**MNIST Digit Classification Report**

**1. Introduction**

The MNIST dataset is a well-known benchmark dataset in machine learning and deep learning, consisting of handwritten digit images (0-9). The objective of this project is to classify these digits using various machine learning and deep learning models, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest Classifier (RFC), and Convolutional Neural Networks (CNN).

**2. Dataset Preparation**

**2.1 Dataset Files**

The MNIST dataset consists of four files:

* train-images-idx3-ubyte
* train-labels-idx1-ubyte
* t10k-images-idx3-ubyte
* t10k-labels-idx1-ubyte

These files contain the training and test images along with their respective labels.

**2.2 Downloading the Dataset**

The dataset can be downloaded manually using the following commands:

curl -O http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz

curl -O http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz

curl -O http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz

curl -O http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz

Alternatively, the dataset can be extracted from a pre-downloaded dataset.zip file:

unzip dataset.zip -d dataset/

**2.3 Placing the Dataset in the Correct Folder**

The dataset should be placed inside the MNIST\_Dataset\_Loader/dataset/ directory under each algorithm folder:

KNN/

|\_ MNIST\_Dataset\_Loader/

|\_ dataset/

|\_ train-images-idx3-ubyte

|\_ train-labels-idx1-ubyte

|\_ t10k-images-idx3-ubyte

|\_ t10k-labels-idx1-ubyte

The same structure should be followed for SVM/ and RFC/ folders.

**3. Implementation Steps**

**3.1 Setting Up the Environment**

Before running the models, install the required dependencies:

pip install -r requirements.txt

**3.2 Running Machine Learning Models**

Navigate to the respective model directory and execute the Python script:

cd KNN

python knn.py

Repeat the steps for SVM and RFC models.

**3.3 Running the CNN Model**

The CNN model automatically downloads the dataset. Run the following command:

python CNN\_MNIST.py

To save the trained model weights:

python CNN\_MNIST.py --save\_model 1 --save\_weights cnn\_weights.hdf5

To load a previously saved model:

python CNN\_MNIST.py --load\_model 1 --save\_weights cnn\_weights.hdf5

**4. Troubleshooting Dataset Issues**

**4.1 Verifying Dataset Files**

To check if the dataset is correctly placed, run:

import os

dataset\_path = "KNN/MNIST\_Dataset\_Loader/dataset"

if os.path.exists(dataset\_path):

files = os.listdir(dataset\_path)

print("Dataset Files:", files)

else:

print("Dataset folder not found! Check your file paths.")

If the output does not show the expected files, follow these steps:

* Run os.listdir() to check the available directories.
* Use os.walk() to locate the dataset:

for root, dirs, files in os.walk("."):

print(root)

* Ensure that the dataset is in the correct folder and move it if necessary:

mv dataset KNN/MNIST\_Dataset\_Loader/

**5. Model Performance and Results**

**5.1 Accuracy Comparison**

| **Model** | **Accuracy** |
| --- | --- |
| KNN | 96.67% |
| SVM | 97.91% |
| RFC | 96.82% |
| CNN (TensorFlow) | 99.70% |
| CNN (Keras + Theano) | 98.75% |
|  |  |

**6. Conclusion**

This project demonstrates how different machine learning and deep learning algorithms perform on handwritten digit classification. The CNN model achieves the highest accuracy due to its ability to learn spatial hierarchies of features from images. Further improvements can be made using data augmentation and hyperparameter tuning.