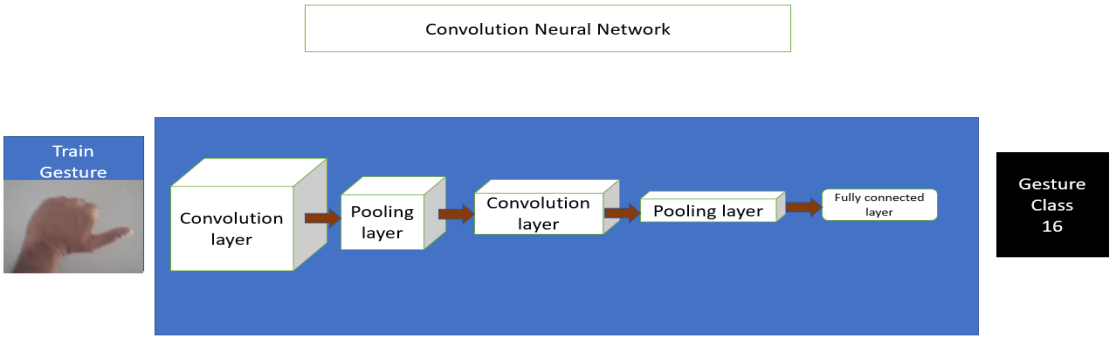
**Gesture Classification using CNN**

This project is about classifying videos to identify the predetermined hand gestures by using CNN machine learning technique

**Theoretical Description:**

The gesture videos are used to train a convolution neural network model developed using the tensor flow and kera modules. The CNN module uses processed images to produce feature vectors for each image and assign a predetermined image classification labels as per the below table.

CNN is made up of multiple layers, the main layers are convolutional layer, pooling layer, and fully connected layer. Each of these layers take input data and produce an output.




**Convolution Layer:**

It is the feature extractor layer. In this layer we pass the gesture video and a corresponding image vector is generated using the Opencv module. These image feature vectors will be then stored in a pandas dataframe.

**Pooling Layer:**

Pooling layer will use a patch filter to reduce the input another ouput so as to apply dimensional reduction into the image.

**Fully Connected layer:**

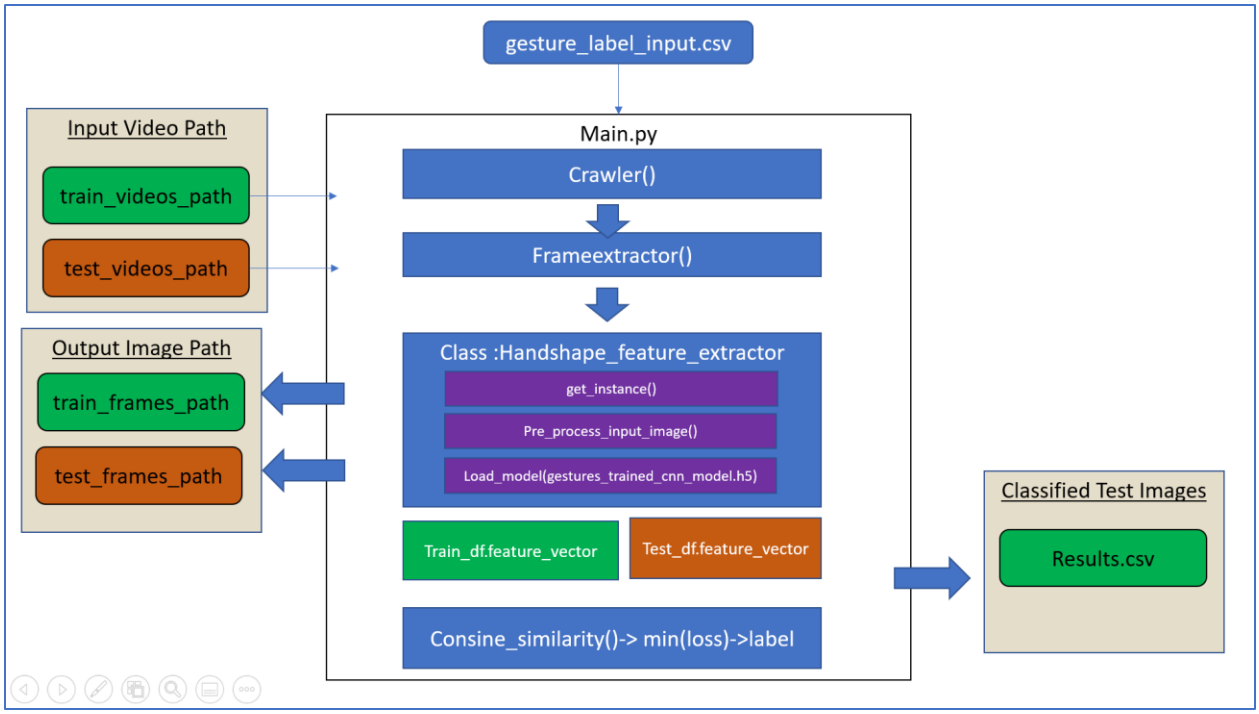
In fully connected layer the output that's produced at the end of the last pooling layer is an input to each node in this fully-connected layer

