1 Problem Statement –

To create an innovative feature for the smart TV platform which aims to accurately identify and interpret five distinct gestures.

- **Thumbs up**: Increase the volume
- **Thumbs down**: Decrease the volume
- **Left swipe**: 'Jump' backwards 10 seconds
- **Right swipe**: 'Jump' forward 10 seconds
- **Stop**: Pause the movie

2. Dataset –

Here after the problem statement, we used a dataset which consists of 2 main folders i.e. training data folder and validation data folder.

- Training Data consists of few hundred videos categorized into one of the five classes. Each video is divided into a sequence of 30 frames (images).
- Few of them have the dimensions as 360x360 or 120x120.

3. Code Approach –

We've used the 3D CNN (Conv3D) approach for the given problem.

- Importing necessary libraries.
- Storing the image names in the lists, validation_doc and training_doc.
  Setting batch_size as 25 initially.
- Generator – A function with parameters of number of frames, source_path, folder_list, batch_size and image_size plays an important role in the project as below –
  1) Iterate over the number of images in the folder, read, resize and normalize the images.
  2) Yields the batch_data and batch_labels.
- History Plot – A function with parameter names "history" basically have the trained model fit, consists of plot of loss and accuracy.
- Model Building.

4. Results –

Experiment Number	Model	Result	Decision + Explanation
1	Conv3D	Training accuracy:0.88 Validation accuracy:0.62 *Parameters: 997,125*	The model exhibited overfitting, as evidenced by a higher training accuracy compared to the validation accuracy. **Decision:** Added more Conv3D layers to reduce overfitting in the next experiment.
2	Conv3D	Training accuracy:0.75 Validation accuracy:0.27 *Parameters: 1,922,661*	No improvement from the previous model, model is clearly overfitting. **Decision:** Increased epochs to 25 and added Batch Normalization after Dense layers to reduce overfitting in the next experiment
3	Conv3D	Training accuracy:0.95 Validation accuracy:0.85 *Parameters: 3,996,517*	Significant improvement in both training and validation accuracies. **Decision:** Reduced number of frames to 10, batch size to 20, and reduced dropout to 0.25 in the dense layers. Planning to check if performance is retained with fewer frames in the next model.
4	Conv3D	Training accuracy:0.99 Validation accuracy:0.91 *Parameters: 3,987,813*	Although the model achieved high accuracy, there remained a noticeable gap between training and validation performances, indicating potential overfitting. **Decision:** Increased image size to (160,160) in the next experiment to reduce the gap between training and validation accuracy
5	Conv3D	Training accuracy:0.99 Validation accuracy:0.89 *Parameters: 6,870,181*	No improvement in model by changing image size. **Decision:** so continuing with model in experiment-4 and reducing number of trainable parameters by reducing the filter size of conv3d in the next experiment to simplify the model complexity to some extent.
6	Conv3D	Training accuracy:0.97 Validation accuracy:0.94 *Parameters: 2,048,997*	The model achieved exceptional performance on both training and validation datasets. **Decision:** Selecting this model as the final one due to its excellent performance on both datasets with significantly less number of trainable parameters.