EE655: Computer Vision and Deep Learning

HOMEWORK - I

Name: Vaibhav Itauriya

Roll Number: 231115

Branch: ME

Date of submission: 25/12/2024

GitHub Repository of this Course:

Contents

1 Introduction

In this assignment, I worked on the problem of handwritten digit recognition using the MNIST dataset. I utilized the Histogram of Oriented Gradients (HOG) feature extraction technique and experimented with various machine learning models. Through systematic analysis and tuning, I aimed to maximize accuracy while gaining insights into model performance.

2 Dataset Overview

The MNIST dataset contains grayscale images of handwritten digits ranging from 0 to 9, each labeled accordingly. The dataset is a widely used benchmark in the field of computer vision.

I obtained the dataset from the GitHub repository MNIST-JPG. The dataset was organized into separate folders for each digit class. This helped in loading and preprocessing the data systematically.

3 Methodology

3.1 HOG Feature Extraction

I chose the HOG algorithm for feature extraction because of its ability to capture edge directions and gradients in images. This method provides a compact yet descriptive representation of image content. The parameters I used for HOG are:

- **Orientations:** 9
- **Pixels per cell:** 8x8
- **Cells per block:** 2x2
- **Block normalization:** L2-Hys

3.2 Data Loading and Preprocessing

I loaded the images using `cv2`, resized them to 28x28 pixels, and extracted HOG features. These features were then used as input for machine learning models.

3.3 Exploratory Data Analysis (EDA)

To understand the dataset better, I plotted the distribution of labels. The following figure shows that the dataset is balanced across all digit classes.

From this plot, I observed that the dataset is evenly distributed, ensuring fair training and evaluation of models.

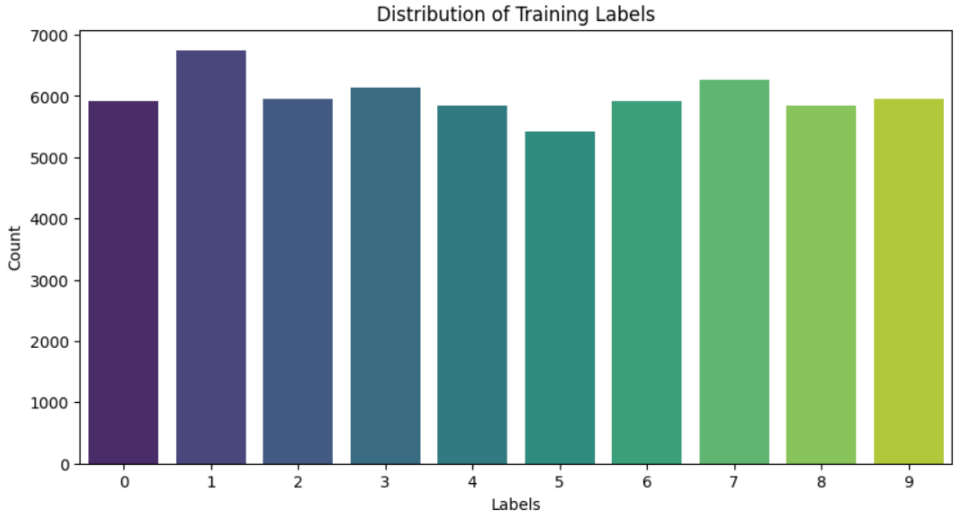

Figure 1: Distribution of Training Labels

4 Machine Learning Models

I experimented with several models:

1. Logistic Regression
2. Support Vector Machine (SVM)
3. Random Forest Classifier
4. XGBoost Classifier
5. Neural Network (Multi-Layer Perceptron)

For each model, I recorded the accuracy and analyzed the results using confusion matrices.

5 Hyperparameter Tuning

I performed hyperparameter tuning for each model to optimize performance:

- SVM: Kernel (`linear`), Regularization (`C=1.0`)
- Random Forest: Number of trees (`n_estimators=100`), Max depth
- XGBoost: Learning rate (`0.1`), Max depth (`6`)
- Neural Network: Hidden layers (`[100]`), Max iterations (`300`)

6 Results and Analysis

Here are the accuracy results for each model:

The XGBoost classifier achieved the highest accuracy, making it the best-performing model.

