**Gesture Recognition Project**

**By Mahalakshmi Totad**

# **Problem Statement**

Develop a cool feature in the smart-TV that can recognize five different gestures performed by the user which will help users control the TV without using a remote.

# **Understanding the Dataset**

The training data consists of a few hundred videos categorized into one of the five classes. Each video (typically 2-3 seconds long) is divided into a sequence of 30 frames(images).

Note that all images in a particular video subfolder have the same dimensions but different videos may have different dimensions. Specifically, videos have two types of dimensions - either 360x360 or 120x160 (depending on the webcam used to record the videos). Hence, you will need to do some pre-processing to standardise the videos.

Each row of the CSV file represents one video and contains three main pieces of information - the name of the subfolder containing the 30 images of the video, the name of the gesture and the numeric label (between 0-4) of the video.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

* 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.

# **Two types of architectures**

1. 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.
2. CNNs - a 3D convolution network , the other popular architecture used to process videos

# **Experiment Details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp No** | **Model** | **Result** | **Observation** | **Decision + Explanation** |
| Conv3D Architecture | | | | |
| 1 | CNN-3D | 3 epochs run successfully. | Model complies and runs with given parameters. | **Decision/Explanation** : Run model for 40 epochs to get the desired results |
| 2 | CNN-3D | Total params: 1866181  Training loss: 0.0902  Training accuracy: 0.9970  Validation loss: 1.4247  Validation accuracy: 0.45 | - Training accuracy is high where as validation accuracy is very low.  - Model seems to be highly over fitting.  - Validation loss has reduced as number of epochs have increased indicating good sign of learning by the model. | **Decision** : Adding a layers to improved the accuracy  **Explanation** : To reduce over fitting and get better results |
| 3 | CNN-3D | Total Params: 900805  Training loss: 0.3445  Training accuracy: 0.8899  Validation loss: 0.6020  Validation accuracy: 0.81 | - Model seems to have learnt more with increased number of epochs.  - Also with less number of parameters. | **Decision** : Change filters to 8,32,64 and remove 128f layer  - Reducing the total number of parameters  **Explanation** :  - Let’s see if we can reduce validation loss further |
| 4 | CNN-3D | Total Params: 480,725  Training loss: 0.0879  Training accuracy: 0.9985  Validation loss: 0.5717  Validation accuracy: 0.8200 | - Validation loss is less by 0.03 and accuracy is more by 0.01 than Model 3 which is a good sign for the model, but Training accuracy is 0.9985 to Validation accuracy of 0.82. Thus indicating that Model is over fitting with less number of parameters | **Decision** : Running for 40 epochs, batch size of 20 and frames per video of 30  **Explanation** : Experiment with frames per second to see if it’s has impact on performance |
| 5 | CNN-3D | Total Params: 896,389  Training loss: 0.2692  Training accuracy: 0.9382  Validation loss: 0.6012  Validation accuracy: 0.8200 | - Increasing number of frames per video has increased the training accuracy from 0.88 to 0.93 has increased the validation accuracy from 0.81 to 0.82  - has reduced the validation loss from 0.6020 to 0.6012  - Not a significant improvement in terms of validations results. Let's change try a different model | **Decision** : Add Dropout of 0.25 to couple of layers and , adding dense layer with 256 and filter (3,3,3)  **Explanation** : Adding dropout to see if more regularization is required. And one more convo3D layer to improve accuracy and loss |
| 6 | CNN-3D | Total Params: 1,131,589  Training loss: 0.3209  Training accuracy: 0.8793  Validation loss: 0.5026  Validation accuracy: 0.8000 | - Both training and validation accuracy’s have dropped after adding dropouts and a new layer.  - There has not been any significant improvement in loss as well. | **Decision** : Remove dense layer with 256 and filter (3,3,3).  **Explanation** : Run with dropout of 0.25 to couple of layers. And (2,2,2) filters to make sure if dropouts are needed or not. |