**Program Structure:**

1. The input training videos are placed in the train videos folder.
2. The program crawls the entire directory and then picks the video file in a loop
3. The penultimate layer is captured using OpenCV and stored to frames folder
4. The images captured this way are passed through the feature extractor
5. A numpy array is generated for each gray scale image and passed through the kera model

```
keras = tf.keras
load_model = keras.models.load_model
Model = keras.models.Model
```

6. Pandas Dataframes are generated for both training and testing images
7. Cosine Similarity is derived and the min loss factor is calculated and corresponding label is assigned to the testing data
8. Result.csv is generated as an output




**Imported Modules:**

```
import cv2
import numpy as np
import os
import tensorflow as tf
```

```
import frameextractor as fe
import handshape_feature_extractor
import crawler
import pandas as pd
```

**Input Folders:**

- Train_video_path
- Train_frames_path
- Test_video_path
- Test_frames_path

**Reference Files:**

The gesture_label_input.csv file is used to assign label to the training feature factor based on the file name.

| train_gesture_name | test_gesture_name | output_label |
|---|---|---|
| H00_PRACTICE_01_PARIDA | T1-H-0 | 0 |
| H01_PRACTICE_01_PARIDA | T1-H-1 | 1 |
| H02_PRACTICE_01_PARIDA | T1-H-2 | 2 |
| H03_PRACTICE_01_PARIDA | T1-H-3 | 3 |
| H04_PRACTICE_01_PARIDA | T1-H-4 | 4 |
| H05_PRACTICE_01_PARIDA | T1-H-5 | 5 |
| H06_PRACTICE_01_PARIDA | T1-H-6 | 6 |
| H07_PRACTICE_01_PARIDA | T1-H-7 | 7 |
| H08_PRACTICE_01_PARIDA | T1-H-8 | 8 |
| H09_PRACTICE_01_PARIDA | T1-H-9 | 9 |
| DecreaseFanSpeed_PRACTICE_01_PARIDA | T1-H-DecreaseFanSpeed | 10 |
| FanOn_PRACTICE_01_PARIDA | T1-H-FanOff | 11 |
| FanOff_PRACTICE_01_PARIDA | T1-H-FanOn | 12 |
| IncreaseFanSpeed_PRACTICE_01_PARIDA | T1-H-IncreaseFanSpeed | 13 |
| LightOff_PRACTICE_01_PARIDA | T1-H-LightOff | 14 |
| LightOn_PRACTICE_01_PARIDA | T1-H-LightOn | 15 |
| SetThermo_PRACTICE_01_PARIDA | T1-H-SetThermo | 16 |