**SUMMARY-**

The task was to identify gestures in video clips. We started with 2D RNN and couldn’t get a good accuracy model. Different configurations were tried and parameters such as batch size, number of images, dropout value were tried and the best validation accuracy obtained was 20 %. We then moved to Conv 3D.

Same exercise was repeated for varying models. Modifying architecture, dropout values were reduced to overcome overfitting, more number of images were used to predict the gesture. The validation accuracy was not going above 40%. There models with 19 million trainable parameters but still we were not able to arrive at a good model which could perform satisfactorily on validation set.

We then moved to simple model architecture. We started with 4 million parameters and model was performing better than previous iterations. Below table – experiment number 40 to last gives us details of the iterations and reasoning behind it.

**Final model –**

ImageSize-120\*120  
Num\_images-22  
Batch\_Size-20  
filter(3,3,3)  
dense\_neurons-128  
dropout- 0.5  
epochs 15

***Trainable parameters(~1M)- 1,455,749***

***Training accuracy- 98%***

***Validation-accuracy- 93%***

***The final model has 1.5 million parameters and more than 90% accuracy on training and validation set.***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Model** | **Name** | **Number of trainable parameters** | **Train accuracy** | **Val accuracy** | **Reasoning** | **Model Details/Changes** |
| 1 | 2D RNN | conv\_2drnn\_1 | 2,053,125 | 0.8 | 0.2 | NA |  |
| 2 | 2D RNN | conv\_2drnn\_2 | 1,004,549 | 0.72 | 0.2 | Reduce number of trainable parameters | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and two layers after flatening with dropouts. Optimizer - adam ImageSize-100\*100 Num\_images-15 Batch\_Size-51 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 3 | 2D RNN | conv\_2drnn\_3 | 1,004,549 | 0.74 | 0.24 | Increasing image size since accuracy of previous model was reduced | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and two layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-51 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 4 | 2D RNN | conv\_2drnn\_4 | 1,004,549 | 0.75 | 0.19 | changing batch size to improve model training | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and two layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 5 | 2D RNN | conv\_2drnn\_5 | 1,004,549 | 0.6621 | 0.19 | Using more number of images to make a prediction to improve accuracy | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and two layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-20 Batch\_Size-39 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 6 | GRU | conv\_2drnn2\_1 | 853,125 | 0.87 | 0.14 | Using GRU to improve model accuracy | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and a GRU layer after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-20 Batch\_Size-39 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 7 | RNN | conv\_2drnn3\_1 | 549,701 | 0.81 | 0.2 | Since GRU is not giving better accuracy we are trying simple RNN | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and a SimpleRNN layer after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-20 Batch\_Size-39 filter(3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
|  | MOVING onto 3DConv networks since previous combinations not giving good validation accuracy (not more than 0.2) | | | | | |  |
| 8 | Conv3D | Model3DConv\_1 | 997,157 | 0.83 | 0.17 |  | Trying architecture with 5 layers (16,32,64,128,256 ) with relu and relu as the activation function and a SimpleRNN layer after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 9 | Conv3D | conv\_3dconv2\_1 | 1,931,749 | 0.85 | 0.2 | Increasing dense neurons to increase model complexity | Trying architecture with 3 layers (16,32,64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-128 dropout- 0.25 epochs 5 |
| 10 | Conv3D | conv\_3dconv2\_2 | 3,774,949 | 0.94 | 0.23 | Increasing number of images used to make prediction to 18 from 15 | Trying architecture with 3 layers (16,32,64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-18 Batch\_Size-39 filter(3,3,3) dense\_neurons-128 dropout- 0.25 epochs 5 |
| 11 | Conv3D | conv\_3dconv3\_1 | 3,825,509 | 0.95 | 0.2 | updating dense neurons to 256 and increase model complexity since previous step of incresing dense neurons to 128 gave a better validation accuracy | Trying architecture with 3 layers (16,32,64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-18 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.25 epochs 5 |
| 12 | Conv3D | conv\_3dconv4\_1 | 3,964,037 | 0.97 | 0.2 | Adding more number of layers to make model more complex | Trying architecture with 5 layers (16, 32, 32, 64, 64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.25 epochs 5 |
