# GestureRecognition\_Assignment\_PrashantBhide

## Problem Statement:

Imagine you are working as a data scientist at a home electronics company that manufactures state of the art smart televisions. You want to 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. You need to create a deep learning-based model which can accurately capture different gestures of the user using the incoming video from the camera.

Each video is a sequence of 30 frames (or images). 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

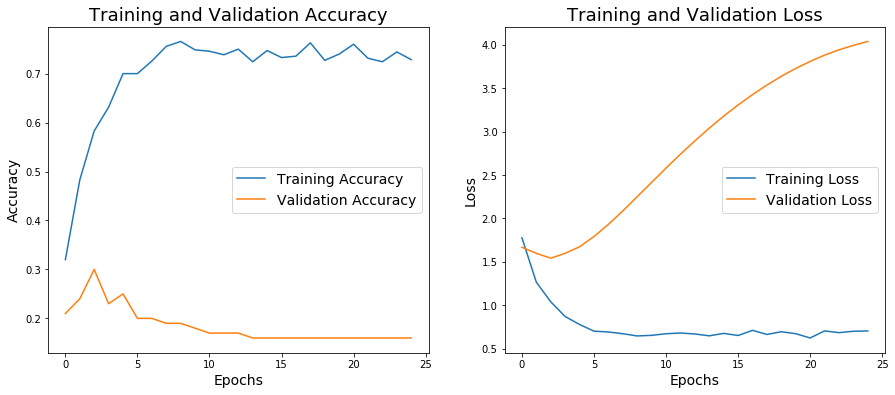
**Left swipe :** 'Jump' backwards 10 seconds

**Right swipe :** 'Jump' forward 10 seconds

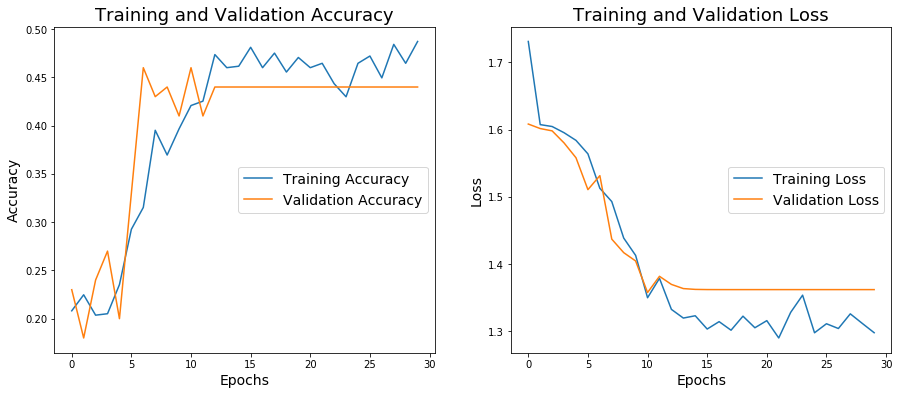
**Stop :** Pause the movie

## Deep-learning Model Building:

|  |  |  |  |
| --- | --- | --- | --- |
| **Exp #** | **Model** | **Result** | **Decision + Explanation** |
| 1 | Conv3D | Kernel crash! | Kernel crashes trying to use image size 100x100 and batch size 100 with full training data. Next to try - Train model with less data and Introduce BatchNormalization layer |
| 2 | Conv3D | Training Accuracy: 0.74  Validation Accuracy: 0.18 | Model is significantly over-fitting as seen in **Figure 1** below. Next to try - Reduce the image size and Introduce Dropout layers |
| 3 | Conv3D | Training Accuracy: 0.48  Validation Accuracy: 0.44 | Model has a balanced fit, but with low validation accuracy due to higher dropout values as seen in **Figure 2** below. Next to try – Reduce the Dropout values and Remove the BatchNormalization layer |
| 4 | Conv3D | Training Accuracy: 0.68  Validation Accuracy: 0.68 | Model has a balanced fit, and fair validation accuracy as seen in **Figure 3** below. Next to try – Increase the Conv3D filters and number of training epochs to 50 |
| 5 | Conv3D | Training Accuracy: 0.80  Validation Accuracy: 0.76 | Model has a balanced fit and decent validation accuracy at 76% reaching convergence around 25 epochs as seen in **Figure 4** below. Next to try – Remove the first Dropout layer and Increase the Dense layer neurons with number of epochs reduced to 30 |
| Final | Conv3D | Training Accuracy: 0.97  Validation Accuracy: 0.84 | The final model achieves a fairly decent validation accuracy peaking at 84%, with the training accuracy reaching 97% allowing for some over-fitting as seen in **Figure 5** below |



