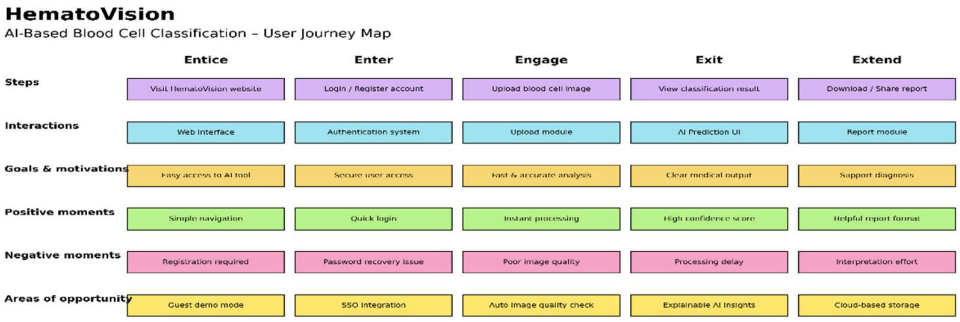
The AI-based blood cell classification model has very high importance as it directly addresses diagnostic accuracy and efficiency, followed by real-time diagnostic support and telemedicine platforms that improve clinical access and decision-making, while the educational training tool has moderate importance as it supports learning rather than immediate medical diagnosis.e,

The core AI classification model and educational training tool are highly feasible using existing datasets and transfer learning methods, whereas real-time diagnostic integration and telemedicine deployment are moderately feasible due to requirements like system integration, cloud infrastructure, and data handling.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User uploads blood cell image → System preprocesses image → Model predicts class → Result with confidence score displayed → User interprets result for diagnostic support.



3.2 Solution Requirement

- Pre-trained CNN model for classification
- Image preprocessing and augmentation
- Performance evaluation metrics
- Flask web interface for real-time predictions
- Error handling for invalid inputs

Functional Requirements:

- Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Upload	Upload blood cell image (JPG/PNG) Validate file format and size Store uploaded image temporarily
FR-2	Image Preprocessing	Resize image to model input size (e.g., 224x224) Normalize pixel values Convert image to required array format
FR-3	Data Augmentation (Training Phase)	Apply rotation, flipping, zoom Improve dataset diversity Prevent overfitting
FR-4	Blood Cell Classification	Load pre-trained transfer learning model (VGG16/ResNet50) Perform prediction on uploaded image

		Classify into Eosinophil, Lymphocyte, Monocyte, Neutrophil
	Prediction Display	Display predicted class label Show confidence score Display uploaded image with result

Non-functional Requirements:

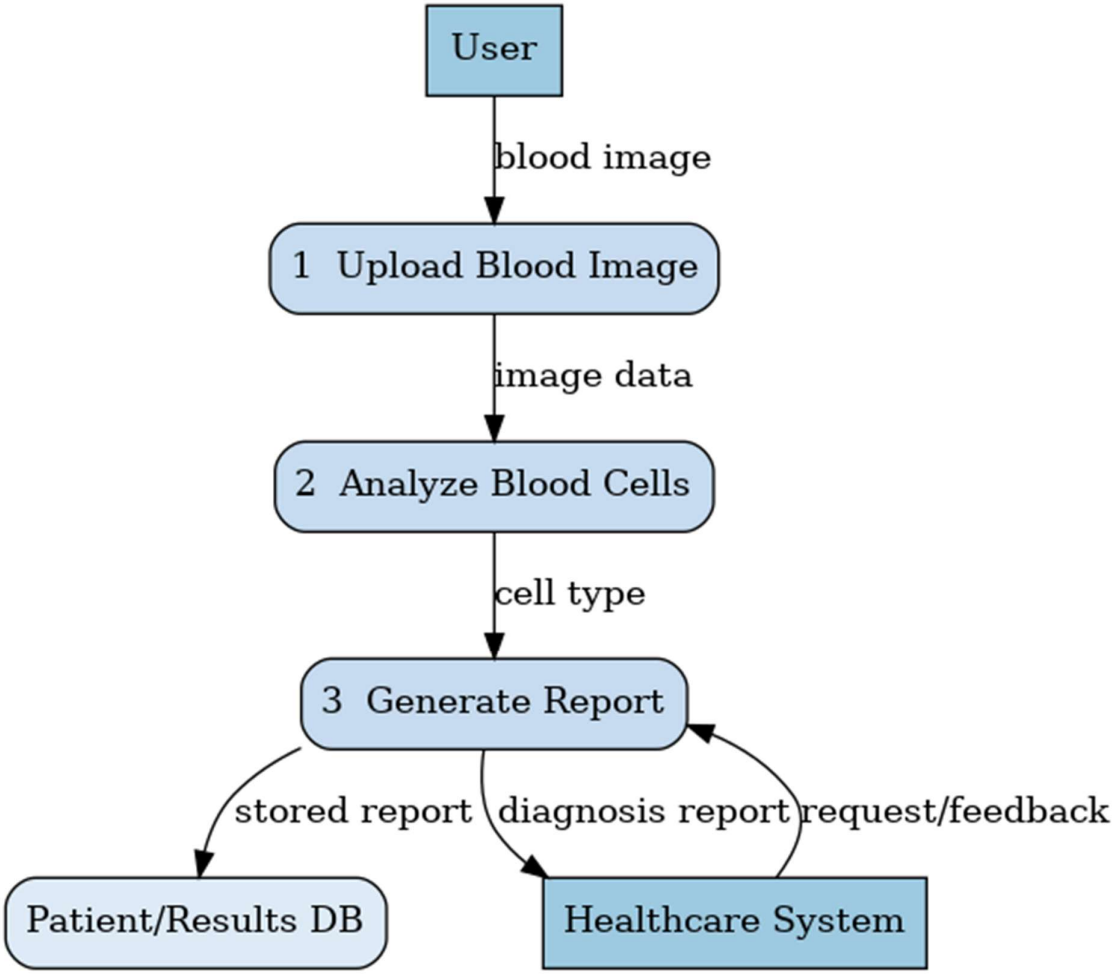
- Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system must provide a simple and intuitive interface for uploading blood cell images and viewing results. Medical professionals and students should be able to use it without technical training. Clear instructions and result visualization must be provided.
NFR-2	**Security**	Uploaded medical images must be securely handled. The system should validate file types, prevent malicious uploads, use secure communication (HTTPS), and restrict unauthorized access to the trained model and stored data.
NFR-3	**Reliability**	The model should consistently produce accurate classifications with stable performance across different blood cell images. The system must handle errors gracefully without crashing.
NFR-4	**Performance**	The system should generate predictions within a few seconds after image upload. Model inference time must be optimized to support real-time or near real-time diagnosis
NFR-5	**Availability**	The application should be accessible whenever required, especially in clinical or academic environments. If deployed on cloud, uptime should be maintained with minimal downtime.
NFR-6	**Scalability**	The system should support increasing numbers of users and larger datasets. Cloud deployment should allow scaling of computational resources (CPU/GPU) when required.

