CleanTech: Transforming Waste Management with Transfer Learning

1. Introduction

1.1 Project Overview

HematoVision is a deep learning-based project designed to classify blood cells using transfer learning. It aims to assist healthcare professionals by automating blood smear image analysis, enhancing diagnostic accuracy and efficiency. The system utilizes a dataset of 12,000 annotated blood cell images across various classes like eosinophils, lymphocytes, monocytes, and neutrophils.

1.2 Objectives

- Utilize transfer learning to improve classification performance.
- Reduce training time while maintaining high accuracy.
- Automate blood cell image classification for clinical and remote applications.
- Provide a user-friendly interface for predictions and diagnostics.

---

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

Manual classification of blood cells is time-consuming and error-prone. The objective is to develop an accurate and automated system using pre-trained CNN models to classify different types of blood cells effectively.

2.2 Project Proposal (Proposed Solution)

This project proposes using transfer learning with models like MobileNet, VGG16, or ResNet50 to classify blood cells. The approach reduces computational cost and leverages pretrained knowledge for high classification accuracy.

2.3 Initial Project Planning

- Defined workflow (data collection, preprocessing, modeling).
- Chose appropriate pre-trained CNN architectures.
- Planned for web integration using Flask for deployment.

---

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Sources

- Dataset: Blood Cell Classification (from Kaggle or equivalent)
- Categories: Eosinophil, Lymphocyte, Monocyte, Neutrophil
- Total Samples: ~12,000 images

3.2 Data Quality Report

- Data Shape: 12,000 images across 4 categories
- Missing/Corrupted Images: Removed
- Image Size: Standardized to 224x224 pixels

3.3 Data Preprocessing

- Resized images to 224x224
- One-hot encoded class labels
- Train-Test Split (80%-20%)
- Applied ImageDataGenerator for augmentation

---

4. Model Development Phase

4.1 Feature Selection Report

- Features: Pixel intensity, patterns from CNN layers
- Target: Blood cell type

4.2 Model Selection Report

Models Used:

- VGG16 (transfer learning)
- MobileNetV2
- ResNet50

Metrics:

- Accuracy
- Confusion Matrix
- Classification Report (Precision, Recall, F1-Score)

4.3 Initial Training & Evaluation

- Base model: VGG16 (pre-trained on ImageNet)
- Fine-tuned last few layers
- Achieved ~91% accuracy on test data
- Best performance observed with MobileNetV2 + Data Augmentation

---

5. Model Optimization and Tuning Phase

5.1 Hyperparameter Tuning

- Learning Rate: Adjusted using ReduceLROnPlateau
- Optimizer: Adam, SGD
- Batch Size: Tested with 16, 32
- Epochs: 20–30 based on early stopping

5.2 Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score
VGG16	90.3%	91.2%	90.0%	90.5%
MobileNetV2	92.6%	93.1%	92.0%	92.5%
ResNet50	91.7%	92.4%	91.2%	91.8%

5.3 Final Model Selection

Selected Model: **MobileNetV2** due to:

- High accuracy and generalization
- Lower memory footprint
- Faster inference time

---

6. Results

6.1 Output Screenshots

- Web interface for image upload
- Model prediction displayed with class name
- Output graphs (accuracy/loss plots)

---

7. Advantages & Disadvantages

Advantages:

- Accurate and fast classification
- Transfer learning reduces training time
- Scalable to other medical image classification tasks

Disadvantages:

- Needs quality image inputs
- Model may require retraining for new blood cell types

---

8. Conclusion

HematoVision successfully demonstrates the potential of transfer learning in medical diagnostics. With 92%+ accuracy, the system can effectively classify blood cell types and assist medical professionals in diagnostics. It is a step toward automating hematology workflows.

---

9. Future Scope

- Expand dataset with more classes (e.g., abnormal cells)
- Integrate with hospital data systems
- Convert to a mobile or desktop application
- Explore attention-based deep learning models

---

10. Appendix

10.1 Source Code

- app.py (Flask backend)
- model_train.py (Training pipeline)
- templates/ (HTML templates)
- static/uploads/ (Image input folder)

10.2 GitHub & Project Demo

GitHub: https://github.com/renuka-matta/Hematovision.git