**Topic: Experimenting with 3d Convulsions using Gesture Recognition**

**Group: Manoj Kuruvella, Ravneet Singh**

Source Data: Gesture Recognition videos Data Downloaded

Experiment: Using the Gesture Data, perform 3DConv and train the model so that it classifies the Gesture into one of the 5 classes.

Approach: We have started with adopting the 3D conv architecture and have used multiple layers of 3D Conv, Batch Normalization , Flattening and the Dense layers using different variation of drop outs. Also, we used different SGT and Adam categorical cross entropy. We used Learning rate (lr) and regularization (l2) in the different experiments.

* Each video is of 30 frames and there are around 663 sequences in total. We have experimented with full set of frames, alternative frames, selective frames.
* We also doubled the input data by taking each sequence and flipping the image and assigning it to the reverse class for Swipe\_left and swipe\_right whereas other labels do not differ.
* And also, we did re-sizing of the images to have all the input of fixed size images.
* We applied Normalization techniques by min-max scaling.
* We also experimented with various batch sizes and regularization factors.

The base Architecture is as :

Input Layer

Conv3D 8 (kernel\_size=(3,3,3), strides=(1,1,1), activation='relu',)

MaxPooling3D(pool\_size=(2,2,2)

BatchNormalization()

Conv3D 16 (kernel\_size=(3,3,3), strides=(1,1,1), activation='relu',)

MaxPooling3D(pool\_size=(2,2,2)

BatchNormalization()

Conv3D 32((kernel\_size=(3,3,3), strides=(1,1,1), activation='relu',)

MaxPooling3D(pool\_size=(2,2,2)

BatchNormalization()

Conv3D 64 (kernel\_size=(3,3,3), strides=(1,1,1), activation='relu',)

MaxPooling3D(pool\_size=(2,2,2)

BatchNormalization()Flatten

Dense(16)

Dropout(.25)

Dense(32)

Dropout(.25)

Softmax

Using the above Architecture , we modified with different Experiments as the Accuracy differs to each variation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Parameters** | **Result** | **Decision + Explanation** |
| **1** | **Frames : 10**  **Batch size : 64**  **Initial Lr : .1**  **Factor : .1** | **Training Accuracy : .40** | **The Training Accuracy is so less and we wanted to increase more frames.** |
| **2** | **Frames : 15**  **Batch size : 64**  **Initial Lr : .1**  **Factor : .1** | **Training Accuracy : .2022** | **The Training Accuracy is still less and we planned to add more layers and added a 64 feature .** |
| **3** | **Frames : 15**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .4922** | **The training Accuracy is still less than 50% and we planned to increase more parameters. We increased the Dense layer from 16 to 64 and also changed Lr to ‘Adam’** |
| **4** | **Frames : 15**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .72**  **Validation Accuracy : .62** | **The training Accuracy is still good at 72% an increase from less . This model we made as a base and tweaked the parameters. Further modified the Dense to 128 and 64 layers.** |
| **5** | **Frames : 15**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .93**  **Validation Accuracy : .62** | **The training Accuracy has improved a lot but the validation accuracy is still at 62% . we planned to increase the input data from 15 to all the frames and added a maxpool layer after input.** |
| **6** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** .8883  **Validation Accuracy : .66** | **The validation accuracy has increased by 5% than earlier. But not sufficient. So we made the Dense layers to 256 and 128. Also increased the Dropouts to .5** |
| **7** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 7933  **Validation Accuracy : .7300** | **The validation accuracy has increased by 5% than earlier. But not sufficient. So we made the Dense layers to 256 and 128. Also increased the Dropouts to .5. This has been the good model at balancing the Train and Validation Scores. We reduced the Dropouts as the training has been compromised** |
| **8** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 85  **Validation Accuracy : .61** | **The validation accuracy has gone down. We planned to add a new Conv3D 128 layer and Maxpooling.** |
| **9** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 93  **Validation Accuracy : .74** | **The validation accuracy has increased and training accuracy also increased. This can be one final model but Training accuracy is little close to overfit. Tried to control dropouts and removed the dropouts after FC** |
| **10** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 834  **Validation Accuracy : .72** | **Training overfit is controlled but not a considerable improvement in Validation. Removed one of the Dense layer and dropout layer.** |
| **11** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 99  **Validation Accuracy : .72** | **Training overfit.**  **Modifed the regularizer to l2 and added Dense layers of 256 , 128** |
| **12** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 90  **Validation Accuracy : .50** | **The validation has hit very high. We understand that Training is overfitting with too high number of parameters.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **13** | **Frames : 30**  **Batch size : 64**  **Initial Lr : default**  **Factor : .1** | **Training Accuracy : .** 99  **Validation Accuracy : .72** | **We doubled the data by flipping each image of a sequence and assigning it as reverse class. But there is not much additional advantage we observed.** |

Out of all the Experiments, Our final model is the Experiment 7.

Our finalized Model looks like this :

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

Layer (type) Output Shape Param #

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

conv3d\_5 (Conv3D) (None, 30, 120, 120, 8) 656

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

max\_pooling3d\_5 (MaxPooling3 (None, 15, 60, 60, 8) 0

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

conv3d\_6 (Conv3D) (None, 15, 60, 60, 16) 3472

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

max\_pooling3d\_6 (MaxPooling3 (None, 7, 30, 30, 16) 0

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

batch\_normalization\_4 (Batch (None, 7, 30, 30, 16) 64

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

dropout\_4 (Dropout) (None, 7, 30, 30, 16) 0

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

conv3d\_7 (Conv3D) (None, 7, 30, 30, 32) 13856

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

max\_pooling3d\_7 (MaxPooling3 (None, 3, 15, 15, 32) 0

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

batch\_normalization\_5 (Batch (None, 3, 15, 15, 32) 128

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

conv3d\_8 (Conv3D) (None, 3, 15, 15, 64) 55360

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

max\_pooling3d\_8 (MaxPooling3 (None, 1, 7, 7, 64) 0

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

batch\_normalization\_6 (Batch (None, 1, 7, 7, 64) 256

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

flatten\_2 (Flatten) (None, 3136) 0

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

dense\_4 (Dense) (None, 256) 803072

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

dropout\_5 (Dropout) (None, 256) 0

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

dense\_5 (Dense) (None, 128) 32896

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

dropout\_6 (Dropout) (None, 128) 0

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

dense\_6 (Dense) (None, 5) 645

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

Total params: 910,405

Trainable params: 910,181

Non-trainable params: 224

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

None