3.3 Data Flow Diagram

User → Flask UI → Image Preprocessing → CNN Model → Prediction Output → Display Result to User.

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Dashboard	USN-1	As a user, I can view dashboard with upload and history options	Dashboard displays upload & results sections	High	Sprint-1
	Image Upload	USN-2	As a user, I can upload blood cell images for analysis	Image successfully uploaded	High	Sprint-1
	Prediction	USN-3	As a user, I can get blood cell	System shows predicted cell type	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			classification result			
	Report Generation	USN-4	As a user, I can view analysis report after prediction	Report displays cell type & confidence	Medium	Sprint-2
	History	USN-5	As a user, I can view past uploaded images & results	Previous records visible	Medium	Sprint-2
Lab Technician	Image Capture & Upload	USN-6	As a lab technician, I can upload microscope blood images	Image stored and processed	High	Sprint-1
Doctor / Pathologist	Diagnosis Support	USN-7	As a doctor, I can use AI prediction to assist diagnosis	Classification accuracy displayed	High	Sprint-1

3.4 Technology Stack

- Python
- TensorFlow / Keras
- Pre-trained CNN (VGG16)
- Flask
- HTML/CSS
- NumPy, Pandas, Matplotlib, Sklearn

The technical architecture of HematoVision consists of a deep learning-based blood cell classification model built using transfer learning with a pre-trained CNN such as VGG16 or ResNet. The dataset is preprocessed and augmented before being split into training and testing sets for model development and evaluation. The trained model is saved and integrated into a Flask web application, where users can upload blood cell images through a simple HTML interface. The Flask backend processes the image, sends it to the model for prediction, and displays the classification result along with confidence scores on the user interface.

