We have experimented with the following 3 architecture in CONV3D and 1 architecture in CONV2D + RNN

**Conv3d - Architecture 1**

CONV3D architecture with two convolutional layers (each followed by a batch normalization and maxpooling). Convolutional layers are followed by 2 dense layers. Also used dropout after the dense layers. (kernel size (2,2,2))

**Conv3d - Architecture 2**

CONV3D architecture with five convolutional layers (each followed by a batch normalization and maxpooling). Convolutional layers are followed by 2 dense layers. Also used dropout after the dense layers. (kernel size (3,3,3))

**Conv3d - Architecture 3**

CONV3D architecture with two convolutional layers. Each of the layer has different feature size (each followed by a batch normalization and maxpooling). Convolutional layers are followed by 3 dense layers. Also used dropout after the dense layers and the convolutional layers. Different kernel sizes were used

**Conv2d + RNN**

Conv2D with 4 CONV2D layers which is used as with in a Timedistributed (each followed by a batch normalization and maxpooling). Dropouts are also added after the pair of convolutional layers and after each dense layers

In each of the three network architecture, we had experimented both in the image processing part and also in the network layers part to arrive at the final architecture

Majority of the experimentation process involved trying out cropping, resizing and affine transformation and a combination of these. Final model contains architecture 3 and the image processing of resizing and affine transformation

Dataset consisted of 663 videos that corresponds to 5 different classes

* 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

The dataset consisted of few videos containing the frame size of 360x360 and few other vides of frame size 120x160

As suggested we started off by making all the frame size to a standard size

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D and Architecture 1** | **Training accuracy was around 87% and was increasing with epoch size but the training accuracy was very low around 17%. Also the parameter count was very huge around 52 million and the size was around 600 Mb** | **Since the image size need to be same all the images were resized to a size of 160x160.**  **Selected every 3 frames out of 30 assuming that the adjacent frames would extremely similar. Also converted the frames to grayscale assuming it would reduce the parameter count** |
| **2** | **Conv3D and Architecture 1** | **Number of parameters got reduced to 19 million with a drop in validation accuracy (training accuracy - ~82% and validation accuracy ~ 12 %)** | **RGB images were used since using grayscale didn’t reduce the parameter size much. Size of the image was reduced to 80x120 and instead of resizing both the image were cropped so that the pixel resolution is not affected. Also went with cropping since majority of the hand gestures is within this range.** |
| **3** | **Conv3D and Architecture 1** | **Training and validation accuracy changed to 83 and 20%** | **Since the accuracy decreased, increased the frame size to 120x120** |
| **4** | **Conv3D and Architecture 1** | **This didn’t work out as expected, the training accuracy dropped to ~75% and validation accuracy to 18%** | **After analyzing the images most of information in the cropped images will be captured in the frames 10 to 20 and hence used the those frames with a skip of 2** |

For the above configuration of network the parameter count decreased only with the reduction in either the frame size or with the decrease in the number of frames used. Hence thought of using a different architecture for the network with a different kernel size and the number of layers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **5** | **Conv3D and Architecture 2** | **Though the train accuracy increased well to the 94% the validation accuracy remained in the range of 20% and the parameter count (13 million)** | **Used a frame size of 120x120 after cropping with the 10 frames between 10 to 20. This is was tried since choosing low frame number impacted the validation accuracy drastically.** |
| **6** | **Conv3D and Architecture 2** | **This had a better performance in terms of the validation accuracy with validation accuracy of 26% and comparable training accuracy of 81%.**  **Even after changing the regularizer value we got a max validation accuracy of 41%** | **Since the training accuracy increased to range of 94 inferred that the model is learning well for the small data. The model already had dropout layers in it. Now along with it added and l2 regularizer of value 0.01** |

Since we were not able to increase the validation accuracy even after adding regularizer beyond 42% we tried another architecture

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **7** | **Conv3D and Architecture 3** | **Model didn’t have much parameters even for a 160x160 image even with 10 frames. For the small amount of data provided the both the training and validation accuracy gradually increased to 80% and 30%** | **Started with resizing the image to a size of 160x160 with 10 frames with a skip of 3. All the three channels were used** |
| **8** | **Conv3D and Architecture 3** | **Model parameter count further decreased to ~800k and the difference between training and validation accuracy decreased further to 70% and 52%** | **Since the past models have shown an improvement in accuracy with increase in the number of frames and the number of parameters is not that large, currently using all the 30 frames.**  **Also resizing the images to 120x120 since earlier that frame size also shown comparable accuracy with 160x160** |
| **9** | **Conv3D and Architecture 3** | **Validation accuracy decreased to 40% and the training accuracy decreased to 65%** | **Applied the same experiment (exp no 8) with cropping the image to 120x120 size instead of resizing it. Hoping that the performance will be improved since the actual pixel is unchanged. But cropping didn’t give much improvement to the performance** |
| **10** | **Conv3D and Architecture 3** | **Validation accuracy decreased to 80% and the training accuracy decreased to 77%** | **Since resizing provided better results going ahead with resizing. Now as a final improvement adding augmented data with some affine transformation. This is done since most of the videos does not have a fixed camera angle and the same can be expected to the validation and test videos as well**  **Keeping all the other parameters as same**  **Image size – 120x120**  **Number of frames used -30**  **Channels - 3** |
| **11** | **CONV2D + RNN** | **Got a validation accuracy of 56% and training accuracy of 94%** | **Used the image size of 120x120 using all the 3 channels and a batch size of 8 we got a good training accuracy but the validation accuracy didn’t increase beyond 56** |