Model	Accuracy
Logistic Regression	93.2%
Support Vector Machine	94.6%
Random Forest	93.4%
XGBoost	96.1%
Neural Network	95.8%

Table 1: Model Performance Comparison

6.1 Confusion Matrices

Confusion matrices provide insights into the misclassification patterns. Below are the confusion matrices for top three model.

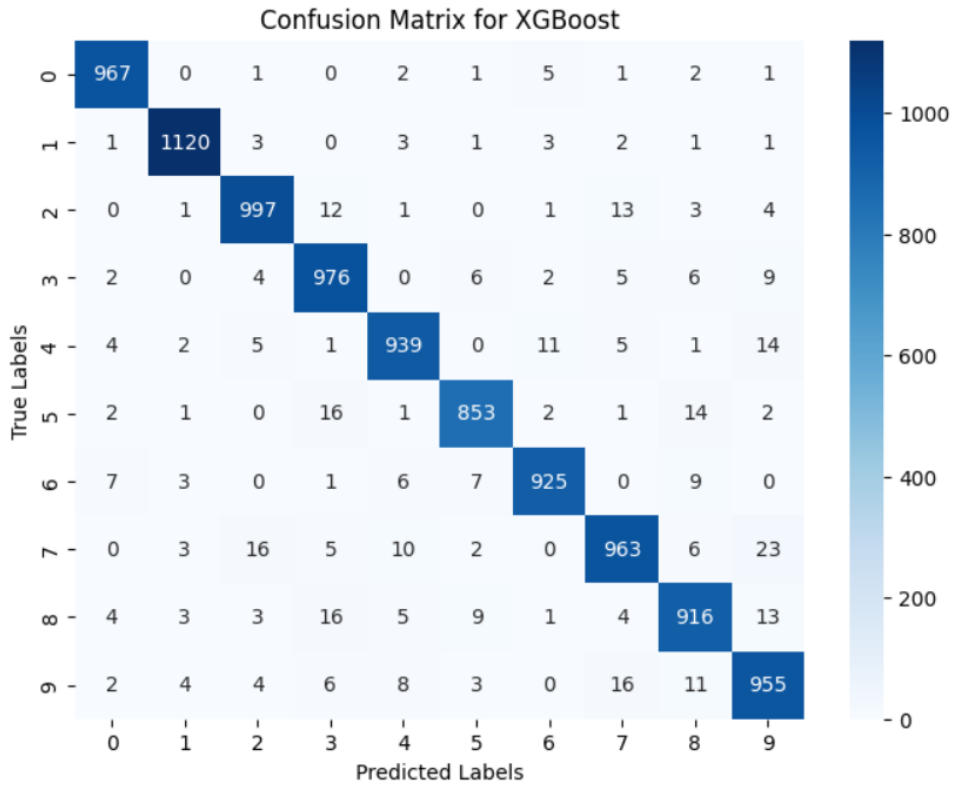

Figure 2: Confusion Matrix for XGBoost

The confusion matrix for the XGBoost model demonstrates its effectiveness in distinguishing between different digit classes. The matrix indicates a low misclassification rate, showcasing its ability to accurately predict the correct class for the majority of test samples.

