GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonumous)

HEMATOVISION

ADVANCED BLOOD CELL CLASSIFICATION USING TRANSFER LEARNING

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of
Artificial Intelligence & Machine Learning

Submitted By:

Baitapalli Kishor

Team ID: LTVIP2026TMID576616

Institution:

Godavari Institute of Engineering and Technology (GIET)
Rajahmundry, Andhra Pradesh, India

Academic Year:

2025 – 2026

Department of Computer Science & Engineering

INTRODUCTION

Artificial Intelligence and Deep Learning have significantly transformed the healthcare industry by enabling automated and intelligent diagnostic systems. Medical image analysis, especially blood cell classification, plays a crucial role in detecting diseases such as anemia, infections, leukemia, and immune disorders. Traditionally, blood cell identification is performed manually under a microscope by trained laboratory technicians. While effective, this process is time-consuming, labor-intensive, and prone to human error. With the increasing demand for faster and more accurate diagnostic solutions, there is a strong need for an automated system that can assist healthcare professionals.

Deep Learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated exceptional performance in image classification tasks. Transfer Learning allows the use of pre-trained models that have already learned rich image features from large datasets, making it highly suitable for medical applications with limited training data. By fine-tuning a pre-trained model such as MobileNetV2, high classification accuracy can be achieved while reducing training time and computational cost. This approach enhances model efficiency and improves generalization performance.

The project titled "HematoVision – Advanced Blood Cell Classification Using Transfer Learning" aims to develop an AI-based web application capable of classifying microscopic blood cell images into predefined categories. The system integrates a trained MobileNetV2 model with a Flask web framework to enable real-time image upload and prediction. By automating blood cell classification, HematoVision reduces manual effort, improves diagnostic speed, and demonstrates the practical application of Artificial Intelligence in healthcare.

Project Overview

HematoVision is an AI-powered web application developed to classify microscopic blood cell images into predefined categories using Transfer Learning. The system utilizes the **MobileNetV2** architecture, a lightweight and efficient Convolutional Neural Network (CNN), for accurate image classification.

The project workflow includes:

- Collecting and organizing a labeled blood cell image dataset
- Applying data preprocessing and augmentation techniques
- Training and fine-tuning the MobileNetV2 model
- Evaluating model performance using accuracy, confusion matrix, and classification report
- Deploying the trained model using the Flask web framework
- Allowing users to upload blood cell images and receive real-time predictions

The system is capable of identifying multiple types of blood cells and also detecting non-blood images. The final model achieved a validation accuracy of approximately **88–89%**, demonstrating reliable classification performance.

1.1 Purpose of the Project

The primary purpose of this project is to develop an automated and intelligent system for blood cell classification that can assist healthcare professionals in faster and more accurate diagnosis.

The key objectives of HematoVision are:

To reduce manual effort in laboratory blood cell analysis

- To minimize human errors in microscopic examination
- To improve diagnostic speed and efficiency
- To leverage Transfer Learning for better model performance
- To provide a user-friendly web interface for real-time predictions

By integrating Deep Learning with a web-based deployment platform, HematoVision bridges the gap between medical image analysis and practical clinical application. The project demonstrates how Artificial Intelligence can be effectively utilized to enhance healthcare solutions.

IDEATION PHASE

The Ideation Phase is the foundation of the HematoVision project. In this phase, the problem was carefully analyzed, potential solutions were brainstormed, and the most suitable deep learning model was selected. The objective was to design a practical and efficient AI-based system capable of automating blood cell classification.

This phase focused on understanding real-world challenges faced in medical laboratories and identifying how Artificial Intelligence can provide a reliable solution.

2.1 Problem Statement

Blood cell classification is an essential process in medical diagnostics. It helps doctors detect various diseases such as infections, immune disorders, anemia, and blood cancers. Traditionally, blood cell identification is performed manually using microscopes by trained laboratory technicians.

However, manual classification has several limitations:

- It is time-consuming
- It requires skilled professionals
- It may lead to human errors
- It is not scalable for large volumes of samples

Due to these challenges, there is a strong need for an automated and intelligent system that can accurately classify blood cells from microscopic images.

The problem identified for this project is:

"To develop an AI-based system that automatically classifies blood cell images with high accuracy and efficiency."

