ML techniques for Handwritten Recognition:

1: Logistic Regression/ SoftMax Regression:

Treats each pixel as a feature and predicts digit by calculating probability across 10 classes. It is Fast, interpretable and good baseline. Its limited capacity is its weakness.

2: K- Nearest Neighbours (K-NN):

Classifies a digit by looking at the “closest” training samples in pixel space. Simple, no training needed, decent accuracy. Slow for large datasets (needs to compare with many samples), memory heavy.

3: Support Vector Machines (SVMs):

Finds an optimal decision boundary (hyperplane) that separates digit classes. Strong theoretical guarantees, high accuracy. Very computationally expensive on large datasets like MNIST.

4: Decision Trees & Random Forests:

Trees split on pixel values to classify digits. Random forests (ensembles of trees) improve robustness. Easy to train, interpretable, works with noisy data. Low accuracy compared to SVM.

5: Multi-Layer Perceptron (MLP):

A fully connected neural network with hidden layers. Learns abstract features from raw pixels. Better generalization, accuracy. Needs tuning (hidden layers, learning rate), slower training than SVM/logistic regression.

6: Convolutional Neural Networks (CNNs):

Specialized for image data. Uses convolutional filters to detect strokes, edges, and patterns. State-of-the-art, captures spatial patterns, more complex, requires more computation (but feasible today).

# Which is Best?

1: Baseline models (Logistic Regression, k-NN, Decision Trees): Good for showing progression, but not final.

2: SVM: High accuracy, but slow for large datasets.

3: MLP: Great balance of accuracy.

4: CNN: Best accuracy, widely considered the gold standard for handwritten digit recognition.