**Image Classification Using Convolutional Neural Networks**

**Problem Statement:**

This Project aims to classify images of handwritten digits (MNIST dataset) using Convolutional Neural Network (CNN). The project is to build a model that can accurately identify digits from 0 to 9 demonstrating the fundamentals of CNNs and deep learning.

**Dataset:**

The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST dataset is a collection of 70,000 handwritten digits (0-9), with each image being 28\*28 pixels.

Training set: 60,000 images

Test set : 10,000 images

**Approach:**

**Dataset Loading**

* Load the MNIST dataset using TensorFlow’s built-in function.
* Visualize a sample of the image to understand the dataset structure.

**Data Processing**

* Reshaped the data to fit the model's input requirements.
* Normalized or standardized the data to improve model performance.
* Performed one-hot encoding of labels for categorical classification.

**Data Augmentation**

* Techiques like rotation, zooming and flipping can increase data diversity

**Model Development**

**Model Architecture:**

* + Designed the CNN architecture, including:
  + Number of convolutional layers
  + Number of filters in each layer
  + Kernel sizes for convolutional layers
  + Pooling layer types (max-pooling, average-pooling)
  + Fully connected layers and their sizes
  + Activation functions (ReLU, softmax)

**Model Implementation:**

* + Implemented the chosen CNN architecture using a deep learning framework (e.g., TensorFlow/Keras).

**Model Training and Evaluation**

**Model Compilation:**

* + Defined the loss function (e.g., categorical crossentropy).
  + Selected an optimizer (e.g., Adam).
  + Specified evaluation metrics (e.g., accuracy).

**Model Training:**

* + Trained the model on the training data.
  + Used techniques like data augmentation (if applicable) to improve model robustness.
  + Monitored training progress using validation data.

**Model Evaluation:**

* + Evaluated the trained model on the test data.
  + Calculated performance metrics: accuracy, precision, recall, F1-score.
  + Analysed the results and identified areas for improvement.

**Save and Deploy the Model**

* Saved the trained model to a file
* Deployed the model using AWS