HematoVision: Advanced Blood Cell Classification Using Transfer Learning

1. Introduction

HematoVision is an intelligent diagnostic system designed to classify white blood cells from microscopic images using state-of-the-art deep learning techniques. The goal of this project is to support hematologists, medical researchers, and laboratory personnel by automating the classification of four primary types of blood cells: Eosinophils, Lymphocytes, Monocytes, and Neutrophils.

---

2. Objective

The objective of HematoVision is to:

- Build a robust and efficient machine learning model using Transfer Learning.
- Provide a web-based interface for uploading and predicting blood cell types.
- Achieve high classification accuracy with low computational cost.
- Make the solution accessible and easy to use for healthcare professionals.

---

3. Technology Stack

Layer	Technology
Programming	Python 3.11
Deep Learning	TensorFlow, Keras
Model Architecture	MobileNetV2 (Transfer Learning)
Web Framework	Flask
Frontend	HTML, Bootstrap 5
Deployment	Localhost via Flask server

---

4. Dataset Summary

- **Source**: Publicly available blood cell dataset.

- **Classes**: Eosinophil, Lymphocyte, Monocyte, Neutrophil
- **Total Images**: ~10,000
- **Train/Validation Split**: 80% training, 20% validation

---

5. Model Architecture

The model is based on MobileNetV2, a pre-trained convolutional neural network optimized for mobile and embedded vision applications. The architecture includes:

- Pre-trained MobileNetV2 base (with include_top=False)
- Global Average Pooling
- Fully connected Dense layer (ReLU activation)
- Output Dense layer with 4 units and Softmax activation

Training Configuration:

- **Optimizer**: Adam
- **Loss Function**: Categorical Crossentropy
- **Metrics**: Accuracy
- **Epochs**: 5

---

6. Application Workflow

1. User accesses the application via a browser.
2. An image of a blood cell is uploaded through the interface.
3. Image is preprocessed and resized to 224x224 pixels.
4. The image is fed into the MobileNetV2 model.
5. Predicted class and confidence score are displayed alongside the uploaded image.

---

7. Performance Metrics

Metric	Value
Final Train Accuracy	~75.2%

Metric	Value
Final Val Accuracy	~75.7%
Loss (Validation)	~0.63
Inference Time	~2 seconds/image

---

8. Project Structure

HematoVision/

├— app.py # Flask backend logic

├— train_model.py # Model training script

├— Red Blood Cell.h5 # Saved trained model

├— images/ # Uploaded images

├— templates/

| ├— index.html # Upload form

| ├— result.html # Result display

├— requirements.txt # Python dependencies

---

9. Installation & Execution

Prerequisites:

- Python 3.10+
- pip package manager

Steps to Run:

pip install -r requirements.txt

python app.py

Visit http://127.0.0.1:5000 in your browser.

---

10. Limitations & Future Scope

- **Current Limitation**: Supports only 4 WBC types.

- **Future Enhancements**:
    - Add more cell types (RBC, Platelets)
    - Cloud-based deployment (AWS, Render, Heroku)
    - Mobile app integration
    - Report generation (PDF)

---

11. Use Cases

- Preliminary blood analysis for pathologists
- Training tool for medical students
- Research-grade classification tool

---

12. Author & Acknowledgements

- **Developed by**: Bandi Pavan Chaitanya
- **Acknowledgements**: TensorFlow community, public WBC dataset providers, and open-source contributors

---

13. License

This project is released under the MIT License.

---

14. Contact

For inquiries or collaborations:
Email:[228x1a4251@khitguntur.ac.in]
GitHub:[https://github.com/Pavan-pavi/HematoVision-Advanced-Blood-Cell-Classification-Using-Transfer-Learning]