| 7 | CNN-3D | Total Params: 895477  Training loss: 0.8865  Training accuracy: 0.6697  Validation loss: 0.9613  Validation accuracy: 0.6900 | - Validation accuracy is more than training accuracy so adding dropouts to layers is reducing number of parameters at unacceptable level. | **Decision** : Build on top of Model 3 or Model 5 by trying different number of neurons in Conv3D (8,32,64) and dense layers(remove 128N layer)  **Explanation** : Model 3 or Model 5 are the best results so far. Removing one Conv3D and dense layers to see the affects. |
| 8 | CNN-3D | Total Params: 985301  Training loss: loss: 0.3568  Training accuracy: 0.9442  Validation loss: 0.7019  Validation accuracy: 0.7100 | - Clearly we need those layers as validation loss has gone up to 0.7 and accuracy has reduce to 0.71 | **Decision** : Add Conv3D layers with different number of neurons (8,16,32,64) dense layer (128,32)  **Explanation** : Trying different neuron patterns to get best results |
| 9 | CNN-3D | Total Params: 3710605  Training loss: 0.2617  Training accuracy: 0.9261  Validation loss: 0.5138  Validation accuracy: 0.8100 | - Number of parameters is 37,10,605 which is a lot more than model 3 and 5  - But there is significant improvement in validation loss and accuracy | **Decision** : Add Conv3D layers with different number of neurons (16,32,64,128,256) dense layer (128,32)  **Explanation** : Trying different neuron patterns to get best results |
| 10 | CNN-3D | Total Params: 1961717  Training loss: 0.3390  Training accuracy: 0.9050  Validation loss: 0.4714  Validation accuracy: : 0.8800 | Epoch 29 giving the best results so far  - loss: 0.3166  - Training accuracy: 0.9080  - Validation loss: 0.4002  - Validation accuracy: 0.9000 | **Decision** : This model can be considered as best so far  **Explanation** : as both training and validation accuracy are high and losses are low. |
| CNN + RNN Architecture | | | | |
| 1 | CNN 2D with LSTM ( 64cells) | Total Params: 1,729,637  Training loss: 0.4255  Training accuracy: 0.8748  Validation loss: 0.8886  Validation accuracy: 0.6700 | - Looking at the accuracy’s and loss Convo3D models had better learning and accuracy. | **Decision** : Try with GRU as it is said to have better results with less computationally burden.  **Explanation** : Not going to experiment much as I already have better results |
| 2 | CNN 2D with GRU ( 128cells) | Total Params: 2,573,925  Training loss: 0.1371  Training accuracy: 0.9698  Validation loss: 0.8177  Validation accuracy: 0.7400 | - Looking at the accuracy’s are better than CNN\_LSTM but number of parameters are more. | **Decision** : Not going to experiment further although better learning and accuracy could be achieved with more experiments  **Explanation** : Total number of parameters is high, so Convo3D is giving comparable results with way less parameters. |
| Transfer Learning | | | | |
| 1 | Using Mobilenet + CNN with RNN (LSTM - 128 cells | Total Params: 3840453  Training loss: 0.0201  Training accuracy: 0.9985  Validation loss: 1.0242  Validation accuracy: 0.7000 | - Validation loss is very high. Model doesn't seem to learn with the architecture being used | **Decision** : Model used without training Mobilenet layers  **Explanation** : Parameters are more, let’s try with GRU |
| 2 | Using Mobilenet + CNN with RNN (GRU - 128 cells) | Total Params: 3446725  Training loss: 0.0310  Training accuracy: 0.9955  Validation loss: 1.1930  Validation accuracy: 0.6000 | - Validation loss is very high. Model doesn't seem to learn with the architecture being used | Going with Conv3D model |
|  |  |  |  |  |
| **Final model : Model 10** | **CNN-3D** | **Total Params: 1961717**  **Training loss: 0.3390**  **Training accuracy: 0.9050**  **Validation loss: 0.4714**  **Validation accuracy: 0.8800** | **Epoch 29 giving the best results**  **-Total Params: 1961717**  **- Training loss: 0.3166**  **- Training accuracy: 0.9080**  **- Validation loss: 0.4002**  **- Validation accuracy: 0.9000** | **Reason to select this model :For 1961717 of total parameters.**  **- Training and validation accuracy's are 0.90**  **- Training and validation losses are low. Between 0.3 to 0.4**  **As compared to other best model with**  **Total Params: 896,389**  **Training loss: 0.2692**  **Training accuracy: 0.9382**  **Validation loss: 0.6012**  **Validation accuracy: 0.8200** |