ML LAB WEEK 14
CNN Image Classification

Name: Cheruku Manas Ram
SRN: PES2UG23CS147
Section: 5 ‘C’
Date: 20/11/2025

1. Introduction

The objective of this lab was to build, train, and test a Convolutional Neural Network (CNN) capable of classifying images of hands playing rock, paper, or scissors. This involved setting up the environment, downloading the dataset, preprocessing the image data, defining a custom CNN architecture, training the model, and finally evaluating its performance on an unseen test set.

2. Model Architecture

The CNN model, named RPS_CNN, consists of two main parts: a convolutional block for feature extraction and a fully connected block for classification.

Convolutional Block:
This block comprises three sequential sets of Conv2d, ReLU, and MaxPool2d layers:
- First Layer: Conv2d with 3 input channels (for RGB images) to 16 output channels, a kernel size of 3, and padding of 1. Followed by ReLU activation and MaxPool2d with a kernel size of 2.
- Second Layer: Conv2d with 16 input channels to 32 output channels, a kernel size of 3, and padding of 1. Followed by ReLU activation and MaxPool2d with a kernel size of 2.
- Third Layer: Conv2d with 32 input channels to 64 output channels, a kernel size of 3, and padding of 1. Followed by ReLU activation and MaxPool2d with a kernel size of 2.

Each MaxPool2d layer halves the dimensions of the feature maps. Starting with 128x128 images, after three layers, the dimensions become 16x16 (128 -> 64 -> 32 -> 16).

Fully Connected Classifier:
This block takes the flattened output from the convolutional layers and performs the final classification:
- Flatten Layer: Converts the 3D feature maps (64 channels 16x16 spatial dimensions = 16384 features) into a 1D vector.
- First Linear Layer: A fully-connected layer that maps 16384 input features to 256 output features.
- ReLU Activation: Applies the Rectified Linear Unit activation function.
- Dropout Layer: A dropout layer with a probability of 0.3, used for regularization to prevent overfitting.
- Second Linear Layer (final): A fully connected layer that maps 256 features to 3 output features, corresponding to the three classes: Rock, Paper, and Scissors.

3. Training and Performance

Hyperparameters for training:
- Optimiser: Adam
- Loss Function: CrossEntropyLoss
- Learning Rate: 0.001
- Number of Epochs: 10

The dataset was split into 80% for training (1750 images) and 20% for testing (438 images). Image preprocessing included resizing to 128x128, converting to PyTorch tensors, and normalising with a mean and standard deviation of 0.5 for each color channel.

After training for 10 epochs, the model achieved a 97.49% accuracy on the test set.

4. Conclusion and Analysis

The model performed very well, with a test accuracy of 97.49%. This accuracy suggests that the chosen CNN architecture and training parameters were effective in learning discriminative features for classifying rock, paper, and scissors gestures from images. The loss decreased consistently during training, indicating successful convergence.

One challenge encountered during development was an initial RuntimeError due to an incorrect device string ('CPU' instead of 'cpu') when moving the model to the computing device. This was resolved by simply making ‘cpu’ lowercase.

Future improvements:
1. Data Augmentation: Applying more aggressive data augmentation techniques (like random rotations, horizontal/vertical flips) during training could make it more robust and generalise to a wider variety of image conditions.
2. Hyperparameter Tuning: Experimenting with different learning rates, optimiser choices, batch sizes and dropout rates could lead to small improvements in accuracy or faster convergence. Techniques like grid search or random search could be employed for this.