CSE 474/574 Project 3
Team - [Rishabh Kanodiya, Shreyas Waikar]

Introduction-
The MNIST dataset is one of the most famous datasets used in machine learning and computer vision. It consists of a collection of 70,000 handwritten digits ranging from 0 to 9, divided into a standard part of 60,000 training images and 10,000 test images. Each number is represented as a 28x28 pixel grayscale image, where each pixel value indicates the lightness or darkness of that pixel, with higher numbers meaning darker. This dataset is widely used for machine learning training and testing due to its simplicity and size, making it an excellent benchmark for evaluating the performance of various image processing systems.

In an effort to explore and evaluate the performance of different classification algorithms on the MNIST dataset, this report dives into a comparative analysis of four distinct machine learning models: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNN ). Each of these models represents a different approach to the image classification problem, ranging from simple distance-based algorithms to complex hierarchical neural architectures.

Algorithms Used-

KNN (K-Nearest Neighbors):

The KNN is a strong candidate for the MNIST dataset as it is a straightforward algorithm, which is easy to interpret and implement. The images in the MNIST dataset are grayscale images with clear differences between the pixel values of their digits and their backgrounds, making it more favorable for KNN to be effective.
The KNN algorithm does not assume linearity in the data, as the shapes of the images in the MNIST dataset vary greatly, the flexibility of KNN to capture nonlinear boundaries between classes makes it a good fit for the dataset.
KNN is fast as there is no training period because it uses the entire dataset as the model for classification.
Given that each image consists of 28 * 28 pixels, totaling 784 dimensions per image, KNN can effectively operate in high-dimensional space.
Also, MNIST’s huge datasets can improve the performance of KNN by providing a lot of nearby points for the classification purpose.

SVM (Single Vector Machine):

SVM are particularly a good fit for the MNIST dataset for a lot of reasons, firstly SVM’s perform very well in high-dimensional spaces, such as the 784(28*28) pixel values of the MNIST images, as they are designed to find the best hyperplane that maximizes the margin between the classes.

SVM’s are known for their generalization capabilities, which means that they tend to perform well on unseen data, which is very useful when tackling problems like classification.
SVM’s have a major advantage of versatility through kernel functions which is a fundamental component of SVM that allows the algorithm to operate in a higher-dimensional space without explicitly computing the coordinates of the data in that space. This flexibility enables SVM to handle not only linearly separable data but also the data which is nonlinearly separable.

Random Forests:

Random Forests are effective in handling high-dimensional data by constructing multiple decision trees, each containing a random subset of features.
Random Forests are less likely to overfit due to the ensemble method of averaging multiple decision trees, which balances the bias-variance tradeoff.
The accuracy, especially in classification tasks when using random forests is a big highlight as when using random forests we get a good accuracy due to the way this algorithm operates. They create multiple trees, where each tree votes on the classification, and the majority vote decides the final class. This method results in a high level of accuracy.
Random Forests can capture nonlinear relationships between features and class labels and interactions between features without the need for manual feature engineering.
Considering these points, Random Forests are a robust choice for classifying MNIST dataset. The ability to handle various features without much preprocessing allows Random Forests eto excel in these situations.

CNN (Convolutional Neural Network):

Convolutional Neural Networks are well-suited for the MNIST dataset due to their architecture. CNN’s excel in capturing intricate patterns in the image data. CNN’s use filters to perform convolution operations that process small chunks of the input image, identifying patterns such as edges and shapes that are crucial for image classification.
CNN’s are advanced in nature as compared to traditional algorithms as they learn the important features directly from the image data. This ability to learn features at various levels is a major advantage.
CNN’s are robust to small translations and other forms of distortion in the input images. This property is essential for handling handwritten digits in MNIST, where each digit can vary in style and orientation.
CNN’s reduce the number of parameters involved in the learning process through shared weights and pooling layers. This property not only makes CNN computationally efficient but also reduces the risk of overfitting.
Given these properties, CNN’s are naturally a good choice when classifying images for MNIST dataset.