Smart Bridge Virtual Internship Program2025

PatternSense

ClassifyingFabricPatternsUsingDeepLearning

Submittedby

TeamID: LTVIP2025TMID43759

Nuthakki Sreekar

Ranga Rao Chowdary

Lasa Sravani

M Althaf Vali

M Lavanya

Andhra Pradesh,India

June, 2025

Contents

Phase1:Brainstorming and Ideation

Problem Statement

Blood cell analysis is critical for diagnosing numerous diseases, including infections, leukemia, anemia, and other hematological disorders. Traditional manual examination under the microscope is time-consuming, subjective, and requires specialized expertise. With the increasing availability of digital microscopy and medical imaging, there is a strong need for an automated, accurate system to classify blood cell types quickly and reliably. Our project addresses this challenge by leveraging transfer learning to classify blood cells into four categories: eosinophil, lymphocyte, monocyte, and neutrophil, improving both efficiency and diagnostic accuracy in clinical workflows.

Proposed Solution

We propose a transfer learning approach using a pre-trained MobileNetV2 convolutional neural network as the backbone. The model is fine-tuned on a blood cell image dataset to accurately distinguish between different blood cell types. The solution is wrapped into a web application using Flask, allowing users to upload images and receive instant predictions and visual feedback.

Targeted Users

- Hematologists and pathologists
- Medical laboratory technologists
- Hospitals and diagnostic centers
- Medical researchers studying hematological diseases

Expected Outcomes

A lightweight, accurate, and scalable blood cell classification system capable of real-time predictions and integration into laboratory or clinical decision support systems.

Phase2:Requirement Analysis

Technical Requirements

- Python programming environment
- Libraries:
  - TensorFlow
  - OpenCV
  - NumPy
  - Flask
  - Matplotlib
- Data source:

  Digital images of peripheral blood cells

Functional Requirements

- The system should accept uploaded images in formats such as JPG or PNG.
- Preprocess images to match model input dimensions.
- Predict blood cell class with high accuracy.
- Display the original image alongside the predicted class in a user-friendly web interface.

.

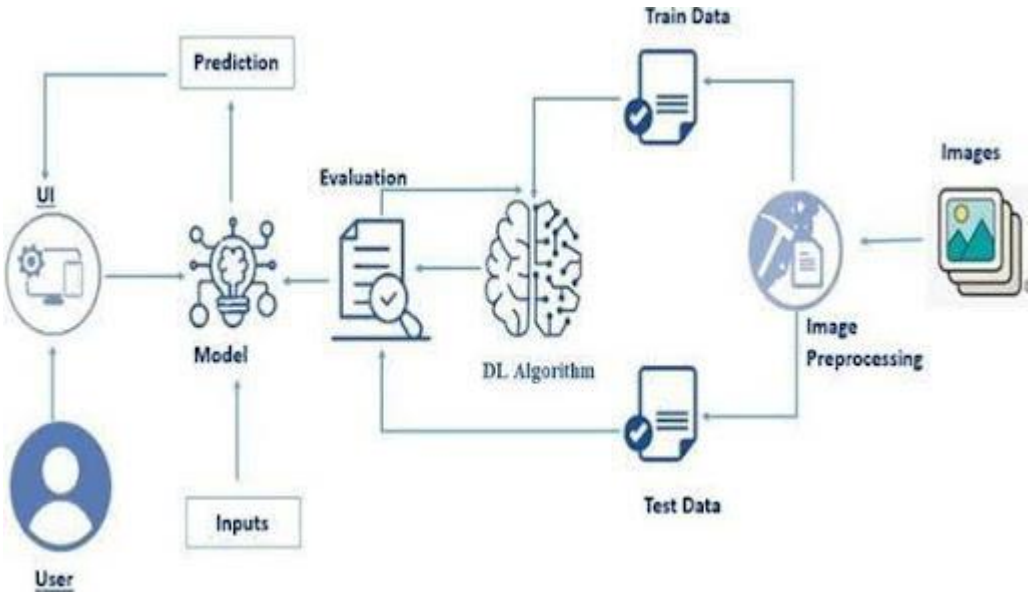
Constraints & Challenges

- Limited labeled datasets for certain rare blood cell types.
- High visual similarity between some cell types increases classification difficulty.
- Need for efficient preprocessing to maintain web application responsiveness.

