

# HematoVision: AI-powered Blood Cell Classification System

## INTRODUCTION

HematoVision is a deep learning-powered web application for classifying blood cells from microscopic images. It leverages Transfer Learning techniques to automatically detect and classify cell types such as RBCs, WBCs (lymphocytes, monocytes, neutrophils, eosinophils), and platelets. The goal is to automate and expedite diagnostics for diseases like leukemia and anemia.

## PURPOSE

The purpose of this project is to assist medical professionals in diagnosing blood-related disorders using an AI-driven tool that classifies blood cells with high accuracy. HematoVision aims to:

- Automate the blood smear analysis process
- Reduce manual workload in clinical labs
- Ensure faster and more consistent diagnostic results
- Make diagnostics accessible in resource-limited areas

## FEATURES

1. Upload-based microscopic image classification
2. Real-time predictions with confidence scores
3. Use of transfer learning with VGG16, ResNet50, and MobileNetV2
4. Graphs showing accuracy/loss per epoch
5. Confusion matrix and evaluation metrics
6. Responsive UI with Streamlit
7. Lightweight model for fast inference

## DESCRIPTION

HematoVision is a medical AI application built using Python, TensorFlow, and Streamlit. Users can upload

# HematoVision: AI-powered Blood Cell Classification System

microscopic images, and the model classifies the blood cell type. It supports three deep learning architectures and provides accuracy-based feedback. The web app is user-friendly and suitable for clinical use.

## SCENARIO

Dr. Ramesh, a rural healthcare worker, receives a patient's blood smear slide and uses HematoVision to upload the image. Within seconds, the system classifies the cell as a 'Neutrophil' with 94% confidence. This assists him in diagnosing the infection quickly and forwarding the report digitally. HematoVision enables fast triage and reduces dependency on expert pathologists.

## TECHNICAL ARCHITECTURE

- Frontend: Streamlit web application with image upload UI
- Backend: Python-based classification using TensorFlow/Keras
- Models Used: ResNet50, VGG16, MobileNetV2
- Hosting: Streamlit sharing / Flask optional
- Flow: Upload -> Preprocess -> Predict -> Display Results

## ER DIAGRAM

- No traditional ER diagram; however, the system architecture maps user input to model output via preprocessing and inference pipeline.

User -> Upload Image -> Resize/Normalize -> Predict Cell Type -> Show Output

## SETUP PHASE

1. Google Colab for training models
2. Dataset from Kaggle (BCCD Dataset)
3. Libraries: TensorFlow, Keras, NumPy, Matplotlib, Streamlit

# HematoVision: AI-powered Blood Cell Classification System

4. Training Image Size: 224x224
5. Deployment: Streamlit app with saved model (.h5)

## PROJECT STRUCTURE

- model/: Contains trained model files
- app/: Streamlit scripts (main.py, utils.py)
- data/: Blood smear dataset
- notebooks/: Model training notebooks
- requirements.txt: Dependencies

## APPLICATION FLOW

1. User uploads an image
2. Image is resized, normalized
3. Model performs inference
4. Prediction with class and confidence is shown
5. Metrics and heatmaps (if selected) are displayed

## RESULTS & TESTING

- ResNet50 achieved highest accuracy (93.1%)
- Functional Testing: Uploads, outputs work
- Performance Testing: <2 seconds inference
- Visual Output: Confusion Matrix, Accuracy vs Epochs, Loss Graph

## ADVANTAGES & LIMITATIONS

Advantages:

- Fast and accurate classification

# **HematoVision: AI-powered Blood Cell Classification System**

- Minimal human intervention
- Web-friendly

Limitations:

- Sensitive to poor image quality
- Less accurate on rare cell types

## **CONCLUSION**

HematoVision validates the power of Transfer Learning in medical diagnostics. With high classification accuracy and a simple web interface, it reduces time and effort in lab analysis. It is suitable for hospitals, clinics, and remote diagnostics.

## **FUTURE SCOPE**

- Add leukemia subtypes
- Deploy on mobile with offline inference
- Integrate with microscopes via camera
- Include Grad-CAM visual explanations

## **TEAM DETAILS**

Team Members:

- Nayudipalli Dayakar
- Nakka Jagan Satya Sai Krishna Mahesh
- Nadella Sindhura
- Nadakuditi Bindu