Report: Handwritten Digit Recognizer

Student Name: Jay Patil **Date:** August 11, 2025

---

1. Project Objective

The goal of this project is to build, train, and evaluate a Convolutional Neural Network (CNN) to classify handwritten digits from the MNIST dataset. The program must demonstrate a complete machine learning workflow, from data preparation to visual confirmation of the model's predictions.

---

2. Solution Architecture and Logic

The solution is implemented using TensorFlow and its high-level Keras API, which is the industry standard for building and training deep learning models.

- **Data Handling:** The standard MNIST dataset is loaded directly from tf.keras.datasets. The image data undergoes two essential preprocessing steps:
    1. **Normalization:** Pixel values are scaled from their original 0-255 range to a 0-1 range. This helps the model's optimizer converge more effectively during training.
    2. **Reshaping:** The images are reshaped to include a channel dimension, which is the expected input format for a CNN.
- **CNN Architecture:** The core of the solution is a sequential CNN model designed for image classification. The architecture is as follows:
    - **Two Convolutional Blocks:** Each block consists of a Conv2D layer followed by a MaxPooling2D layer. The convolutional layers act as feature extractors, identifying patterns like edges, curves, and loops. The max-pooling layers then downsample the feature maps, reducing computational complexity and making the detected features more robust.
    - **Flatten Layer:** This layer converts the 2D feature maps into a 1D vector.
    - **Dropout Layer:** A dropout layer is included to prevent overfitting, which occurs when a model memorizes the training data instead of learning general patterns.
    - **Dense Output Layer:** The final layer is a Dense layer with 10 neurons and a softmax activation function. It outputs a probability distribution across the 10 possible digits (0-9).
- **Training and Evaluation:** The model is compiled with the adam optimizer and sparse_categorical_crossentropy loss function, which are standard choices for multi-class classification. It is trained for 5 epochs. After training, the model's final accuracy is evaluated on the unseen test set.
- **Prediction and Visualization:** To provide a clear visual demonstration of the model's capability, the program makes predictions on random samples from the test set. It then uses matplotlib to display these images, titling each one with its true label and the model's predicted label. Correct predictions are colored green, and incorrect ones are colored red for immediate visual feedback.

---

3. Conclusion

This program successfully implements an end-to-end image classification pipeline. The well-structured CNN model achieves high accuracy (typically >98%) on the MNIST dataset, and the final visualization provides compelling proof of its effectiveness in recognizing handwritten digits.