| 13 | Conv3D | conv\_3dconv4\_2 | 3,964,037 | 0.82 | 0.18 | Increasing dropout to reduce overfitting | Trying architecture with 5 layers (16, 32, 32, 64, 64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 14 | Conv3D | conv\_3dconv5\_1 | 947,605 | 0.8 | 0.29 | Changing filter size to reduce complexity | Trying architecture with 3 layers (16,32,64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(2,2,2) dense\_neurons-256 dropout- 0.25 epochs 5 |
| 15 | Conv3D | conv\_3dconv6\_1 | 997,157 | 0.51 | 0.18 | Adding dropout layers to avoid overfitting | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.25 epochs 5 |
| 16 | Conv3D | conv\_3dconv6\_2 | 997,157 | 0.35 | 0.22 |  | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.5 epochs 5 |
| 17 | Conv3D | conv\_3dconv6\_3 | 997,157 | 0.32 | 0.12 | increasing batch size with same dropout value | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-51 filter(3,3,3) dense\_neurons-64 dropout- 0.5 epochs 5 |
| 18 | Conv3D | conv\_3dconv6\_4 | 997,157 | 0.36 | 0.17 | Model accuracy has increased compared to earlier model but not to an extent we wanted | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.5 epochs 15 |
| 19 | Conv3D | conv\_3dconv6\_5 | 997,157 | 0.33 | 0.14 | training for high value of epochs of around 25 | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.5 epochs 25 |
| 20 | Conv3D | conv\_3dconv6\_6 | 997,157 | 0.37 | 0.21 | training for dropout value of 0.4  earlier dropout was 0.5, reducing it to make model learn more complex features | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-64 dropout- 0.4 epochs 5 |
| 21 | Conv3D | conv\_3dconv6\_7 | 3,825,509 | 0.51 | 0.24 | Training for dense neurons 256  Model parameters have increased almost 4 times. At the same time accuracy has also increased. | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.4 epochs 5 |
| 22 | Conv3D | conv\_3dconv6\_8 | 3,825,509 | 0.42 | 0.17 | increasing dropout to 0.5 | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 23 | Conv3D | conv\_3dconv6\_9 | 3,825,509 | 0.64 | 0.17 | Reducing value of dropout to 0.3  Reducing underfitting | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-120\*120 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.3 epochs 5 |
| 24 | Conv3D | conv\_3dconv6\_10 | 6,692,709 | 0.45 | 0.19 | Increasing input image size to 160\*160 to capture more details | Trying architecture with 3 layers (16,32,64) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-15 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 25 | Conv3D | conv\_3dconv7\_1 | 3,637,477 | 0.4 | 0.15 | Making model more complex by adding more parameters | Trying architecture with 4 layers (16,32,64,128) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 26 | Conv3D | conv\_3dconv8\_1 | 3,637,477 | 0.97 | 0.2 | Reducing dropout value so that model can learn features | Trying architecture with 4 layers (16,32,64,128) and dropouts with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-39 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 5 |
| 27 | Conv3D | conv\_3dconv8\_2 | 3,637,477 | 0.78 | 0.2 | Updating batch size and increasing dropout to 0.5 | Trying architecture with 4 layers (16,32,64,128) and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 28 | Conv3D | conv\_3dconv9\_1 | 3,637,477 | 0.41 | 0.21 | Adding dropout to above network to reduce overfitting | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 29 | Conv3D | conv\_3dconv9\_2 | 3,637,477 | 0.43 | 0.13 | after updating generator by changing the normalisation technique | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 30 | Conv3D | conv\_3dconv9\_3 | 3,637,477 | 0.42 | 0.16 | Using 30 images to make a prediction | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-30 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 31 | Conv3D | conv\_3dconv10\_1 | 3,636,997 | 0.49 | 0.27 | Removing Batch normalisation  and using a dropout value of 0.2 | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 5 |
| 32 | Conv3D | conv\_3dconv10\_2 | 3,636,997 | 0.77 | 0.38 | Since above model is giving good result, we will train it for more epochs | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 15 |
| 33 | Conv3D | conv\_3dconv9\_3 | 1,540,325 | 0.78 | 0.38 | Using model 9 with changed parameter  Training for 15 epochs | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-160\*160 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 15 |
| 34 | Conv3D | conv\_3dconv9\_4 | 1,540,325 | 0.42 | 0.15 | Using model 9 with reduced image size  To reduce overfitting | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-100\*100 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 15 |
| 35 | Conv3D | conv\_3dconv9\_5 | 1,540,325 | 0.68 | 0.27 | Using model 9 with low dropout and high epoch value | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-100\*100 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 30 |
| 36 | Conv3D | conv\_3dconv11\_1 | 1,540,325 | 0.66 | 0.23 | Updating optimizer to rmsprop in Model9 | Trying architecture with 4 layers (16,32,64,128) and dropouts and relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - rmsprop ImageSize-100\*100 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 5 |
