Malaria Cell Image Classification Using Feedforward Neural Networks

CS 747 - Deep Learning

Assignment 1: Building Feedforward Classifiers for Images

Fall 2025

Team: Rithvik Pranao Nagaraj (G01501815), Ashwin Ravichandran (G01525331)

1 Introduction

Malaria is one of the deadliest infectious diseases worldwide, responsible for millions of infections and hundreds of thousands of deaths annually, particularly in tropical and subtropical regions. Traditional diagnosis relies on manual microscopic examination of blood smears, which is time-consuming and requires skilled technicians.

To address this challenge, deep learning techniques can automate the classification of infected and uninfected cells, significantly improving diagnostic speed and accuracy. This project focuses on developing, training, validating, and testing a feedforward neural network (FFN) to classify malaria cell images into two categories: *Parasitized* and *Uninfected*. We also investigate the effects of different regularization techniques (L2, Dropout, Early Stopping) and compare our deep learning model with a traditional Support Vector Machine (SVM) baseline.

2 Methods

2.1 Dataset Preparation

The malaria dataset contains two classes: *Parasitized* and *Uninfected* cell images. To prepare the data for model training:

- Images were resized to 64×64 pixels using a preprocessing script.
- All images were converted to RGB format and normalized to the $[0, 1]$ range.
- The dataset was split into training (70%), validation (15%), and testing (15%).
- Light augmentation (rotation, flipping, brightness) was applied.
- Corrupt or unreadable files were skipped automatically.

2.2 Model Architecture

We designed a fully connected feedforward neural network (FFN) for binary classification. Each image was flattened into a 1D vector and passed through dense layers:

- Input: Flattened vector of size $64 \times 64 \times 3 = 12,288$
- Hidden Layer 1: 256 neurons (ReLU)
- Hidden Layer 2: 128 neurons (ReLU)
- Output Layer: 2 neurons (Softmax)

Weights were initialized using He initialization. We experimented with L2 regularization, dropout ($p = 0.2$–0.5), and early stopping.

2.3 Training Strategy

- **Loss Function:** Cross-entropy
- **Optimizer:** Adam (learning rate 1×10^{-4}–1×10^{-3})
- **Batch Sizes:** 512
- **Epochs:** Up to 100 with early stopping (patience 5–10)
- **Metrics:** Accuracy, Precision, Recall, F1-score
- **Checkpoints:** Best validation model saved for testing

3 Implementation

The project was implemented in **PyTorch** for the FFN and **scikit-learn** for the SVM. Image preprocessing used the Pillow library, while training and evaluation ran on an NVIDIA RTX 3060 GPU. The pipeline was modular, allowing experiments with different regularization settings (L2, dropout, early stopping) without altering the architecture. For the SVM, PCA feature reduction and grid search hyperparameter tuning were carried out with multi-core CPU support.

4 Results

We experimented with the fixed FFN architecture and regularization techniques. Table 1 shows the results:

Table 1: Model Performance Comparison

Model	**Validation Acc.**	**Test Acc.**	**Configuration**
Baseline	0.7034	0.7102	hidden=(256,128), lr=0.0003
L2 Regularization	0.7036	0.7154	hidden=(256,128), wd=0.0005
Dropout	0.7005	0.7084	hidden=(256,128), p=0.2
Early Stopping	0.6954	0.6992	hidden=(256,128), patience=5

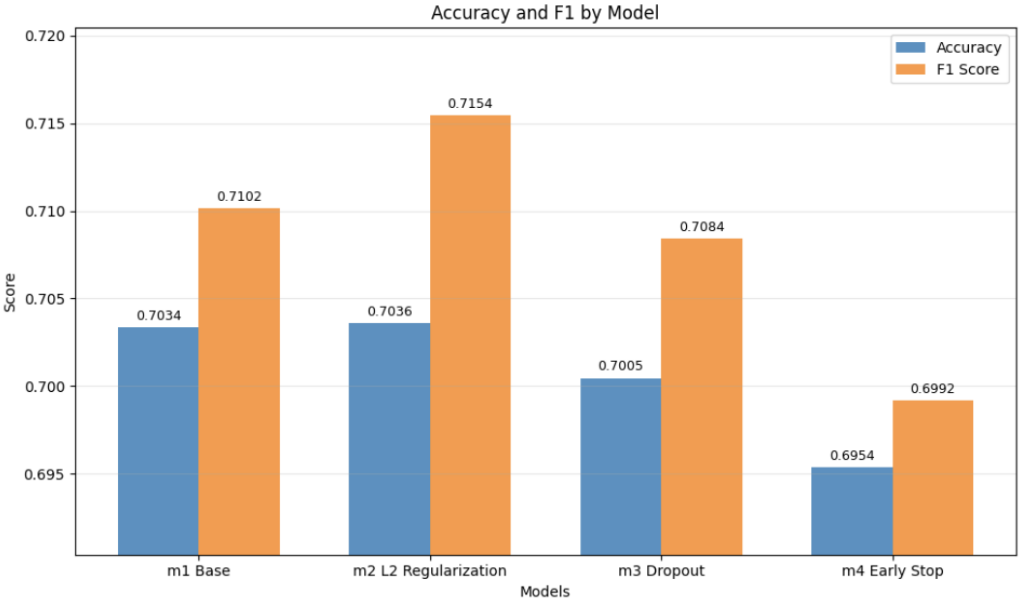

Figure 1: Accuracy and F1 score across FFN models with different regularization techniques.