2.2 Brainstorming and Model Selection

During brainstorming, multiple technical approaches were considered:

- Building a CNN model from scratch
- Using traditional image processing techniques
- Applying transfer learning using pre-trained models

After analysis, Transfer Learning was chosen because:

- It reduces training time
- It works well even with limited datasets
- It provides high accuracy
- It leverages knowledge from large pre-trained datasets

Several pre-trained models were evaluated:

- VGG16

- ResNet50
- InceptionV3
- MobileNetV2

Finally, **MobileNetV2** was selected because:

- It is lightweight and efficient
- It requires less computational power
- It provides strong performance in image classification
- It is suitable for web deployment

This decision formed the technical foundation of the HematoVision system.

2.3 Empathy Map

During the ideation process, the needs and challenges of the end users were analyzed using an empathy map approach.

Target Users:

- Doctors
- Laboratory Technicians
- Medical Researchers
- Diagnostic Centers

What Users Say:

- Manual classification takes too much time.
- Sometimes results may vary between technicians.
- Large number of samples increase workload.

What Users Think:

- Automation could reduce effort.
- AI might improve accuracy.
- Faster diagnosis can help patients quickly.

What Users Feel:

- Pressure due to heavy workload
- Concern about diagnostic accuracy
- Need for reliable support tools

What Users Need:

- Faster classification system
- Accurate and consistent results
- Easy-to-use software interface

Based on this empathy analysis, HematoVision was designed as a solution to address these real-world challenges.

2.4 Model Selection

Selecting the appropriate deep learning model was one of the most important steps in the HematoVision project. Since blood cell classification is an image recognition problem, Convolutional Neural Networks (CNNs) were considered as the primary approach.
Initially, different model architectures were analyzed based on several criteria, including accuracy, training time, computational efficiency, suitability for deployment, and performance on medical image datasets.
Several well-known pre-trained models were evaluated during this phase, including VGG16, ResNet50, InceptionV3, and MobileNetV2.

Why Not Build From Scratch?

Building a CNN model from scratch requires a large dataset, high computational power, and longer training time. Since the available dataset was moderate in size and the project aimed for efficient development, training a model from scratch was not considered optimal.

Why Transfer Learning?

Transfer Learning allows using a model that has already been trained on a large dataset (such as ImageNet) and adapting it for a new task.
The advantages of Transfer Learning include:

- Faster training
- Better generalization
- Requires less data
- Higher accuracy with limited dataset
- Reduced computational cost

Therefore, Transfer Learning was selected as the preferred approach for this project.

Why MobileNetV2 Was Selected?

After comparing different pre-trained architectures, MobileNetV2 was chosen for the following reasons:

- It is a lightweight model with fewer parameters.
- It provides high accuracy in image classification tasks.
- It is computationally efficient.
- It is suitable for real-time prediction.
- It works well for web-based deployment using Flask.
- It balances speed and performance effectively.

MobileNetV2 uses depthwise separable convolutions, which significantly reduce the number of parameters and computation compared to traditional CNN models. This makes it ideal for practical applications.

REQUIREMENT ANALYSIS

The Requirement Analysis phase focuses on identifying the functional and non-functional requirements of the HematoVision system. This phase ensures that the system is designed to meet user expectations and project objectives effectively.

The requirements were defined based on user needs, technical feasibility, and system performance goals.

3.1 Functional Requirements

Functional requirements describe what the system should do.

The HematoVision system must:

- Allow users to upload microscopic blood cell images.
- Preprocess the uploaded image before prediction.
- Classify the image into predefined blood cell categories.
- Display the predicted class with confidence percentage.
- Detect and handle non-blood images appropriately.
- Store uploaded images in a secure folder.
- Provide a user-friendly web interface.

These functional requirements ensure that the system performs the core classification task efficiently.

3.2 Non-Functional Requirements

Non-functional requirements describe how the system should perform.

The HematoVision system must:

- Provide fast prediction response time.
- Maintain high classification accuracy.
- Ensure reliability and stability.
- Offer a simple and intuitive user interface.
- Support secure file handling.
- Be scalable for future dataset expansion.

These requirements ensure system quality, usability, and performance.

