**Project 2: MNIST Classification Using Feedforward and Convolutional Neural Networks**

**Project Overview:**

This project aims to classify handwritten digits from the MNIST dataset using two neural network architectures: a feedforward neural network (FFNN) with fully connected layers and a convolutional neural network (CNN). The goal is to achieve a testing accuracy of 95% or higher with high probability. The project uses PyTorch for model implementation and is executed in a Google Colab environment.

**Datasets:**

* The MNIST dataset consists of grayscale images of handwritten digits (0–9) for classification, with 60,000 training images (train-images.idx3-ubyte) and corresponding labels (train-labels.idx1-ubyte), along with 10,000 testing images (t10k-images.idx3-ubyte) and their labels (t10k-labels.idx1-ubyte).
* Each image is of size 28×28 pixels, representing a single handwritten digit, and the labels are integers in the range {0, 1, ..., 9} corresponding to the digit depicted in the image.
* The dataset is provided in .idx format, and the goal is to use this data to train and test two neural network architectures: a feedforward neural network with at least two hidden layers and a convolutional neural network with at least two convolutional layers and two fully connected layers.

**Files included :**

1. train-images.idx3-ubyte (Training images)(included in the data folder)

2. train-labels.idx1-ubyte (Training labels)(included in the data folder)

3. t10k-images.idx3-ubyte (Testing images)(included in the data folder)

4. t10k-labels.idx1-ubyte (Testing labels)(included in the data folder)

5. Code notebook: DATA603\_Project-2\_MNIST\_Classification\_using\_FNN\_and\_CNN.ipynb

6. Report: DATA603\_Project-2\_MNIST\_Classification\_using\_FNN\_and\_CNN\_Report.pdf

**Steps to run the notebook :**

1. Open the file `DATA603\_Project-2\_MNIST\_Classification\_using\_FNN\_and\_CNN.ipynb` in Google Colab or Jupyter Notebook.

2. Upload the MNIST dataset files to the same directory as the notebook.

3. Execute all the cells sequentially to:

- Train the FFNN and CNN architectures.

- Evaluate their performance over 5 runs.

- View average accuracies and confusion matrices.

4. Results, including plots of accuracy and loss metrics, will be displayed in the output cells.

**Required dependencies:**

PyTorch , torchvision ,matplotlib , numpy, sklearn