

# Handwritten Digit Recognition Using CNN & OpenCV

## Project Overview

This project demonstrates a simple digit recognition system using a convolutional neural network (CNN) trained on the MNIST dataset. It allows users to draw a digit (0-9) on a virtual canvas using OpenCV, and uses TensorFlow to predict the digit with confidence.

It is built in Python with libraries like TensorFlow, NumPy, and OpenCV.

## Libraries Used

1. TensorFlow/Keras - For building and training the CNN model.
2. OpenCV - For drawing interface and real-time interaction.
3. NumPy - For numerical operations.

## CNN Model Logic

1. Load MNIST dataset, normalize pixel values (0-255 to 0-1), and reshape images.
2. Build a CNN with two Conv2D and MaxPooling2D layers, followed by Flatten and Dense layers.
3. Train for 10 epochs, then save the model as `mnist_cnn_model.keras`.

If the model is already trained, it loads directly without retraining.

## Drawing Interface Logic (OpenCV)

1. A 400x400 black canvas is created using NumPy.
2. The user draws with the mouse, and OpenCV captures the strokes using `setMouseCallback()`.
3. Press 'p' to predict, 'c' to clear the board, and 'q' to quit.

When 'p' is pressed:

- The image is resized to 28x28 pixels.
- The color is inverted to match MNIST format (white digit on black background).
- It is reshaped and passed to the model for prediction.
- The result and confidence are printed to the console.

## **Summary**

This project demonstrates the practical integration of machine learning with computer vision. It's a foundational example for beginners exploring handwritten digit classification using neural networks.

The logic can be extended to recognize alphabets, gestures, or more complex drawings using similar preprocessing techniques and more advanced models.