- **Precision:** 96%
- **F1 Score:** 96%
- **Recall:** 96%
- **Accuracy:** 96.1%

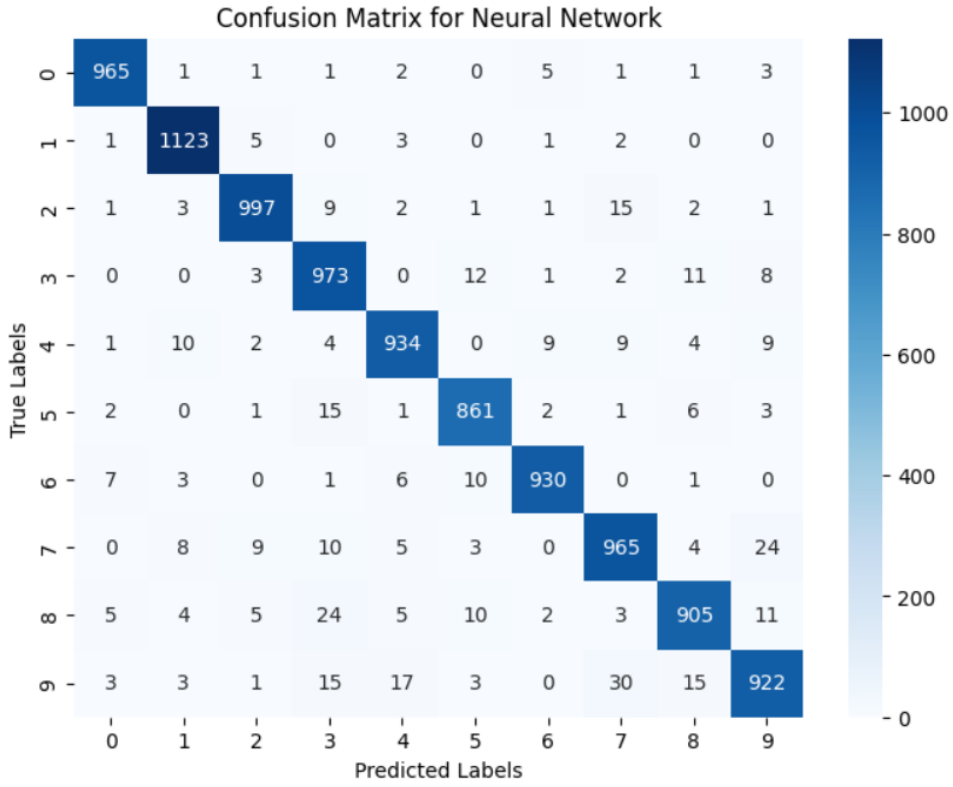

Figure 3: Confusion Matrix for Neural Network

The Neural Network confusion matrix also shows fewer errors, indicating its strong performance in accurately distinguishing the majority of digit classes. The model effectively captures the intricate patterns in the data, resulting in fewer misclassifications. This demonstrates the capability of neural networks to learn complex relationships, making them a competitive choice for classification tasks. Despite not being the top-performing model, its results highlight its potential in handling challenging datasets with high dimensionality.