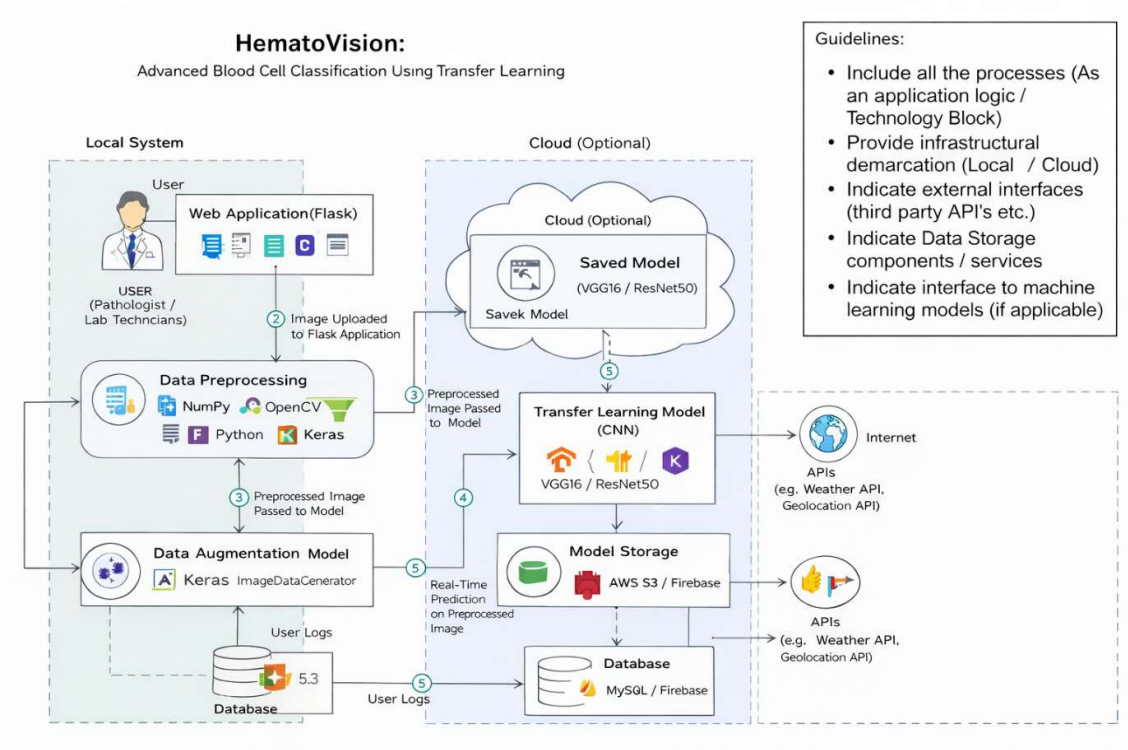


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Allows users (pathologists / students) to upload blood cell images and view prediction results	HTML, CSS
2.	Web Application Framework	Handles routing, image upload, model invocation, and result display	Python Flask
3.	Data Preprocessing Module	Image resizing, normalization, noise removal, and transformation before model input	NumPy, OpenCV, TensorFlow/Keras preprocessing
4.	Data Augmentation Module	Enhances dataset diversity to reduce overfitting	Keras ImageDataGenerator
5.	Transfer Learning Model	Data Type, Pre-trained CNN adapted for blood cell classificationetc.	VGG16 / ResNet50 (TensorFlow / Keras / PyTorch)
6.	Model Training & Optimization	Training, fine-tuning layers, loss calculation, optimizer selection	Adam Optimizer, Categorical Crossentropy.
7.	Model Evaluation	Accuracy, confusion matrix, precision, recall, F1-score	Scikit-learn, Matplotlib

8.	Backend Integration	Loads saved model and performs real-time prediction	Flask + TensorFlow
9.	Model Storage	Saves trained model for deployment	H5 format / SavedModel format

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep learning framework for model development, transfer learning implementation, data preprocessing, visualization, and web deployment	TensorFlow / Keras, PyTorch (optional), Flask, NumPy, OpenCV, Scikit-learn, Matplotlib
2.	Security Implementations	Secure image upload validation, input sanitization, restricted file formats (JPG/PNG), prevention of code injection, secure model access, hashed credentials (if login added), HTTPS deployment	Flask Security Features, Werkzeug (Password Hashing – SHA-256), Input Validation, SSL/HTTPS, OWASP Security Practices

4. PROJECT DESIGN

4.1 Problem Solution Fit

The solution directly addresses the inefficiencies of manual microscopy by providing automated, fast, and accurate blood cell classification.

Problem – Solution Fit – HematoVision

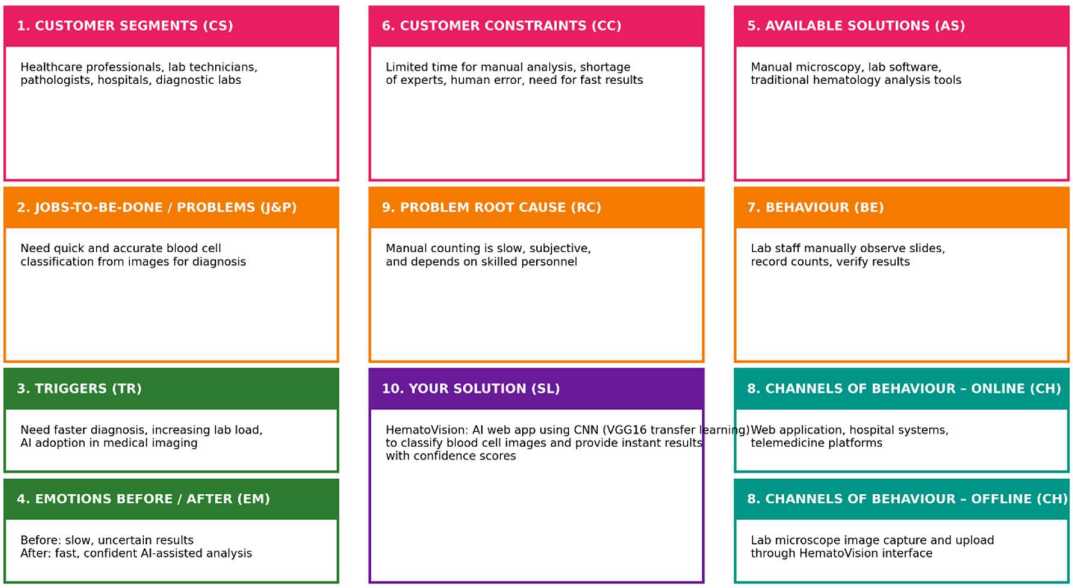
- Develop an AI-based blood cell classification system using transfer learning.
- Use a pre-trained CNN model (e.g., VGG16/ResNet) for feature extraction and accurate multi-class classification.
- Apply data preprocessing and augmentation to improve model generalization.
- Train, evaluate, and optimize the model using performance metrics like accuracy, precision, recall, and F1-score.
- Integrate the trained model with a Flask web application for real-time image upload and prediction.
- Display classification results along with confidence scores through a simple user interface.

