1. Introduction

The objective of this lab was to build and train a Convolutional Neural Network (CNN) to classify hand gestures representing rock, paper, and scissors. Using a labelled image dataset, the goal was to design a model that could accurately recognize these gestures and generalize well to unseen examples, simulating a basic visual recognition task.

2. Model Architecture

The CNN architecture consists of three convolutional blocks followed by a fully-connected classifier:

Convolutional Blocks:

Block 1: Conv2d(3 -> 16 channels, kernel size=3, padding=1), followed by ReLU and MaxPool2d(kernel size=2)

Block 2: Conv2d(16 -> 32 channels, kernel size=3, padding=1), followed by ReLU and MaxPool2d(kernel size=2)

Block 3: Conv2d(32 -> 64 channels, kernel size=3, padding=1), followed by ReLU and MaxPool2d(kernel size=2)

These layers progressively reduce the spatial dimensions from 128×128 to 16×16 while increasing feature depth.

Fully-Connected Classifier:

The output of the final convolutional block is flattened to a vector of size 64 × 16 × 1= 16384

A Linear layer maps this to 256 units, followed by ReLU and Dropout(p=0.3)

A final Linear layer maps to 3 output units, corresponding to the three gesture classes

3. Training and Performance

Optimizer: Adam

Loss Function: CrossEntropyLoss

Learning Rate: 0.001

Epochs: 10

After training for 10 epochs, the model achieved a **Test Accuracy of 97.03%**, indicating strong performance on the classification task

4. Conclusion and Analysis

The model performed very well, achieving high accuracy with a relatively simple architecture. The training loss decreased steadily across epochs, and the final test accuracy suggests good generalization. One challenge was ensuring consistent preprocessing, especially normalization and resizing, to match training conditions during inference. To further improve accuracy, data augmentation techniques (e.g., rotation, flipping) could be introduced to increase robustness, and experimenting with deeper architectures or fine-tuning dropout rates might help reduce overfitting

1