AIM:-

In this project, you will build a model to recognise 5 hand gestures.

Different Kinds of models used:-

For analysing videos using neural networks, **two types of architectures**are used commonly. One is the standard **CNN + RNN architecture** in which you pass the images of a video through a CNN which extracts a feature vector for each image, and then pass the sequence of these feature vectors through an RNN. This is something you are already familiar with (in theory).

The other popular architecture used to process videos is a natural extension of CNNs - a **3D convolutional network**. In this project, you will try both these architectures.

1. **3D Convolutional Network, or Conv3D**  
   3D convolutions are a natural extension to the 2D convolutions you are already familiar with. Just like in 2D conv, you move the filter in two directions (x and y), in 3D conv, you move the filter in three directions (x, y and z). In this case, the input to a 3D conv is a video (which is a sequence of 30 RGB images). If we assume that the shape of each image is 100x100x3, for example, the video becomes a 4-D tensor of shape 100x100x3x30 which can be written as (100x100x30)x3 where 3 is the number of channels. Hence, deriving the analogy from 2-D convolutions where a 2-D kernel/filter (a square filter) is represented as (fxf)xc where f is filter size and c is the number of channels, a 3-D kernel/filter (a 'cubic' filter) is represented as (fxfxf)xc (here c = 3 since the input images have three channels). This cubic filter will now '3D-convolve' on each of the three channels of the (100x100x30) tensor.
2. **Convolutions + RNN**  
   The conv2D network will extract a feature vector for each image, and a sequence of these feature vectors is then fed to an RNN-based network. The output of the RNN is a regular softmax (for a classification problem such as this one).

Results:-

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1st** | **Conv3D** | **Train Accuracy: 0.85**  **Test Accuracy: 0.26** | **Model is overfitting. Train accuracies are much higher compared to Validation Accuracies.**  **Try ConvLSTM as Conv3D not giving desired accuracy** |
| **2nd** | **ConvLSTM** | **Train Accuracy: 0.56**  **Test Accuracy: 0.36** | **Model is still overfitting but it is overfitting much lesser.** |