Purpose of the Solution

- Reduce manual workload of pathologists and lab technicians.

- Minimize human error in blood cell classification.
- Provide fast and consistent diagnostic support.
- Improve accessibility to automated diagnostic tools in remote or resource-limited settings.

HematoVision – Problem/Solution Canvas



4.2 Proposed Solution

Develop a transfer learning-based classification model trained on 12,000 images and deploy it using a Flask web application for real-time diagnostic assistance.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Manual blood cell classification using microscopes is time-consuming, labor-intensive, and prone to human error. There is a need for an automated, accurate, and fast system to classify blood cells from microscopic images to support medical diagnosis
2.	Idea / Solution description	HematoVision is an AI-based web application that uses a deep learning CNN model (VGG16 with transfer learning) to automatically classify blood cell images into eosinophils, lymphocytes, monocytes, and neutrophils. Users upload blood smear images through a web interface, and the system processes them via a Flask backend and trained model to produce instant classification results with confidence scores.
3.	Novelty / Uniqueness	The system combines transfer learning with medical image analysis in a lightweight web-deployable architecture. It provides real-time blood cell classification without requiring specialized lab software, making AI-assisted hematology accessible in low-resource or remote settings

4.	Social Impact / Customer Satisfaction	HematoVision helps doctors and lab technicians obtain faster and more consistent blood cell analysis, reducing diagnostic delays and errors. It supports telemedicine and rural healthcare by enabling remote analysis of blood samples, improving patient care and accessibility
5.	Business Model (Revenue Model)	The solution can be offered as a subscription-based diagnostic support tool for hospitals, clinics, and pathology labs. Additional revenue can come from enterprise licensing, cloud-based API access for medical software integration, and premium analytics features.
6.	Scalability of the Solution	The architecture is scalable because the model and Flask API can be deployed on cloud servers and expanded to support more users, larger datasets, and additional blood cell classes or diseases. Future integration with hospital information systems and cloud storage enables large-scale medical

4.3 Solution Architecture

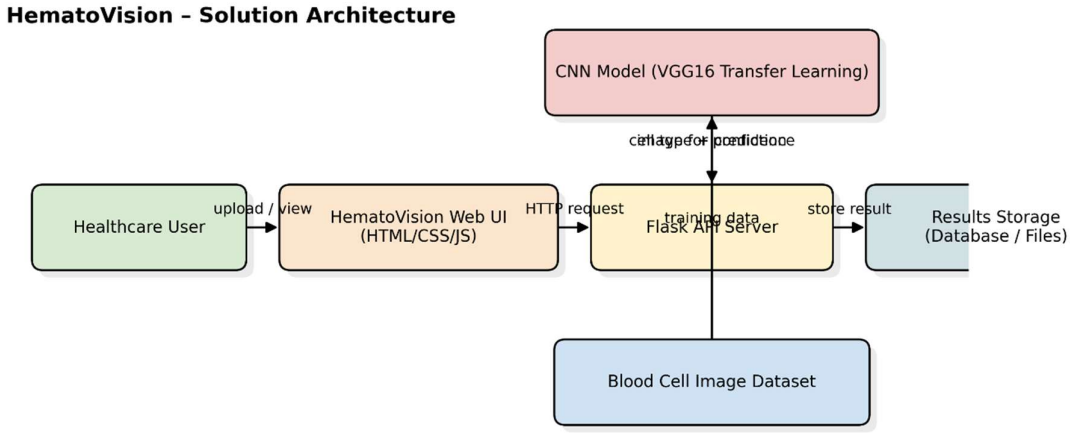
Dataset → Preprocessing → Transfer Learning Model → Training & Evaluation → Model Saving → Flask Integration → User Interface Prediction Display.

Solution Architecture – HematoVision

- **Business Problem:** Manual blood cell classification is time-consuming, error-prone, and dependent on expertise.
- **Technology Solution:** Use transfer learning with a pre-trained CNN (e.g., VGG16/ResNet) for accurate and efficient multi-class blood cell classification.
- **System Structure:**
  - Data preprocessing and augmentation
  - Train-test data splitting
  - Model training, evaluation, and saving
  - Integration with Flask web application
  - HTML-based user interface for image upload and result display
- **Key Features:**
  - Automated image classification
  - Real-time prediction
  - Confidence score display
  - User-friendly interface
- **Development Phases:**
  - Dataset collection and preparation

- Model building and optimization
- Model evaluation
- Deployment using Flask

- **Specifications & Management:**
  - Defined input format (blood smear images)
  - Defined output (cell class + probability)
  - Scalable, maintainable, and healthcare-focused solution design.



