# WEEK 14 - CNN IMAGE

NAME: MEGHANA SAISRI BISA
SEM 5, SEC F

## 1. Introduction

The objective of this lab was to design, build, train, and evaluate a Convolutional Neural Network (CNN) using PyTorch to classify images of hand gestures into three categories: rock, paper, and scissors. The dataset provided consists of labeled images stored in separate folders, making it ideal for using PyTorch's ImageFolder utility. The completed notebook downloads the dataset, applies preprocessing, builds a CNN model, trains it, and finally evaluates its performance.

---

## 2. Model Architecture

The CNN model built for this classification task consists of three convolutional blocks, each followed by a ReLU activation and MaxPooling:

1. Conv Block 1:
    - Input channels: 3
    - Output channels: 16
    - Kernel size: 3 × 3
    - Padding: 1
    - Activation: ReLU
    - MaxPool: 2 × 2
2. Conv Block 2:
    - Input channels: 16
    - Output channels: 32
    - Kernel size: 3 × 3
    - Padding: 1
    - Activation: ReLU
    - MaxPool: 2 × 2
3. Conv Block 3:
    - Input channels: 32
    - Output channels: 64
    - Kernel size: 3 × 3
    - Padding: 1

    - Activation: ReLU
    - MaxPool: 2 × 2

After the convolution layers, the spatial dimensions shrink from 128 → 64 → 32 → 16, giving a final feature map of size:

64 × 16 × 16 = 16,384 features

Fully Connected Classifier

The classifier consists of:

- Flatten layer
- Linear (16,384 → 256)
- ReLU activation
- Dropout (p = 0.3)
- Linear (256 → 3) for the three gesture classes.

This architecture allows the network to extract visual features through convolution layers and then classify them using fully connected layers.

---

**3. Training and Performance**

Hyperparameters Used

- Loss Function: CrossEntropyLoss
- Optimizer: Adam
- Learning Rate: 0.001
- Epochs: 10
- Batch Size: 32
- Transformations:
    - Resize to 128×128
    - Convert to Tensor
    - Normalize(mean=0.5, std=0.5)

**Final Test Accuracy: 98.63%**

## 4. Conclusion and Analysis

The CNN performed reasonably well on the Rock–Paper–Scissors classification task, achieving good overall accuracy. The model successfully learned distinguishable features of each gesture because the dataset is clean and well-structured.

Some challenges included correctly configuring the dataset paths, ensuring the transforms were applied consistently, and debugging initialization issues such as missing device, train_dataset, or uninitialized model components.

To further improve the model accuracy, you could:

1. Add Data Augmentation
   Techniques like random rotations, flips, and brightness changes could help the model generalize better.
2. Use a deeper CNN or transfer learning
   Pretrained models like ResNet18 or MobileNet can significantly boost accuracy.
3. Increase training epochs
   Training for 20–30 epochs could allow the model to learn richer features.