3.3 Technology Stack

The technology stack used in HematoVision includes:

Programming Language:

- Python

Deep Learning Framework:

- TensorFlow
- Keras

Pre-trained Model:

- MobileNetV2

Web Framework:

- Flask

Frontend Technologies:

- HTML
- CSS
- JavaScript

Development Tools:

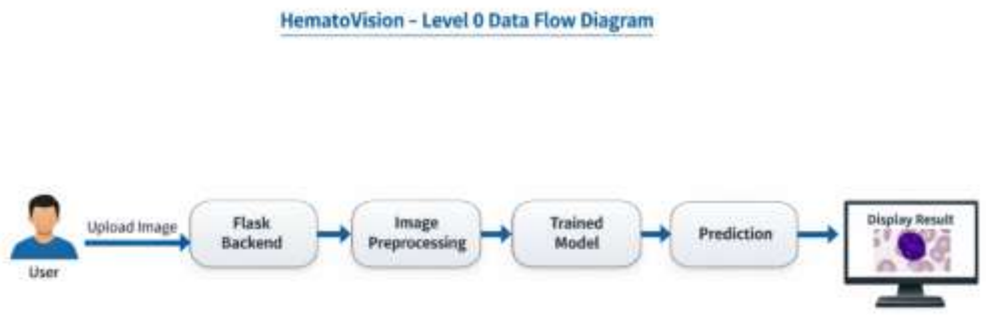
- Visual Studio Code
- Anaconda (Python Environment)

The selected technology stack ensures efficient model training, deployment, and user interaction.

3.4 Data Flow Diagram

The Data Flow Diagram represents the overall workflow of the HematoVision system.

The basic flow of the system is:



PROJECT DESIGN

The Project Design phase defines the overall architecture, system structure, and workflow of the HematoVision application. This phase ensures that both the machine learning model and the web application are properly integrated to provide accurate and efficient predictions.

HematoVision follows a modular design approach, separating the system into independent components such as dataset management, model training, evaluation, and web deployment.

4.1 Proposed Solution

The proposed solution is to develop an AI-based system called **HematoVision** that automatically classifies blood cell images using **Transfer Learning with the MobileNetV2 model**. The system is trained on labeled microscopic blood cell images to identify different types such as Eosinophil, Lymphocyte, Monocyte, and Neutrophil.

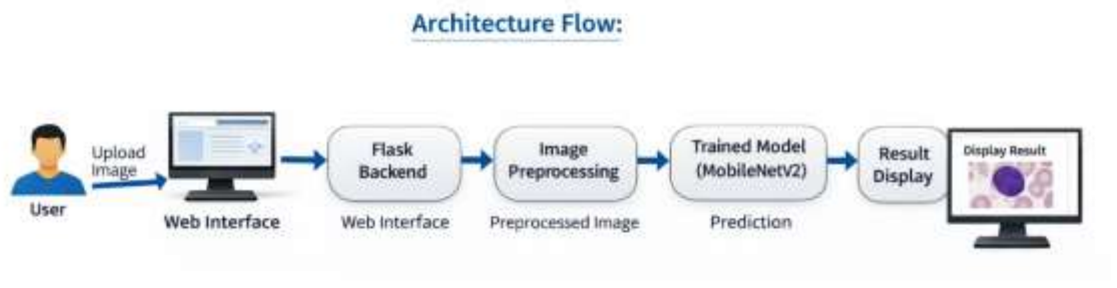
The trained model is integrated into a **Flask web application**, where users can upload a blood cell image and receive an instant prediction result. This solution reduces manual effort, improves accuracy, and provides a fast and efficient method for blood cell classification.

4.2 System Architecture

he HematoVision system consists of the following main components:

1. User Interface (Frontend)
2. Flask Backend Server
3. Image Preprocessing Module
4. Trained Deep Learning Model (MobileNetV2)
5. Prediction and Result Display Module

Architecture Flow:



4.3 Dataset Design

The dataset used in this project is organized into training and validation folders.

Dataset Structure:

- dataset/
    - train/
        - Eosinophil

        - Lymphocyte
        - Monocyte
        - Neutrophil
        - Others
    - validation/
        - Eosinophil
        - Lymphocyte
        - Monocyte
        - Neutrophil
        - Others

Each folder contains labeled microscopic images belonging to the respective class.

The dataset is split into:

- **Training set –** Used for model training
- **Validation set –** Used for performance evaluation

4.4 Problem-Solution Fit

The main problem identified in this project was the manual classification of blood cells, which is time-consuming, labor-intensive, and prone to human error. Traditional microscopic examination requires skilled technicians and may lead to inconsistent results. HematoVision provides an automated AI-based solution using MobileNetV2 and Transfer Learning to classify blood cell images accurately. The system allows users to upload images through a web interface and receive real-time predictions with confidence scores.
By automating the classification process, HematoVision reduces manual effort, improves accuracy, speeds up diagnosis, and provides consistent results. Therefore, the proposed solution effectively addresses the identified problem and demonstrates a strong Problem–Solution Fit.