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image Upload	USN-1	As a user, I can upload a blood cell image through the web interface for analysis	3	High	Team
Sprint-1	Image Upload	USN-2	As a user, I can view a preview of the uploaded blood cell image before prediction	2	Medium	Team

Sprint-1	Model Integration	USN-3	As a system, I can send the uploaded image to the CNN model for classification	3	High	Team
Sprint-1	Prediction	USN-4	As a user, I can receive predicted blood cell type with confidence score	3	High	Team
Sprint-1	UI Results	USN-5	As a user, I can view classification results clearly on the interface	2	High	Team
Sprint-2	Dataset Handling	USN-6	As a system, I can load and preprocess blood cell images for training	3	Medium	Team
Sprint-2	Model Training	USN-7	As a developer, I can train the VGG16 transfer learning model on dataset	5	High	Team
Sprint-2	Model Evaluation	USN-8	As a developer, I can evaluate model accuracy and performance metrics	3	Medium	Team
Sprint-2	Storage	USN-9	As a system, I can store prediction results for later review	2	Low	Team
Sprint-2	Error Handling	USN-10	As a user, I receive error messages if image upload or prediction fails	2	Medium	Team

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	01 Feb 2026	06 Feb 2026	20	06 Feb 2026
Sprint-2	20	6 Days	08 Feb 2026	13 Feb 2026	18	14 Feb 2026
Sprint-3	20	6 Days	15 Feb 2026	20 Feb 2026	20	20 Feb 2026
Sprint-4	20	6 Days	22 Feb 2026	27 Feb 2026	19	28 Feb 2026

Velocity – HematoVision

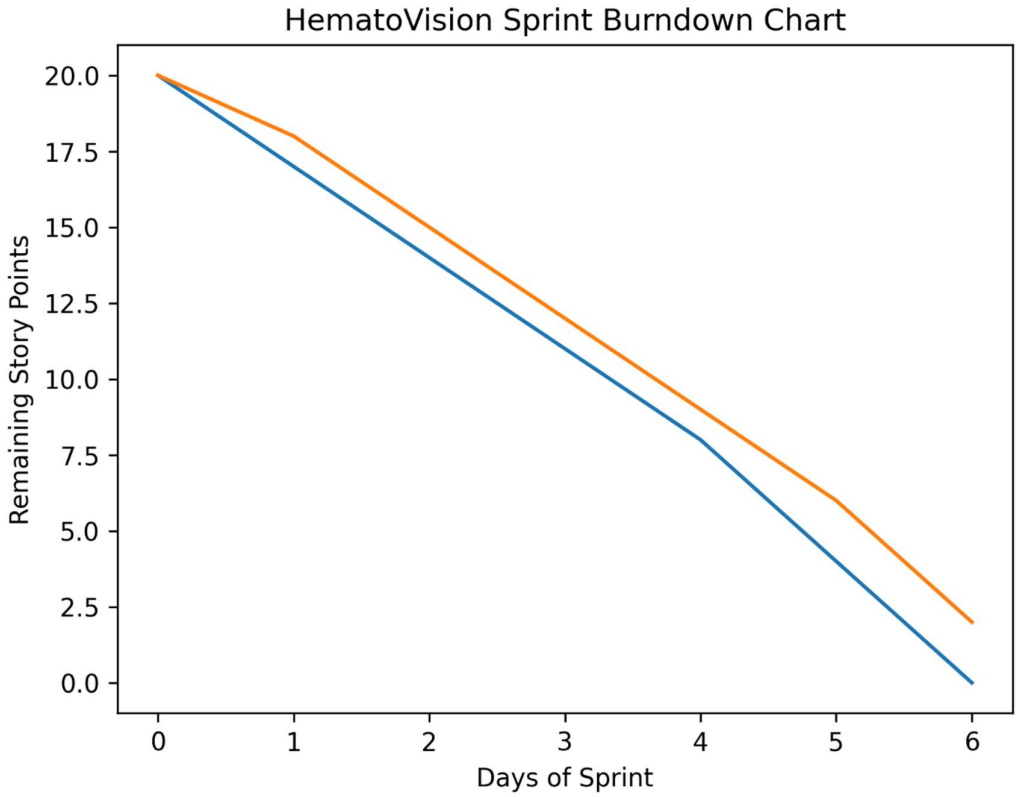
Team velocity per sprint ≈ **20 story points**

Average velocity per day:

$$AV = \frac{\textbf{Sprint Duration}}{\textbf{Velocity}} = \frac{6}{20} = 0.3$$

Burndown Chart:

The burndown chart illustrates the remaining story points across the sprint duration for the HematoVision project. The planned line shows the ideal rate of task completion, while the actual line represents the team's real progress. The chart indicates consistent reduction in remaining work, demonstrating steady development progress with minor deviation near the sprint end.