Extra Credit: Support Vector Machine (SVM)

Per the extra-credit requirement, we trained, validated, and tested an SVM on the same resized images used for the FFN models.

Pipeline summary

- Inputs: 64×64 RGB images flattened (12,288D), then standardized.
- Feature engineering: PCA → 256 components to reduce dimensionality.
- Classifier: RBF-kernel SVM with a small grid over C and γ.
- Model selection: tuned on the validation split; final metrics on the held-out test set.

Results

- **Best DL (L2-FFN)** — Test Accuracy: **0.7154**
- **Best SVM (PCA256 + RBF)** — Test Accuracy: **0.7654**
- **Difference: +0.0500** absolute (+5.0 pp)

Table 2: SVM Confusion Matrix (Test). Rows: true class; columns: predicted class.

	Pred: Parasitized	**Pred: Uninfected**
True: Parasitized	1565	502
True: Uninfected	468	1599

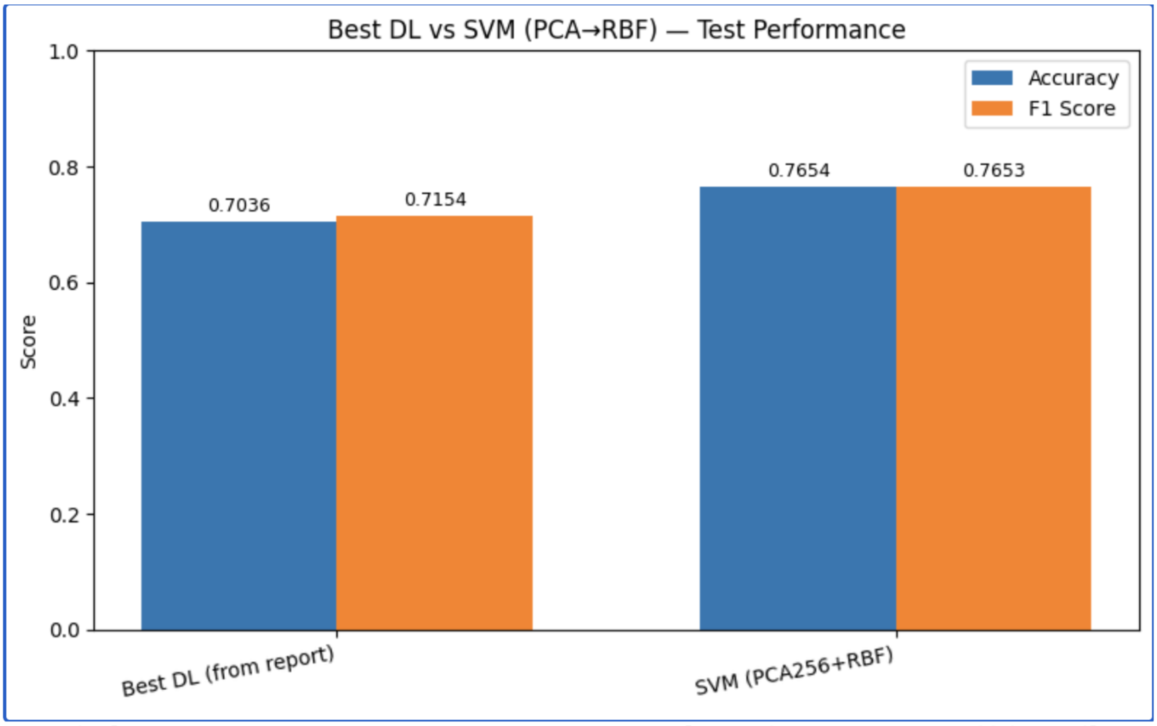

Figure 2: Comparison of test performance between the best DL model and the SVM (PCA256 + RBF).

Takeaway The SVM with PCA and an RBF kernel outperformed the best FFN by **5.0 percentage points** on the test set. This suggests that for small, fixed-size images and limited compute, PCA+SVM can be competitive or superior to simple FFNs. However, CNNs or transfer learning would likely surpass both.

5 Discussion

Our experiments demonstrate that even a relatively simple feedforward neural network can achieve reasonable performance on the malaria classification task. Data augmentation improved robustness, and regularization techniques like dropout and L2 weight decay helped mitigate overfitting but did not dramatically outperform the baseline. Early stopping avoided unnecessary training epochs but sometimes halted learning prematurely.

The architecture and preprocessing pipeline were effective, but improvements are possible. Techniques such as convolutional neural networks (CNNs), transfer learning, or transformer-based approaches could exceed 90% accuracy. Integrating interpretability tools such as Grad-CAM would also help visualize predictions, improving trust and adoption in medical contexts. The stronger SVM result shows that classical pipelines can remain competitive under certain constraints, though deep architectures are more scalable.

6 Conclusion

This study demonstrates that a simple FFN with two hidden layers (256, 128) can classify malaria cell images using a flattened representation, reaching $\sim 71.5\%$ test accuracy. Light augmentation provided implicit regularization, while L2, dropout, and early stopping gave marginal improvements.

The SVM baseline with PCA achieved $\sim 76.5\%$ accuracy, outperforming the FFN by 5 percentage points. This highlights that for fixed-size images and limited compute, SVMs remain viable baselines. Future work should explore CNNs, transfer learning, richer augmentations, and interpretability tools for improved accuracy and clinical reliability.