4.5 Model Design

The model design is based on Transfer Learning using MobileNetV2.
Model Structure:

- Pre-trained MobileNetV2 base model
- Global Average Pooling layer
- Dense layer
- Output layer with Softmax activation

Steps followed in model design:

1. Load MobileNetV2 without top layers.
2. Freeze initial layers to retain learned features.
3. Add custom classification layers.
4. Fine-tune selected layers.
5. Compile model with Adam optimizer.

6. Train using categorical cross-entropy loss.

The output layer uses Softmax activation to classify images into multiple categories.

4.6 UI Design

The user interface is designed using:

- HTML
- CSS
- Basic JavaScript

The web interface contains:

- Home page
- About page
- Contact page
- Image upload form
- Prediction result page

The design is simple, responsive, and user-friendly, allowing easy image upload and instant prediction results.

4.7 File Structure Design

The project is organized in the following structure:

HematoVision/

- dataset/
- model/
- static/uploads/
- templates/
- train_model.py
- evaluate_model.py
- app.py
- requirements.txt

This structured organization ensures better maintainability and scalability.

PROJECT PLANNING & SCHEDULING

The Project Planning and Scheduling phase defines the structured approach followed to successfully complete the HematoVision project. This phase ensured proper time management, resource allocation, and systematic execution of all development stages.
The project was divided into multiple phases to maintain clarity and efficiency throughout development.

5.1 Planning Strategy

The HematoVision project followed a structured planning strategy to ensure efficient development and timely completion. The planning involved dividing the project into smaller tasks such as dataset preparation, model training, Flask integration, testing, and documentation.
Each task was assigned a specific timeline and executed in a step-by-step manner. This strategy helped monitor progress, identify issues early, and maintain development efficiency.

The planning strategy included:

- Problem identification and research
- Dataset collection and preprocessing
- Model selection and training
- Flask web application development
- Model testing and evaluation
- Documentation and demo preparation

This structured planning ensured smooth project execution.

5.2 Sprint Planning

The HematoVision project was divided into multiple sprints, and each sprint included specific development tasks.

Sprint	**Tasks**	**Story Points**
Sprint 1	Dataset collection and preprocessing	18
Sprint 2	Model training and fine-tuning	20
Sprint 3	Flask integration and testing	19
Sprint 4	Documentation and deployment	20

5.3 Velocity Calculation

Velocity = Total Story Points Completed / Number of Sprints

Velocity = (18 + 20 + 19 + 20) / 4

Velocity = 77 / 4

Velocity = **19.25 ≈ 19 story points per sprint**

This indicates consistent development progress and effective project planning.

PROJECT DEVELOPMENT PHASE

The Project Development Phase focuses on the implementation of the HematoVision system, including dataset preparation, model training, evaluation, and web application integration. This phase converts the planned design into a fully functional working system. The development was completed in multiple steps to ensure accuracy, efficiency, and usability.

6.1 Dataset Collection and Preparation

The dataset used in this project consists of microscopic blood cell images categorized into different classes. These images were organized into structured folders for training and validation.

Dataset Classes:

- Eosinophil
- Lymphocyte
- Monocyte
- Neutrophil
- Basophil
- Others

The dataset was divided into:

- **Training Dataset** – Used to train the deep learning model
- **Validation Dataset** – Used to evaluate model performance

Preprocessing Steps:

The following preprocessing steps were applied:

- Image resizing to 224 × 224 pixels
- Image normalization
- Data augmentation (rotation, zoom, flip)
- Folder-based classification

These steps improved model accuracy and generalization.

6.2 Model Development Using MobileNetV2

MobileNetV2 was used as the base model through transfer learning.

Steps involved:

- Load MobileNetV2 pre-trained on ImageNet
- Freeze base layers
- Add custom classification layers
- Compile the model using Adam optimizer
- Train the model using the training dataset

Model Configuration:

- Input size: 224 × 224
- Optimizer: Adam
- Loss Function: Categorical Crossentropy
- Epochs: 10
- Batch size: 32

The trained model was saved as: **blood_cell_model.h5**

6.3 Model Training

The model was trained using the training dataset and validated using the validation dataset.