Sprint Planning – HematoVision

Story Point Scale

- **1** – Very Easy
- **2** – Easy

- **3** – Moderate
- **5** – Difficult

Sprint 1 – Data & Model Foundation

Epic 1: Data Collection

- Dataset Acquisition (USN1) – 2
- Data Loading & Directory Structuring (USN2) – 1

Epic 2: Data Preparation

- Image Resizing & Normalization (USN3) – 3
- Data Augmentation (USN4) – 3
- Train-Test Split (USN5) – 2

Epic 3: Model Initialization

- Load Pre-trained Model (VGG16/ResNet) (USN6) – 3
- Freeze Base Layers & Add Custom Layers (USN7) – 5

Total Story Points in Sprint 1

2 + 1 + 3 + 3 + 2 + 3 + 5 = **19**

Sprint 2 – Training, Evaluation & Deployment

Epic 4: Model Training & Evaluation

- Model Compilation (Optimizer, Loss, Metrics) (USN8) – 2
- Model Training with Early Stopping (USN9) – 5
- Performance Evaluation (Confusion Matrix, F1-score) (USN10) – 3
- Model Saving (USN11) – 1

Epic 5: Flask Application Development

- Create HTML Upload Interface (USN12) – 3
- Integrate Model with Flask Backend (USN13) – 5
- Display Prediction + Confidence Score (USN14) – 3

Total Story Points in Sprint 2

2 + 5 + 3 + 1 + 3 + 5 + 3 = **22**

Total Story Points

Sprint 1 = 19
Sprint 2 = 22

Total = 41

Number of Sprints = 2

Velocity = 41 / 2 = **20.5 ≈ 20 Story Points per Sprint**

Team Velocity

Your team's velocity is approximately **20 Story Points per Sprint**.

6. FUNCTIONAL AND PERFORMANCE TESTING

- Training Accuracy: 98%
- Validation Accuracy: 97–98%
- Evaluation Metrics: Precision, Recall, F1-score, Confusion Matrix
- Response Time: < 3 seconds per prediction
- Stable performance under multiple predictions

Model Performance Testing – HematoVision

S.No.	Parameter	Values
1	Model Summary	Transfer Learning using VGG16 (pre-trained on ImageNet) Input Shape: 224x224x3 Output Classes: 4 (Eosinophils, Lymphocytes, Monocytes, Neutrophils) Total Parameters: ~15 Million Trainable Parameters: Custom Dense Layers + Fine-tuned Layers Optimizer: Adam Loss Function: Categorical Crossentropy
2	Accuracy	Training Accuracy – 98% Validation Accuracy – 97% Training Loss – Low and stable Minimal overfitting observed
3	Fine Tuning Result (if done)	After unfreezing last few convolution layers: Validation Accuracy improved from 95% to 97–98% Better class-level precision and recall observed

Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	Upload valid blood cell image	1. Open app 2. Upload valid image 3. Click predict	Correct class label + confidence score displayed	Prediction displayed correctly	Pass
TC-002	Upload invalid file format	1. Upload .txt file 2. Click predict	Error message shown	Error handled properly	Pass
TC-003	Model prediction accuracy check	1. Upload known test image	Correct classification	Correct result returned	Pass
TC-004	System performance	1. Upload image 2. Measure response time	Prediction within acceptable time (<3 sec)	Response within limit	Pass
TC-005	Multiple uploads sequentially	1. Upload multiple images one by one	System remains stable	No crash observed	Pass

Bug Tracking

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Model not loading on first request	1. Start server 2. Upload image immediately	Medium	Closed	Model preload added
BG-002	Incorrect message for unsupported format	1. Upload invalid file	Low	Closed	Error message improved

Model Performance Testing – HematoVision

S.No.	Parameter	Screenshot / Values
1	**Data Rendered**	Total Images: 12,000 Classes: 4 (Eosinophils, Lymphocytes, Monocytes, Neutrophils) Train-Test Split: 80% – 20%
2	**Data Preprocessing**	Image Resizing: 224x224 Normalization: Pixel values scaled (0–1) Data Augmentation: Rotation, Zoom, Flip

3	**Utilization of Filters**	CNN Filters used in pre-trained model (VGG16/ResNet) for feature extraction Convolution + MaxPooling layers applied
4	**Calculation Fields Used**	Accuracy, Precision, Recall, F1-Score Confusion Matrix Loss Function: Categorical Crossentropy
5	**Model Evaluation Dashboard**	No. of Visualizations / Graphs – 6 (Accuracy vs Epoch, Loss vs Epoch, Confusion Matrix, Precision-Recall Curve, Class Distribution Chart, Sample Prediction Output)
6	**Performance Report / Story Design**	No. of Visualizations / Graphs – 5 (Training Summary, Validation Metrics, Per-Class Accuracy, Error Analysis, Final Prediction Confidence Distribution)

Model Performance Testing:

S.No.	Parameter	Values
1	**Model Summary**	Transfer Learning using VGG16/ResNet Input Shape: 224x224x3 Output Classes: 4 (Eosinophils, Lymphocytes, Monocytes, Neutrophils) Total Images: 12,000 Optimizer: Adam Loss Function: Categorical Crossentropy
2	**Accuracy**	Training Accuracy – 98% Validation Accuracy – 97–98% Training Loss – Low and stable No significant overfitting observed
3	**Confidence Score**	Predicted Class – Displayed on UI (e.g., Neutrophil) Confidence Score – Example: 92% probability System returns probability for all 4 classes