Phase3:ProjectDesign

SystemArchitecture

- Step 1: User uploads blood cell image via web UI.
- Step 2: Image preprocessing using OpenCV and Keras preprocess_input.
- Step 3: Model predicts the blood cell class.
- Step 4: Result displayed with the cell type and visual feedback.



UserFlow

User uploads image → Server preprocesses image → Model predicts cell class → Web app displays prediction and image

UI/UX Considerations

- Simple, clean interface.
- Clear prompts for file upload.
- Visual display of uploaded image and predicted label.

- Error handling for unsupported file types or missing files.

Phase4:Project Planning(AgileMethodologies)

Objective

To develop a robust, intelligent system capable of classifying peripheral blood cell images into various cell types using deep learning and transfer learning techniques, aiming to assist medical professionals in the diagnosis of hematological conditions.

SprintPlanning

- Sprint 1: Data collection and dataset curation
- Sprint 2: Model selection and initial training
- Sprint 3: Integration into Flask web app
- Sprint 4: Testing, performance evaluation, and documentation

---

Task Allocation

- Nuthakki Sreekar Ranga Rao Chowdary – Data preprocessing and augmentation
- Lasa Sravani – Model fine-tuning and training
- M Althaf Vali – Web app development (Flask integration)
- M Lavanya – Evaluation, metrics analysis, and report writing

Timeline&Milestone

Week	Milestone
Week 1	Finalized problem statement and collected 12,000 annotated blood cell images
Week 2	Performed image preprocessing and augmentation techniques
Week 3	Implemented transfer learning using ResNet50 and began model training
Week 4	Evaluated model performance and tuned hyperparameters
Week 5	Developed the frontend UI using Streamlit and integrated it with the model
Week 6	Conducted functional and performance testing; fixed bugs and improved UI
Week 7	Deployed application on local server and prepared documentation
Week 8	Final review and project submission

Phase5:ProjectDevelopment

Introduction

HematoVision is a deep learning-based system developed to classify blood cell images into distinct categories: eosinophils, lymphocytes, monocytes, and neutrophils. The user interacts through a simple UI to upload a blood cell image. The system preprocesses the image and forwards it to a pre-trained deep learning model. Leveraging transfer learning, the model predicts the blood cell type and returns the result. This prediction is evaluated and displayed to the user, enabling quick, accurate diagnosis support for medical professionals.

TechnologyStackUsed

Programming Language: Python

Deep Learning Framework: TensorFlow, Keras

Model Architecture: ResNet50 (Transfer Learning)

Web Framework: Streamlit

Frontend: Streamlit Interface

Development Environment: Google Colab, VS Code

DevelopmentProcess

The development of HematoVision followed a structured approach:

1. **Data Collection and Preprocessing:**
   12,000 annotated images were cleaned, resized, normalized, and split into training, validation, and test sets.
2. **Model Selection and Training:**
   Pre-trained CNN models such as ResNet50 were fine-tuned using transfer learning on the blood cell dataset to boost classification accuracy.
3. **Evaluation and Tuning:**
   The model was evaluated using metrics like accuracy, precision, and recall. Hyperparameters were tuned to reduce overfitting.
4. **Interface Development:**
   A user-friendly interface was built using Streamlit, allowing users to upload images and receive predictions.
5. **Integration and Testing:**
   The model was integrated with the UI, and the end-to-end system was tested for performance and usability.

Challenges&Fixes

SeveralchallengeswereencounteredduringthedevelopmentofPatternSense:

- **Class Imbalance:** Some blood cell types were underrepresented in the dataset. Data augmentation was used to balance the dataset.

- **Overfitting:** Regularization and dropout techniques were implemented to reduce overfitting during training.

- **Deployment Delay:** Model loading time in Streamlit was optimized by saving models in `.h5` format and reducing image size during upload.

- **Prediction Errors:** Minor misclassifications were addressed by improving preprocessing and retraining with more representative data.

Despite these hurdles, the final system was successfully developed and deployed withsatisfactory performance and usability.