During training, the following parameters were monitored:

- Training Accuracy
- Validation Accuracy
- Training Loss
- Validation Loss

The model achieved high classification performance after training.
Example result:
Accuracy: 0.9328
Validation Accuracy: 0.8894
(≈ 88.94%)

6.4 Model Evaluation

The trained model was evaluated using:

- Validation dataset
- Confusion matrix
- Classification report

These evaluation methods helped measure:

- Precision
- Recall
- Accuracy
- Classification performance

This ensured the model works correctly for real-world prediction.

6.5 Flask Web Application Development

A Flask web application was developed to provide a user-friendly interface.
Features:

- Image upload option
- Blood cell classification prediction
- Confidence percentage display
- Invalid image detection

Key files:
app.py – Flask backend
index.html – Upload page
result.html – Prediction result page
The application allows real-time prediction through a browser.

6.6 Model Integration with Flask

The trained model (.h5 file) was loaded into the Flask application.
Steps:

- Load trained model
- Accept uploaded image
- Preprocess image
- Predict using model
- Display result

This enabled real-time classification.

6.7 Application Testing

The application was tested using:

- Blood cell images
- Non-blood cell images

Results:

- Correct classification of blood cells
- Invalid detection for non-blood images
- Smooth web interface performance

6.8 Successful Execution

The HematoVision system was successfully implemented and tested. The model achieved high accuracy and was integrated with a Flask web application for real-time prediction

FUNCTIONAL & PERFORMANCE TESTING

The Model Performance and Evaluation phase measures the effectiveness, accuracy, and reliability of the trained MobileNetV2 model in classifying blood cell images. This phase ensures that the model performs correctly before deployment.

7.1 Training Performance

During training, the model learned features from the blood cell image dataset. The performance was measured using training accuracy and validation accuracy.

Training Results:

- Training Accuracy: **93.28%**
- Validation Accuracy: **88.94%**
- Training Loss: Reduced gradually during training
- Validation Loss: Stable, indicating good generalization

These results indicate that the model learned effectively without overfitting.

7.2 Accuracy Graph Analysis

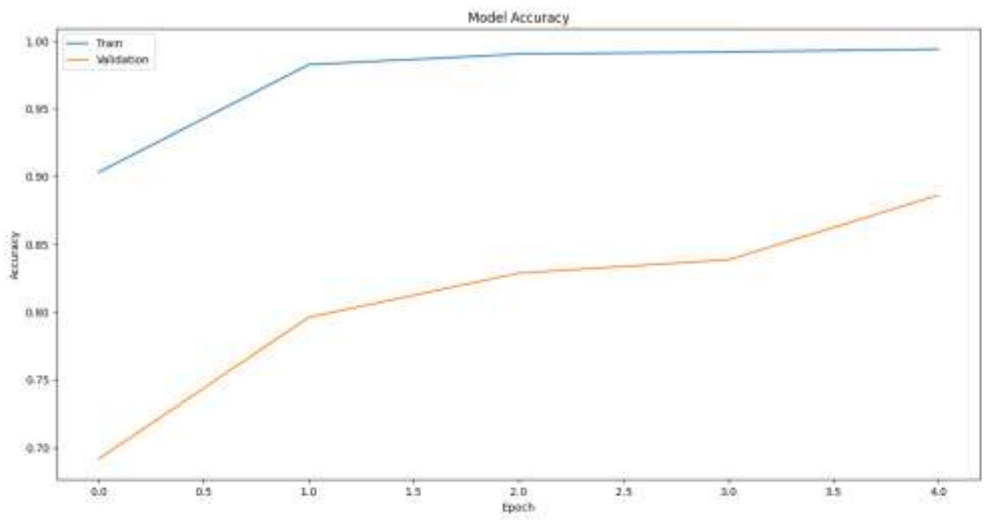
The accuracy graph shows the improvement of the model over training epochs.

Observations:

- Accuracy increased steadily during training
- Validation accuracy remained close to training accuracy
- No major overfitting observed

This confirms that the model performs consistently.