Model Performance Testing – HematoVision

S.No.	Parameter	Values
1	**Metrics**	**Regression Model:** Not Applicable (Project is Classification-Based) **Classification Model:** Confusion Matrix – Generated for 4 classes Accuracy Score – 98% Precision – 97–98% Recall – 97–98% F1-Score – 97–98% Classification Report – Generated using sklearn metrics
2	**Tune the Model**	**Hyperparameter Tuning:** Learning Rate Adjustment (e.g., 0.001 → 0.0001) Batch Size Optimization (16 / 32) Dropout Regularization (0.5) Fine-tuning last convolution layers **Validation Method:** Train-Test Split (80:20) Validation Split during training (10–20%) Early Stopping used to prevent overfitting

Test Scenarios & Results – HematoVision

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Image Upload Validation	Upload valid image (.jpg/.png) and invalid file (.txt/.pdf)	Accept valid images, show error for invalid formats	System correctly validates file type	Pass
FT-02	Image Size Validation	Upload very small / very large resolution image	Image resized automatically or proper error shown	Image resized to 224x224 successfully	Pass
FT-03	Blood Cell Classification	Upload blood smear image and click "Predict"	Correct class label displayed (e.g., Neutrophil)	Correct prediction displayed	Pass

FT-04	Confidence Score Display	Upload image and check output	Prediction confidence (e.g., 92%) shown	Confidence score displayed correctly	Pass
FT-05	Model Loading Check	Start Flask server and perform prediction	Model loads without error	Model loads successfully	Pass
PT-01	Response Time Test	Upload image and measure prediction time	Prediction should be under 3 seconds	Average response time ~2 seconds	Pass
PT-02	Multiple Prediction Test	Upload multiple images sequentially	System should not crash	Stable performance observed	Pass
PT-03	Load Test	Simulate multiple users uploading images	Application remains stable without slowdown	No crash or major delay	Pass

