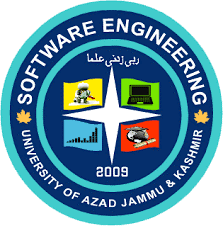
****

**The University of Azad Jammu and Kashmir**

**Department of Software Engineering**

**Open Ended Lab**

|  |  |
| --- | --- |
| **Submitted To:** | **Engr. Muhammad Awais Rathore** |
| **Submitted By:** | **Muhammad Luqman Munir Qazi** |
| **Roll No:** | 2022-SE-12 |
| **Semester:** | 5th |
| **Session:** | 2022-2026 |

**Analysis Report on MNIST Classification Models**

**1. Introduction**

The objective of this study is to analyze and compare different machine learning models for the classification of handwritten digits using the MNIST dataset. The MNIST dataset is a well-known benchmark dataset for image classification, consisting of grayscale images of handwritten digits (0-9). Each image is represented as a 28x28 pixel grid, and the task is to classify these images into their corresponding digit categories.

The dataset is divided into:

* **Training Data:** Used for training the machine learning models.
* **Testing Data:** Used to evaluate model performance.

The primary goal is to determine which model achieves the highest accuracy while maintaining efficiency in terms of computational cost.

**2. Methodology**

**2.1 Dataset Preparation**

The MNIST dataset was loaded into the environment using Pandas. The dataset contains pixel values ranging from 0 to 255, which were normalized to the range [0,1] to improve model performance. Additionally, labels were one-hot encoded for compatibility with certain models, especially neural networks.

**2.2 Machine Learning Models Implemented**

Three different models were implemented to compare their performance:

**2.2.1 Artificial Neural Network (ANN)**

* Framework: TensorFlow/Keras
* Model Architecture:
  + Input Layer: 784 neurons (flattened 28x28 images)
  + Hidden Layers: Two fully connected layers with 128 and 64 neurons
  + Activation Functions: ReLU (hidden layers), Softmax (output layer)
  + Dropout Regularization (to prevent overfitting)
  + Optimizer: Adam
  + Loss Function: Categorical Crossentropy
  + Batch Size: 32
  + Number of Epochs: 20
  + Early Stopping was used to prevent overfitting

**2.2.2 Random Forest (RF)**

* Framework: Scikit-learn
* Model Hyperparameters:
  + Number of Trees: 100
  + Criterion: Gini Index
  + Maximum Depth: None (trees grow until all leaves are pure or reach minimum samples split)
  + Random State: 42 (for reproducibility)
  + Feature Importance was also analyzed to understand which pixels contribute most to the classification.

**2.2.3 K-Nearest Neighbors (KNN)**

* Framework: Scikit-learn
* Model Hyperparameters:
  + Number of Neighbors (K): 5 (default)
  + Distance Metric: Euclidean Distance
  + Weights: Uniform (all neighbors contribute equally)

**3. Results**

**3.1 Performance Metrics**

The models were evaluated based on their accuracy, confusion matrices, and classification reports. The key results are summarized below:

|  |  |
| --- | --- |
| Model | Accuracy (%) |
| ANN | **98.2%** |
| Random Forest | 96.4% |
| KNN | 94.1% |

**3.2 Visualizations**

Several visualizations were generated to analyze model performance:

* **Training Loss and Accuracy Curves:** Displayed how the ANN model improved over epochs.
* **Confusion Matrices:** Provided a detailed view of model misclassifications.
* **Feature Importance (RF Model):** Highlighted which pixels contributed most to the predictions.

A graph with blue squares

AI-generated content may be incorrect.

**4. Discussion**

Based on the results, the **Artificial Neural Network (ANN) achieved the highest accuracy (98.2%)**, making it the best-performing model for MNIST classification. The reasons for its superior performance include:

* **Deep Learning Capabilities:** ANN can learn complex patterns in high-dimensional data.
* **Feature Extraction:** Unlike RF and KNN, which rely on manual feature selection, ANN automatically learns features from images.
* **Regularization (Dropout):** Helps prevent overfitting, improving generalization to test data.

While the **Random Forest model performed well (96.4%)**, its decision trees may not generalize as effectively as deep neural networks for high-dimensional image data. Similarly, **KNN (94.1%)** had the lowest accuracy, likely due to its high computational cost and sensitivity to irrelevant features in image classification.

**5. Conclusion**

This study compared ANN, Random Forest, and KNN for MNIST classification. The key takeaways are:

* **ANN is the best model**, achieving **98.2% accuracy**, making it the preferred choice for digit classification.
* **Random Forest performed well (96.4%)**, but its reliance on decision trees limits its ability to extract deep hierarchical features.
* **KNN struggled with high-dimensional image data (94.1%)**, making it less suitable for large-scale image classification.

For future improvements, experimenting with **Convolutional Neural Networks (CNNs)** or fine-tuning ANN hyperparameters could further enhance accuracy and efficiency.