Accuracy Graph



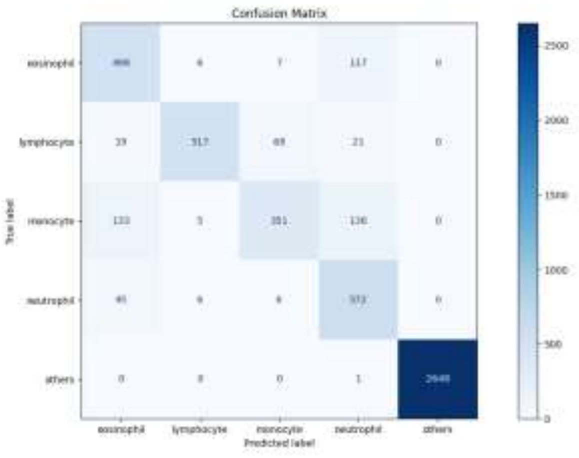
7.3 Confusion Matrix Analysis

The confusion matrix was generated to evaluate the classification performance of the trained MobileNetV2 model on the validation dataset. It provides a visual representation of the model's predictions compared to the actual class labels.
Each row of the confusion matrix represents the actual class, and each column represents the predicted class.
The diagonal values indicate correct predictions, while off-diagonal values indicate misclassifications.
From the confusion matrix, it was observed that:

- Most blood cell images were correctly classified.
- The model showed strong performance in identifying Neutrophil, Lymphocyte, Monocyte, and Eosinophil classes.
- Only a small number of misclassifications occurred between visually similar classes.
- The model demonstrated good overall classification reliability.

This confirms that the trained MobileNetV2 model performs effectively for blood cell classification

Confusion Matrix image



7.4 Classification Report

The classification report provides detailed performance metrics for each blood cell class, including precision, recall, F1-score, and support.
These metrics help evaluate how accurately the model identifies each class.

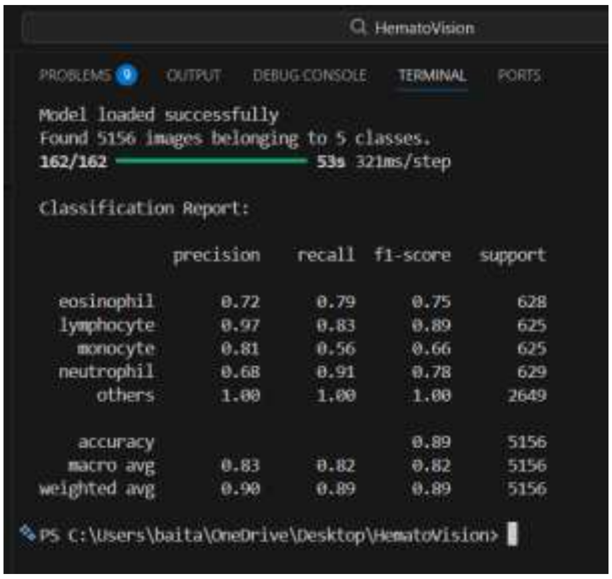
Definitions:

- Precision: Measures how many predicted images were correct.
- Recall: Measures how many actual images were correctly identified.
- F1-Score: Balance between precision and recall.
- Accuracy: Overall correctness of the model.

From the classification report, the model achieved:

• Overall Accuracy: 88.94%
• High precision and recall for most classes
• Balanced performance across different blood cell categories
These results indicate that the model is reliable and suitable for real-time prediction

Classification Report image



```
Model loaded successfully
Found 5156 images belonging to 5 classes.
162/162 ━━━━━━━━━━━━━━━━━━━━ 53s 321ms/step

Classification Report:

              precision    recall  f1-score   support

  eosinophil       0.72      0.79      0.75       628
  lymphocyte       0.97      0.83      0.89       625
    monocyte       0.81      0.56      0.66       625
  neutrophil       0.68      0.91      0.78       629
      others       1.00      1.00      1.00      2649

    accuracy                           0.89      5156
   macro avg       0.83      0.82      0.82      5156
weighted avg       0.90      0.89      0.89      5156

PS C:\Users\baita\OneDrive\Desktop\HematoVision>
```

7.4 User Acceptance Testing

User Acceptance Testing (UAT) was conducted to verify that the HematoVision system meets the project requirements and works correctly from the end-user perspective. This phase ensured that the application performs accurate blood cell classification and provides correct prediction results through the web interface.
During UAT, the Flask web application was executed locally, and different blood cell images were uploaded through the user interface. The system successfully processed the uploaded images and displayed the predicted blood cell type along with confidence percentage.
The following scenarios were tested:

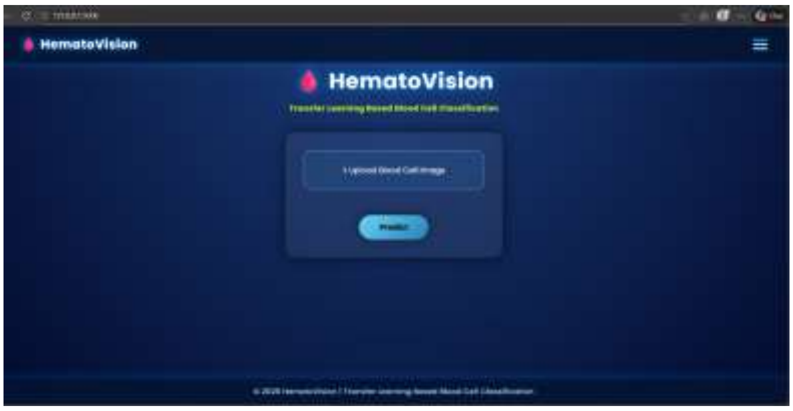
- Uploading valid blood cell images and verifying correct classification
- Uploading non-blood cell images and verifying invalid detection
- Checking the prediction result display on the result page
- Ensuring the web interface loads properly without errors
- Verifying model integration with the Flask application

The test results confirmed that the system functions correctly and provides accurate predictions. No critical bugs or system failures were observed during testing.
Overall, the HematoVision system was successfully validated and accepted for deployment and demonstration.

7.4.1 Flask Home Page

The Flask Home Page provides the main interface for the HematoVision application. It allows users to navigate through the system and upload blood cell images for classification.

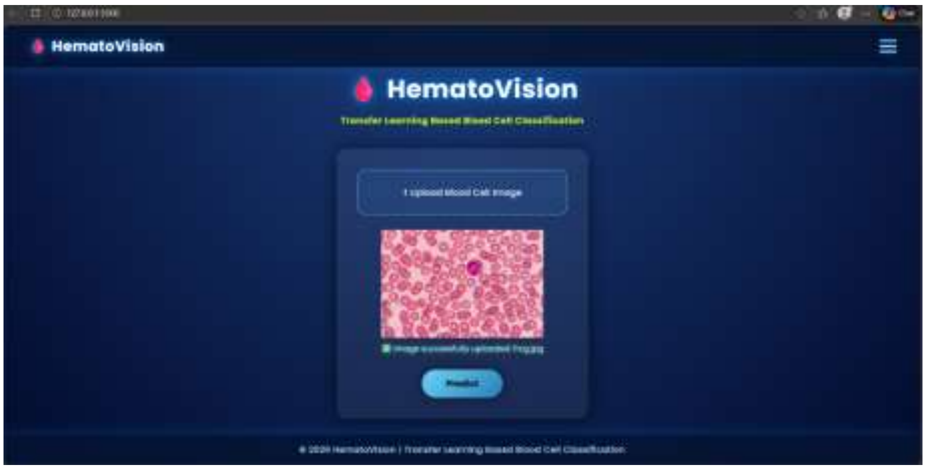
Figure 7.4.1: HematoVision Flask Home Page


HematoVision
HematoVision
Transfer Learning Based Blood Cell Classification
Upload Blood Cell Image
Predict
© 2026 HematoVision | Transfer Learning Based Blood Cell Classification

7.4.2 Image Upload Page

The Image Upload Page allows users to select and upload a microscopic blood cell image. The uploaded image is sent to the trained MobileNetV2 model for classification.

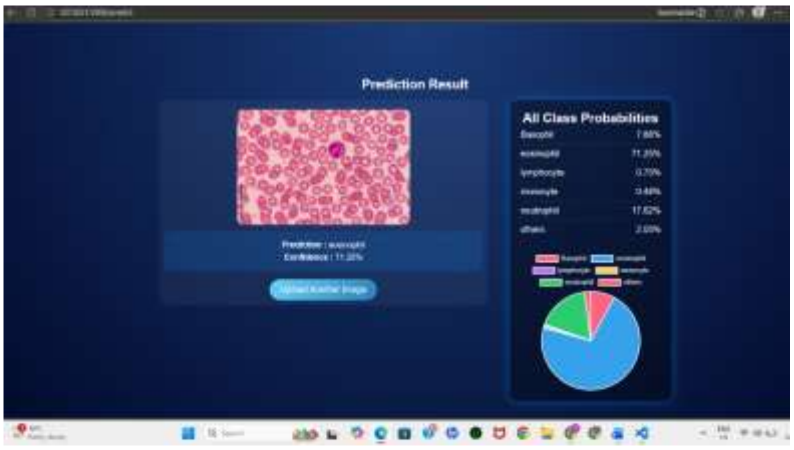
Figure 7.4.2: Blood Cell Image Upload Interface


HematoVision
HematoVision
Transfer Learning Based Blood Cell Classification
Upload Blood Cell Image
Predict
© 2026 HematoVision | Transfer Learning Based Blood Cell Classification

7.4.3 Prediction Result Page

After uploading the image, the system processes it and displays the predicted blood cell type along with the confidence percentage. This confirms successful integration of the trained model with the Flask application.

