HematoVision: Blood Cell Classification Using Machine Learning

Author: Nithin Reddipalli

Abstract

HematoVision is a machine learning-based diagnostic model designed to classify blood cell types using microscopic image data. The system utilizes Convolutional Neural Networks (CNNs) to process and learn features from hematological images. Although a simulated dataset was used for demonstration, the workflow supports scaling to real-world datasets like ALL-IDB or BCCD for clinical deployment.

1. Dataset Preparation

1.1 Data Source (Simulated)

Due to constraints in accessing actual medical data, a dummy dataset was programmatically generated:

- Image Size: 128×128 pixels (RGB)
- Classes: lymphocyte, neutrophil
- Samples: 100 total (50 per class)

1.2 Preprocessing Steps

- Normalization: Rescaled pixel values to range [0, 1]
- Resizing: All images resized to uniform shape (128×128)
- Label Encoding: String labels mapped to integers
- Data Augmentation: Random horizontal flips, Random cropping and resizing

2. Model Architecture

2.1 CNN Model Design

A Sequential Convolutional Neural Network was built from scratch using TensorFlow/Keras.

Layers: Conv2D, MaxPooling2D, Flatten, Dense, Dropout

2.2 Compilation

- Loss Function: Sparse Categorical Crossentropy

- Optimizer: Adam

- Metrics: Accuracy

3. Model Training

3.1 Data Split

- Training Set: 80 images

- Validation Set: 20 images

3.2 Hyperparameters

- Epochs: 10

- Batch Size: 32

3.3 Training Output (Key Epochs)

Epoch 1: val_accuracy = 0.50 | val_loss = 0.8125

Epoch 5: val_accuracy = 0.50 | val_loss = 0.6907

Epoch 10: val_accuracy = 0.55 | val_loss = 0.7289

4. Evaluation

4.1 Validation Performance

- Final Accuracy: 55%

- Final Loss: 0.7289

4.2 Observations

- The model shows limited accuracy due to small dataset size and randomized image content.

- Performance is expected to improve significantly with real image data and transfer learning.

5. Web Interface

Two HTML pages were created using Bootstrap 5:

1. Upload Page: Allows users to upload blood cell images.

2. Result Page: Displays predicted class (lymphocyte or neutrophil).

This interface will connect to a Flask backend (app.py) for full functionality.

6. Key Learnings

- Image classification requires consistent preprocessing and balanced datasets.

- CNNs are effective but limited by data availability.

- Transfer learning is essential for high-accuracy classification in medical imaging.

- HTML/Flask integration enables easy deployment of ML models for end-users.

7. Future Scope

- Integrate real blood cell datasets like ALL-IDB or BCCD.

- Upgrade to transfer learning (e.g., using MobileNetV2 or ResNet50).

- Add multi-class classification (monocytes, eosinophils, etc.).

- Deploy the Flask app to cloud platforms like Heroku, Render, or AWS.

- Add confusion matrix, precision, recall, and F1-score for robust evaluation.

Appendix

- dummy_hematology_data/: Directory with simulated images

- model.py: CNN training script

- templates/index.html: Upload interface

- templates/result.html: Classification result display

- blood_cell.h5: Saved model

- requirements.txt: Python dependencies