**Project**: Recognize hand gesture using state of the art neural networks.

**Team:**

1. Shadab
2. Vamshi
3. Sharun

**Problem Statement:**

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art smart televisions. You want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote.

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

**Analysis and reasoning for the models used:**

* We started with the base model which was simpler as mentioned model 1, in which, we got pretty good accuracy and validation scores.
* We further changed the dimensions of input image to 160x160. Using this some models gave higher accuracies as compared to initial model
* Same approach has been taken for Transfer learning + RNN (LSTM/GRU).  
  We used Mobile Net. It is efficient architecture with reasonable less parameters amongst transfer learning algorithms.
* We used batch size as 10 since it went good with memory handling. We focused on inputting 20 images, with experimenting 120x120 and 160x160.
* Data transformations/augmentation didn’t help in improving accuracies and we ignored for final model selection. It could be due to the information loss using augmentation.
* Final models are selected based on Train and Validation Accuracies (its tradeoff).

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | Conv 3D Model :  (16, 32, 64, 128 filters conv 3D layers)  + 256 dense layer  + 128 dense layer  + image size 120 by 120 | Training Acc: 92%  Validation Acc: 80% | The Validation accuracy is much lower than Train accuracy. It might fit lower on Test data as well.  Frames per video = 20  Dropout =0.20 |
| 2 | Conv 3D Model :  16, 32, 64, 128 filters conv 3D layers  + 256 dense nodes  + 128 dense nodes  + image size 160 by 160 | Training Acc: 98%  Validation Acc: 87% | The validation accuracy got improved as the image size was 160x160. The validation loss also got lowered.  Frames per video = 20  Dropout =0.20 |
| 3 | Conv 3D Model :  (16, 32, 64, 128 filters conv 3D layers + 256 dense layer + 128 dense layer) + Random data transformations on training data set for 120x120 image | Training Acc: 84%  Validation Acc: 78% | Addition of data augmentation resulted lower accuracies for both Train and Validation. Data Augmentation might be losing information.  Frames per video = 20  Dropout =0.20 |
| 4 | Conv 3D Model :  (32, 64, 128, 256) filters conv 3D layers + 128 dense layer + 128 dense layer + image size 120 by 120) | Training Acc: 92%  Validation Acc: 85% | Rescaled images to 120x120 with reducing layer neurons to 128 of 2 layers.  Frames per video = 20  Dropout =0.20 |
| 5 | Mobilenet  + GRU (128 cells)  + Dense (128 nodes)  + image size 120 by 120  + 20 images per video | Training Acc: 99%  Validation Acc: 95% | Using transfer learning, along with GRU, gave exceptional accuracies.  Dropout : 0.20  Parameters: 3693253 |
| 6 | Mobilenet  + GRU (128 cells)  + Dense (128 nodes)  + Dense (128 nodes)  + image size 120 by 120  + 20 images per video  + random transformations | Training Acc: 99%  Validation Acc: 93% | Added additional Dense layer of 128 nodes, along with random augmentation. The accuracy for validation dropped little as compared to initial model. |
| 7 | Mobilenet  + LSTM (128 cells)  + Dense (128 nodes)  + image size 120 by 120  + 20 images per video | Training Acc: 97%  Validation Acc: 94% | Using LSTM instead of GRU. It gave Good accuracies. |
| 8 | Mobilenet  + LSTM (128 cells)  + Dense (128 nodes)  + Dense (128 nodes)  + image size 120 by 120  + 20 images per video  + random transformations | Training Acc: 98%  Training loss: 0.04  Validation Acc: 97%  Validation loss: 0.12 | Added additional Dense layer of 128 nodes, along with random augmentation.  Surprisingly, it gave results over the bar.  Parameters: 3856965 |
|  |  |  |  |
| Final Model 1 | Conv 3D Model :  16, 32, 64, 128 filters  + 256 dense nodes  + 128 dense nodes  + image size 160 by 160 | Training Acc: 98%  Training loss: 0.08  Validation Acc: 87%  Validation loss: 0.36 | The Training accuracy and validation accuracy is best using CONV 3D model.  Dropout: 0.20  Paramters: 3604933 |
| Final Model 2 | Mobilenet  + GRU (128 cells)  + Dense (128 nodes)  + image size 120 by 120  + 20 images per video | Training Acc: 99%  Training loss: 0.01  Validation Acc: 95%  Validation loss: 0.20 | Using transfer learning, along with GRU, gave exceptional accuracies.  Dropout : 0.20  Parameters: 3693253 |

**Final Thoughts:**   
Out of the approach taken for Conv3D, we selected Conv 3D Model with 16, 32, 64, 128 filters, 256 dense nodes, 128 dense nodes, image size 160 by 160 feeding 20 images with 0.20 dropout.

And from CNN+RNN perspective, MobileNet (weight full retrained) with GRU (128 cells), Dense (128 nodes), image size 120 by 120, dropout as 0.20 and 20 images per video.  
We found out tradeoff between train and validation accuracies as well on parameters.