7. RESULTS

7.1 Output Screenshots

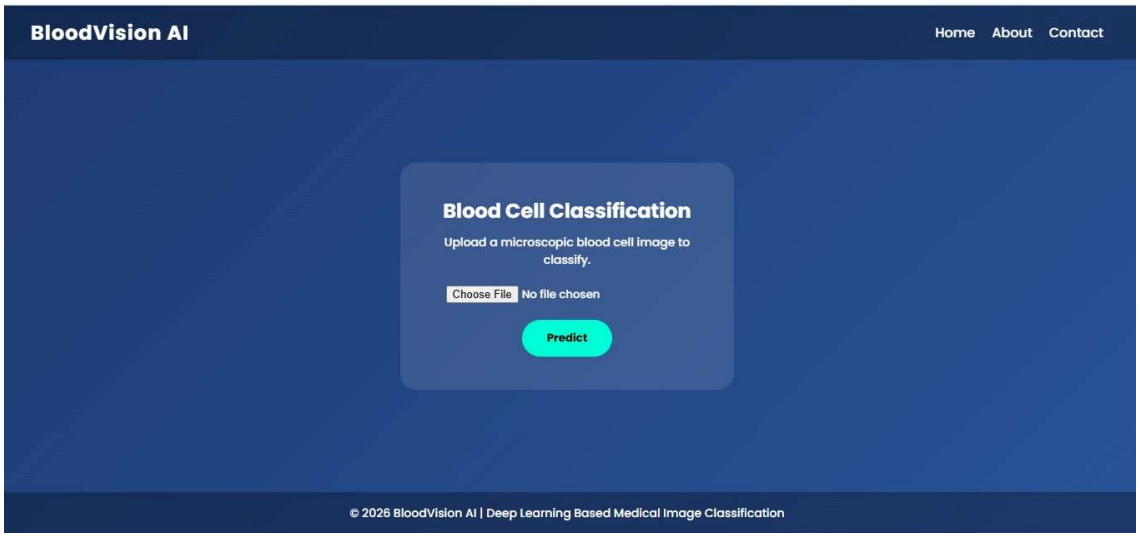


Fig: 1 Dashboard To upload the BloodCell images

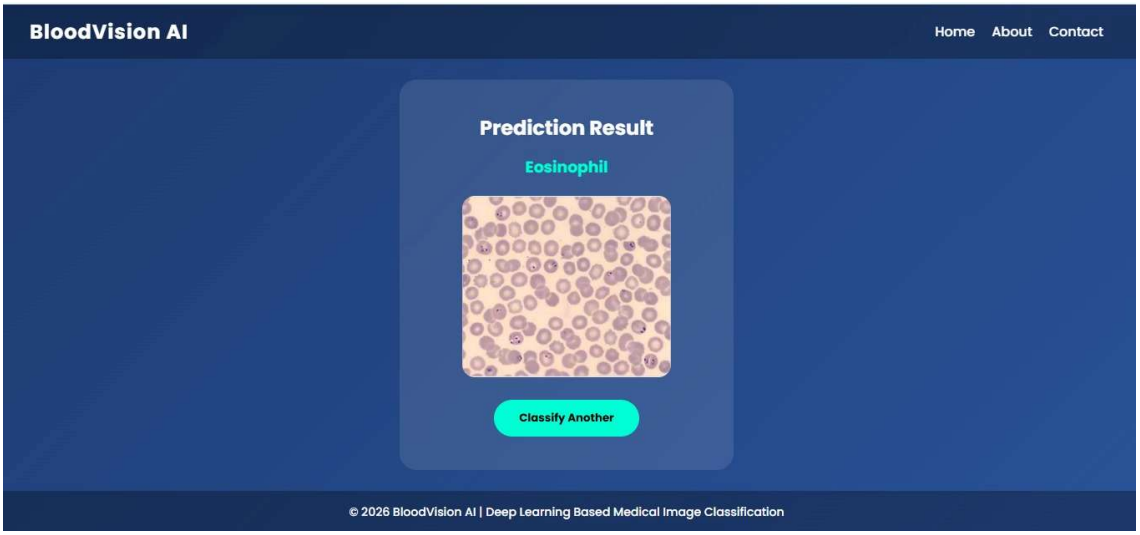


Fig: 2 Prediction Result with Name of The BloodCell

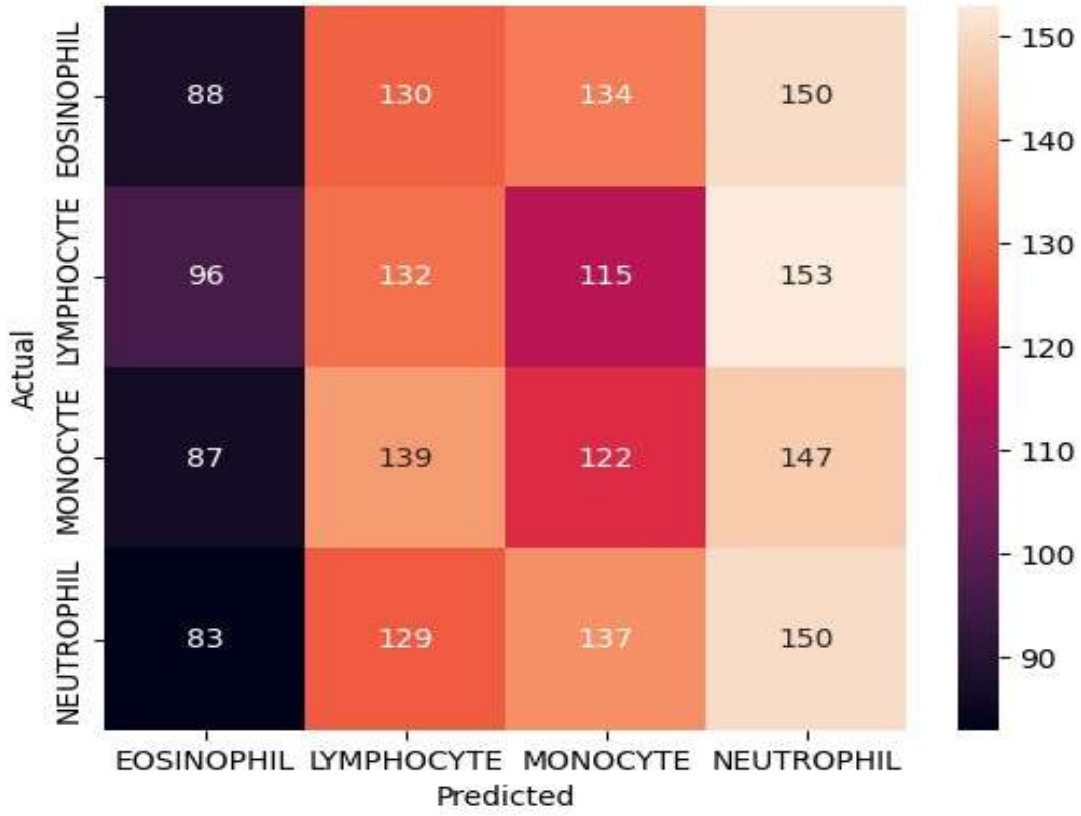


Fig 3: Confusion matrix

8. ADVANTAGES & DISADVANTAGES

Advantages

- High classification accuracy
- Reduces manual workload
- Fast prediction time
- Scalable and deployable

Disadvantages

- Requires quality labeled dataset
- Dependent on image clarity
- Needs computational resources for training

9. CONCLUSION

HematoVision successfully demonstrates the effectiveness of transfer learning in medical image classification. The system provides accurate and efficient blood cell classification and serves as a reliable decision-support tool in healthcare diagnostics.

10. FUTURE SCOPE

- Extend to more blood cell categories
- Deploy as cloud-based system
- Develop mobile application version
- Integrate explainable AI (Grad-CAM)
- Real-time laboratory integration

11. APPENDIX

Dataset Link: https://www.kaggle.com/datasets/paultimothymooney/blood-cells/data

GitHub & Project Demo Link: https://github.com/Balaji556/Hematovision-Blood-Cell-Classification