1. **The Model File:** The code is *not* running the training process in train\_model.py when you run main.py. Instead, it's using a pre-trained model that is saved in the "mnist\_model.h5" file.
2. **Training the Model (Manually):** If you want to train a new model with different data or parameters, you would run the code in train\_model.py. This would create a new "mnist\_model.h5" file with the trained model.

**Here's why it works this way:**

* **Training is Time-Consuming:** Training a deep learning model can take a significant amount of time, depending on the size of the dataset, the complexity of the model, and your hardware.
* **Saving and Loading:** The train\_model.py file is designed to be run once (or occasionally) to train the model. Then, the trained model is saved, so you don't have to train it again every time you want to use it.
* **main.py for Recognition:** The main.py file focuses on the user interface and using the *already trained* model for digit recognition. It loads the model from "mnist\_model.h5" and then uses it to predict digits from the images the user selects.

**How to Use the Code:**

1. **Train a Model (Optional):** If you need to train a new model, run the code in train\_model.py to create "mnist\_model.h5".
2. **Run the Application:** Execute main.py to use the pre-trained model.

**Warning**: The images that are acceptable are images that are initially 28 x 28 pixel size like the sample images that are supplied in the directory ./pictures/handwritten unless this requirement is meet the images will get the wrong prediction but in some cases like the images in the directory ./pictures/correct\_predections where you get the right answer.

Also you must have the following packages to run this code properly:

>>> pip install tensorflow keras opencv-python matplotlib pillow numpy scipy

This process will take time.