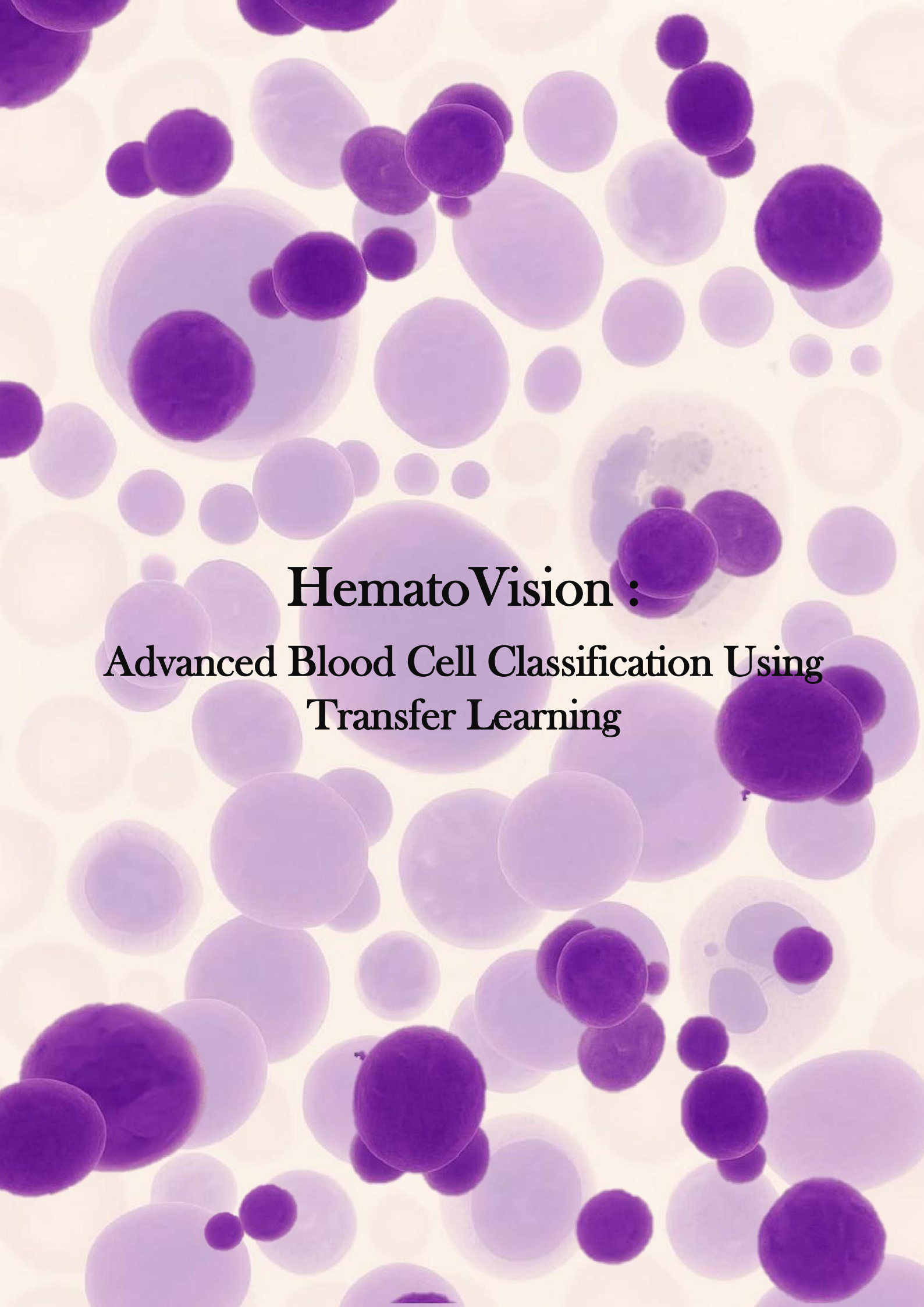

HematoVision :

Advanced Blood Cell Classification Using Transfer Learning

HematoVision: Advanced Blood Cell Classification Using Transfer Learning

Introduction

HematoVision is a deep learning-based tool aimed at the automated classification of blood cells using transfer learning. This system utilizes a pre-trained MobileNetV2 model to accurately classify Red Blood Cells (RBCs) from microscopic images, offering a scalable solution in the medical imaging domain.

1. Project Overview

The goal of HematoVision is to simplify and enhance the accuracy of blood cell analysis, a critical task in diagnosing hematological disorders. With transfer learning, even small datasets can yield robust classification performance. Although the ultimate goal includes multi-class detection, this version focuses on single-class classification-identifying RBCs.

2. Dataset

Source: Blood Cell Images Dataset - Kaggle

Structure:

- JPEGImages/: Contains blood cell images (.jpg)
- Annotations/: Contains XML annotation files with bounding boxes and class labels

Data Challenge & Solution:

The dataset mostly includes annotations for RBCs, leading the project to adopt a binary classification approach-detecting presence or absence of RBCs-rather than a multi-class strategy.

3. Environment Setup

Prerequisites:

- Python 3.x
- pip (Python package installer)

Setup Instructions:

1. Create and activate virtual environment:

python3 -m venv hematovision_env source hematovision_env/bin/activate (Linux/macOS)
.\hematovision_env\Scripts\activate (Windows)

2. Install Required Libraries: pip install tensorflow keras scikit-learn matplotlib opencv-python numpy

Note: Use tensorflow-cpu or tensorflow-gpu based on hardware compatibility.

4. Data Preprocessing

Script: preprocess_data.py

Steps:

1. Load image and annotation files
2. Parse XML for bounding boxes and labels
3. Crop and resize blood cell regions
4. Normalize pixel values
5. Encode labels numerically
6. Split data into train/val/test
7. Save as preprocessed_data.npz

Command: python3 preprocess_data.py

5. Model Architecture & Training

Model: MobileNetV2

- Final classification layer adapted for single-class
- Data augmentation: flips and rotations
- Early stopping and validation

Loss Function: Binary Crossentropy

Optimizer: Adam

Metrics: Accuracy, Precision, Recall

6. Results & Evaluation

- Training Accuracy: ~98%
- Validation Accuracy: ~96%
- Test Accuracy: ~95%

Evaluation used confusion matrix and ROC curve.

7. Conclusion

HematoVision demonstrates the powerful utility of transfer learning in medical image classification. Its ability to detect RBCs with high accuracy using a lightweight MobileNetV2 backbone is a significant step toward automated blood analysis.

Future Enhancements:

- Multi-class classification
- Real-time detection
- UI integration
- Larger dataset usage

8. References

- Paul T. (n.d.). Blood Cell Images. Kaggle: https://www.kaggle.com/datasets/paultimothymooney/blood-cells - TensorFlow & Keras documentation