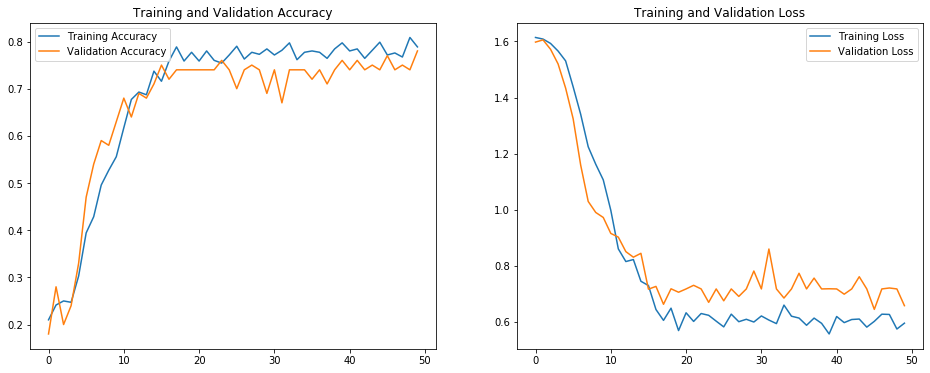
**Figure 1: Model is significantly over-fitting, with validation loss increasing across 25 epochs**



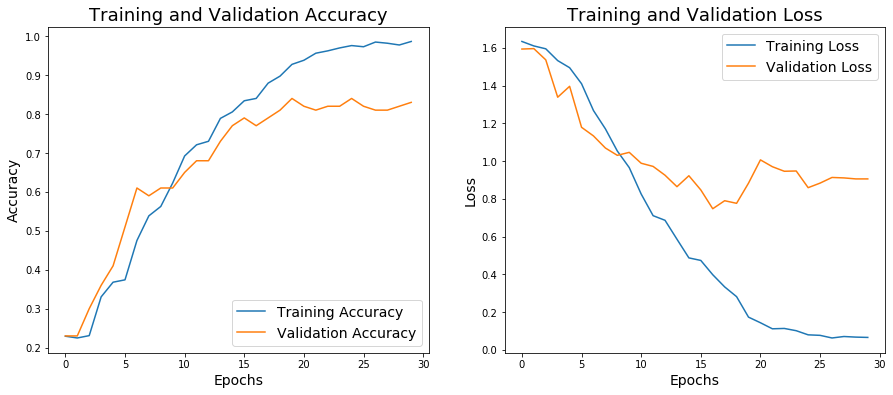
**Figure 2: Model has a balanced fit, but with low validation accuracy due to higher dropout values**



**Figure 3: Model has a balanced fit, with decreasing training and validation loss across 25 epochs**



**Figure 4: Model has a balanced fit, with decreasing training and validation loss across 50 epochs**



**Figure 5: Best model with some over-fitting, but with the highest validation accuracy across 30 epochs**

## Final Conv3D model summary:

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d (Conv3D) (None, 20, 40, 40, 32) 2624

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_1 (Conv3D) (None, 20, 40, 40, 32) 27680

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d (MaxPooling3D) (None, 10, 20, 20, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 20, 20, 64) 55360

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 10, 20, 20, 64) 110656

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 10, 10, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 5, 10, 10, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 32000) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 128) 4096128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 325

=================================================================

Total params: 4,301,029

Trainable params: 4,301,029

Non-trainable params: 0

## Saving the final Conv3D model:

The model architecture and weights are saved at the end of each training epoch as part of the ModelCheckpoint callbacks. The saved model filename corresponding to the best model (validation accuracy = 0.84) is **[model-00025-0.07910-0.97587-0.85918-0.84000.h5]** and will be included with the project submission zip file.