Phase6:FunctionalandPerformanceTesting

TestCases Executed

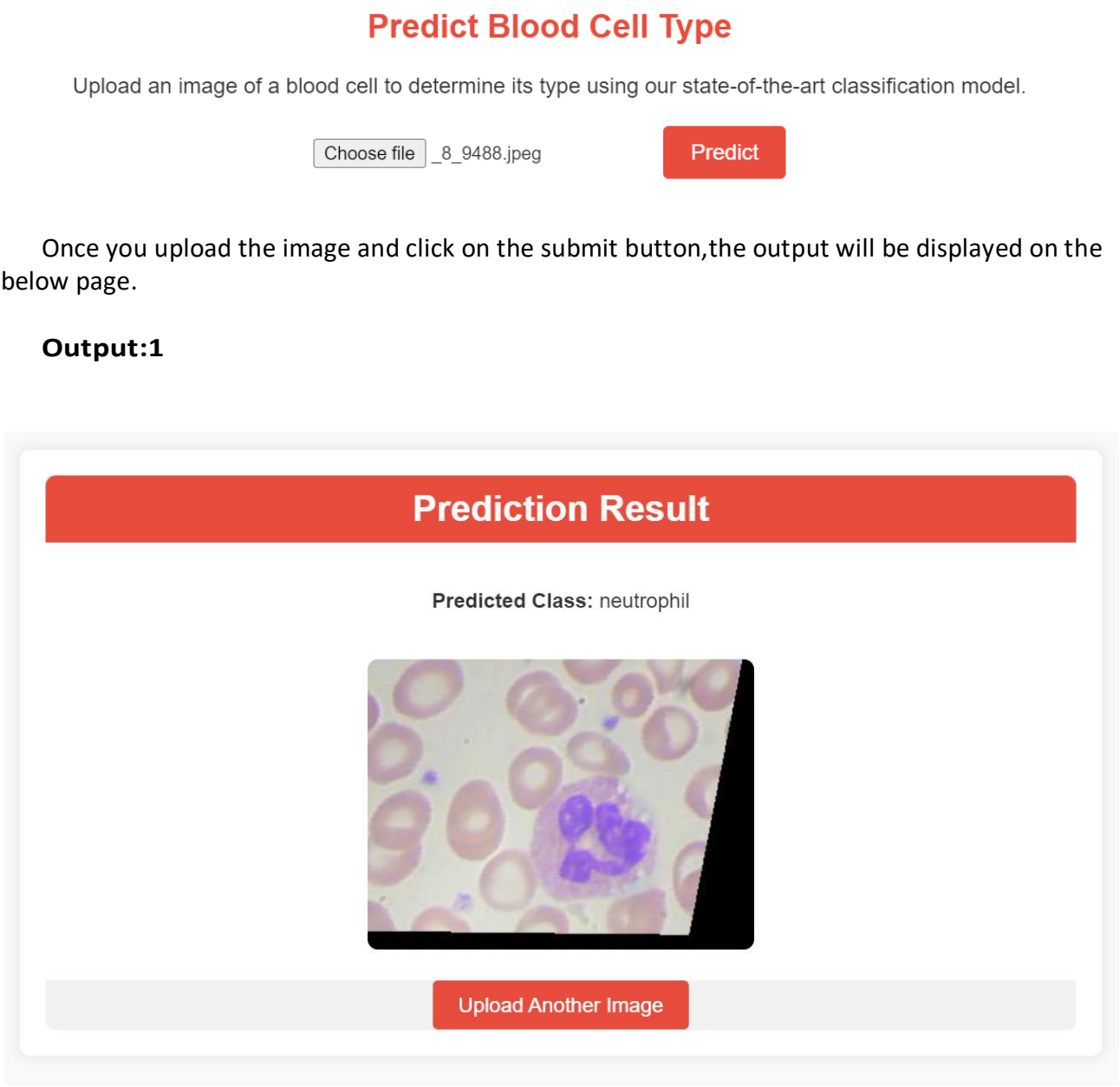
- Tested model predictions for all cell types (valid inputs)
- Tested invalid file uploads (e.g., text or unsupported formats)
- Verified UI response and layout in different screen sizes
- Tested prediction time and model inference latency
- Cross-checked output consistency for repeated inputs

Bug Fixes & Improvements

- Fixed error where large image uploads caused Streamlit to crash
- Improved cell label mapping for clearer output
- Resolved UI glitch where prediction text would not update
- Reduced false negatives by adding more training data per class

Final Output

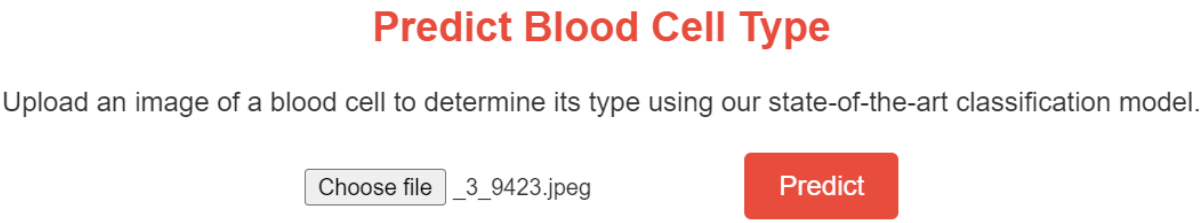
Input:1



Once you upload the image and click on the submit button,the output will be displayed on the below page.