- **Precision:** 96%
- **F1 Score:** 96%
- **Recall:** 96%
- **Accuracy:** 95.8%

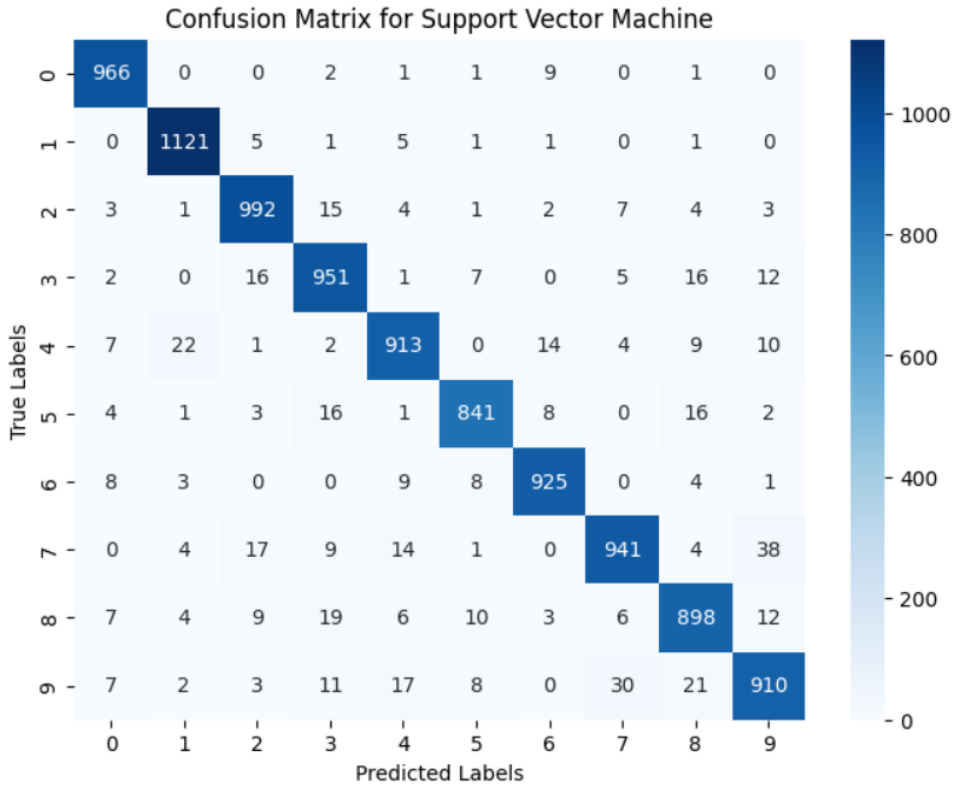

Figure 4: Confusion Matrix for SVM

From the SVM confusion matrix, it is evident that most misclassifications occur between visually similar digits, such as 3 and 8. This highlights SVM's challenges in distinguishing digits with overlapping features or shared characteristics.

- **Precision:** 93%
- **F1 Score:** 93%
- **Recall:** 93%
- **Accuracy:** 94.6%

6.2 Accuracy Comparison

The following figure compares the accuracies of all models.

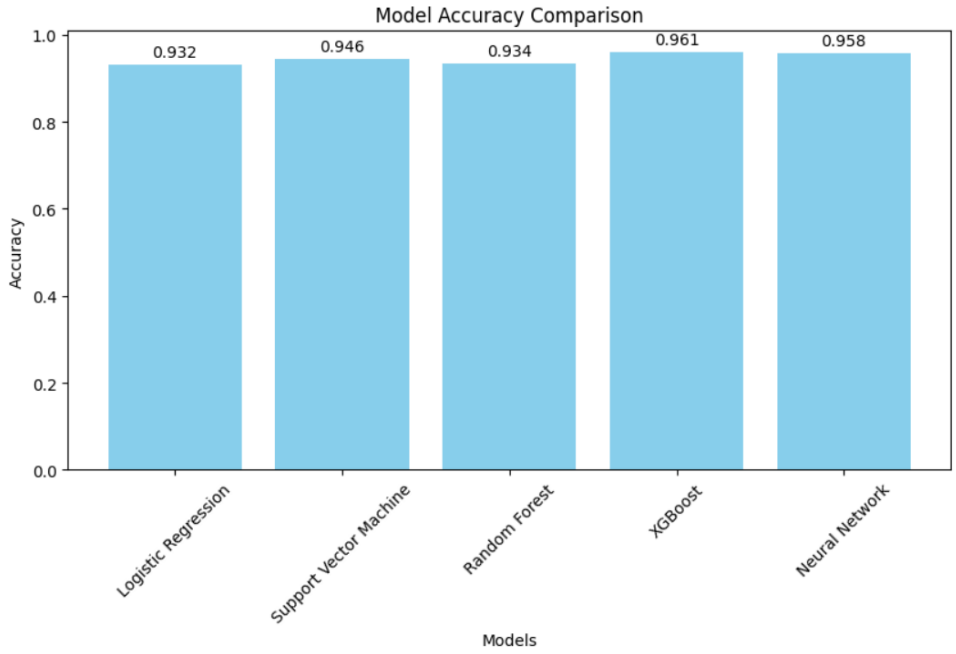

Figure 5: Model Accuracy Comparison

This graph highlights that XGBoost and Neural Network performed significantly better than other models.

7 Conclusion

I successfully implemented handwritten digit recognition using HOG features and multiple ML models. The XGBoost classifier emerged as the best model with an accuracy of 96.1%. Moving forward, I plan to:

- Experiment with Convolutional Neural Networks (CNNs) for improved accuracy.
- Apply data augmentation techniques to enhance model generalization.
- Further tune the HOG parameters for better feature extraction.

8 Code

Here is the complete code for the Homework Assignment on the given repo:

Code Link: Click Here