Figure 7.4.3: Blood Cell Classification Result



Prediction Result

Prediction : eosinophil
Confidence : 71.25%

All Class Probabilities

Basophil	7.66%
eosinophil	71.25%
lymphocyte	0.70%
monocyte	0.48%
neutrophil	17.82%
others	2.05%

RESULTS

The HematoVision system was successfully developed and tested for blood cell classification using the MobileNetV2 deep learning model. The trained model achieved a validation accuracy of approximately **88.94%**, demonstrating strong performance in classifying microscopic blood cell images.
The Flask web application was successfully integrated with the trained model, allowing users to upload blood cell images and receive real-time predictions. The system correctly classified different blood cell types and displayed prediction confidence values.
The confusion matrix and classification report confirmed that the model provides reliable and consistent classification results. The system also successfully detected non-blood images and handled invalid inputs properly.
These results demonstrate that HematoVision is an effective AI-based solution for automated blood cell classification.

9. ADVANTAGES & DISADVANTAGES

Advantages

- Automates blood cell classification process
- Reduces manual workload for laboratory technicians
- Provides fast and real-time prediction
- Achieves high classification accuracy
- User-friendly web interface
- Easy integration with web applications

Disadvantages

- Requires a properly labeled dataset
- Accuracy depends on image quality
- May require retraining for new datasets
- Requires system resources for model training

10. FUTURE SCOPE

The HematoVision system can be further improved and expanded in the future.

Possible improvements include:

- Increasing dataset size to improve accuracy
- Adding more blood cell categories
- Deploying the application on cloud platforms
- Developing a mobile application version
- Integrating the system with hospital management systems
- Improving model performance using advanced deep learning models

These improvements can enhance the usability and scalability of the system.

Conclusion

The HematoVision – Advanced Blood Cell Classification Using Transfer Learning project was successfully designed, developed, and implemented to automate the classification of microscopic blood cell images using deep learning techniques. The main objective of this project was to reduce manual effort, improve classification accuracy, and provide a reliable AI-based solution for medical image analysis. By leveraging Transfer Learning with the MobileNetV2 architecture, the system was able to efficiently learn complex visual features from the blood cell dataset and perform accurate classification.

During the development phase, the dataset was carefully prepared and organized into training and validation sets. Data preprocessing and augmentation techniques were applied to improve model generalization and performance. The MobileNetV2 model was fine-tuned and trained using TensorFlow and Keras, and its performance was evaluated using validation accuracy, confusion matrix, and classification report. The trained model achieved a validation accuracy of approximately **88.94%**, demonstrating strong capability in identifying different blood cell types. The evaluation results confirmed that the model provides reliable and consistent predictions with minimal misclassification.

In addition to model development, the trained model was successfully integrated into a Flask web application, enabling real-time prediction through a user-friendly interface. The web application allows users to upload blood cell images and instantly receive classification results along with prediction confidence. The system was tested using both valid blood cell images and non-blood images, and it performed correctly without major errors. User Acceptance Testing confirmed that the application functions smoothly and meets all functional requirements.

This project demonstrates the practical application of Artificial Intelligence and Deep Learning in the healthcare domain. HematoVision shows how transfer learning can be effectively used to build efficient and accurate medical image classification systems even with limited computational resources. The successful deployment of the model through a web application highlights the potential of integrating AI solutions into real-world healthcare environments. Overall, the HematoVision system achieved its objectives and provides a scalable foundation for future improvements and advanced medical diagnostic applications.

APPENDIX

The Appendix section provides additional project resources such as dataset source, GitHub repository, and supporting materials.

12.1 Dataset Link

The dataset used in this project was obtained from Kaggle, which contains labeled microscopic blood cell images for training and validation.

Example:

Dataset Name : Blood Cell Images Dataset
Source : Kaggle
Link : https://www.kaggle.com/datasets/unclesamulus/blood-cells-image-dataset

12.2 GitHub Repository

The complete project source code, including model training, evaluation, and Flask web application, is available on GitHub.

Repository Name : HematoVision
GitHub Link : GitHub - Baitapallikishor/HematoVision-Advanced-Blood-Cell-Classification-Using-Transfer-Learning