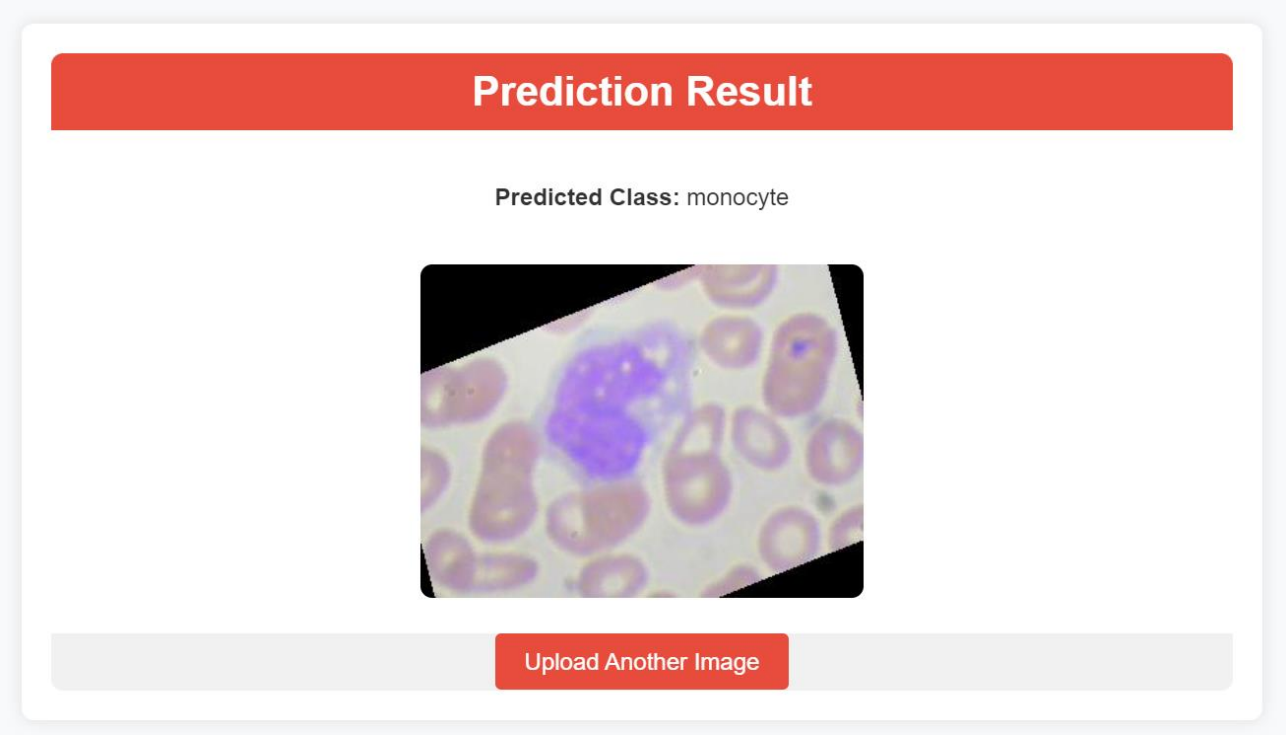
Output:1

Input:2



Onceyouuploadtheimageandclickonthesubmitbutton,theoutputwillbedis- played on the below page

Output:2



Deployment

HematoVision was deployed using **Streamlit** and tested on a local server as well as shared via public URL. The system runs smoothly on mid-range systems with CPU or GPU support, allowing real-time classification from the browser.

Appendix

ProjectResources

The following resources are available to better understand, reproduce, or extend the work done in the Pattern Sense project:

- **GitHubRepositoryLink:** LASASRAVANI/hematovision-advanced-blood-cell-classification-using-transfer-learning
- **DemoVideo:** https://drive.google.com/file/d/1Vbb7tr5Vdsa4AVug7_taOtlwDsBWwzC/view?usp=sharing
- **DatasetSamples:** https://www.kaggle.com/datasets/paultimothymooney/blood-cells/data