| 37 | Conv3D | conv\_3dconv12\_1 | 2,636,933 | 0.69 | 0.13 | Adding dropout in model4 as it had highest train accuracy | Trying architecture with 5 layers (16, 32, 32, 64, 64) with relu and relu as the activation function and 2 layers after flatening with dropouts. Optimizer - adam ImageSize-100\*100 Num\_images-18 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.2 epochs 5 |
| 38 | Conv3D | conv\_3dconv13\_1 | 1,541,829 | 0.63 | 0.14 | Adding l1 regularisation | Trying architecture with 5 layers (16, 32, 64, 128) with elu and elu as the activation function and with dropouts and l1 regularisation. Optimizer - adam ImageSize-100\*100 Num\_images-16 Batch\_Size-20 filter(3,3,3) dense\_neurons-256 dropout- 0.5 epochs 5 |
| 39 | Conv3D | conv\_3dconv14\_1 | 19,158,213 | 0.99 | 0.48 | Adding l2 regularisation | Trying architecture with 5 layers (16, 32, 64, 128) with elu and elu as the activation function and with dropouts and l2 regularisation. Optimizer - adam ImageSize-100\*100 Num\_images-18 Batch\_Size-30 filter(3,3,3) dense\_neurons-512 dropout- 0.5 epochs 30 |
| 40 | Conv3D | conv\_3dconv14\_2 | 19,158,213 | 0.89 | 0.21 | #Using model 9 with changed parameters #with new generator with new cropping logic | Trying architecture with 5 layers (16, 32, 64, 128) with elu and elu as the activation function and with dropouts and l2 regularisation. Optimizer - adam ImageSize-100\*100 Num\_images-18 Batch\_Size-30 filter(3,3,3) dense\_neurons-512 dropout- 0.5 epochs 30 |
| 41 | Conv3D | conv\_3dconv15\_1 | 4,806,277 | 0.92 | 0.7 | Complex models are not giving good validation accuracy. Even increasing parameters, chaninging dropout size is not helping. Trying a simple model once. | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- not included epochs 5 |
| 42 | Conv3D | conv\_3dconv15\_2 | 4,806,277 | 0.99 | 0.76 | Simple model is giving good result and hence training for more epochs | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- not included epochs 15 |
| 43 | Conv3D | conv\_3dconv16\_1 | 4,806,277 | 0.89 | 0.79 | Adding dropout to Model 15 | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.2 epochs 5 |
| 44 | Conv3D | conv\_3dconv16\_2 | 4,806,277 | 0.99 | 0.84 | running for higher number of epochs with 0.2 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.2 epochs 15 |
| 45 | Conv3D | conv\_3dconv16\_3 | 4,806,277 | 0.99 | 0.84 | running for higher number of epochs with 0.3 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.3 epochs 30 |
| 46 | Conv3D | conv\_3dconv16\_4 | 4,806,277 | 0.98 | 0.83 | Increasing dropout to 0.5 | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.5 epochs 15 |
| 47 | Conv3D | conv\_3dconv16\_5 | 4,806,277 | 0.96 | 0.85 | running for right number of epochs with 0.5 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-30 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.5 epochs 12 |
| 48 | Conv3D | conv\_3dconv16\_6 | 2,438,789 | 0.98 | 0.86 | running for less number of images whihc do no give negative dimension error with 0.5 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-22 Batch\_Size-16 filter(3,3,3) dense\_neurons-128 dropout- 0.5 epochs 12 |
| 49 | Conv3D | conv\_3dconv16\_7 | 2,438,789 | 0.94 | 0.79 | running for less number of images which do not give negative dimension error,higher batch size , with 0.5 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-150\*150 Num\_images-22 Batch\_Size-39 filter(3,3,3) dense\_neurons-128 dropout- 0.5 epochs 12 |
| 50 | Conv3D | conv\_3dconv16\_8 | 1,455,749 | 0.94 | 0.91 | running for smaller image size and batch size (as higher batch size didn't prove good) , with 0.5 droput value | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-120\*120 Num\_images-22 Batch\_Size-20 filter (3,3,3) dense\_neurons-128 dropout- 0.5 epochs 12 |
| 51  Final Model | Conv3D | conv\_3dconv16\_9 | 1,455,749 | 0.98 | 0.93 | running for the model above with higher epochs for even better accuracy | Trying architecture with 5 layers (16, 32, 64) with relu and relu as the activation function. Optimizer - adam ImageSize-120\*120 Num\_images-22 Batch\_Size-20 filter(3,3,3) dense\_neurons-128 dropout- 0.5 epochs 15 |

**Tried to see if the accuracy could be improved further but no luck and hence considered the above Model 51 as the final one.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 52 | Conv3D | conv\_3dconv17\_1 | 15 | 1,455,749 | 0.98 | 0.81 |  | Changed the optimizer to ‘rmsprop’ keeping all the other parameters same as in Model 50. |
| 53 | Conv3D | conv\_3dconv18\_1 | 15 | 1,455,749 | 0.99 | 0.85 |  | Changed the activation function to ‘elu’ keeping all the other parameters same as in Model 50. |

**Conclusion-**

As there was no further improvement in the val/train model accuracy and no decrease in the val/train loss after conv\_3dconv16\_9 (Model 51 in above table) hence considering it the best and final model with below parameters:

**Model Architecture - Model3DConv\_16**

**Model Name- conv\_3dconv16\_9**

**Trainable parameters(~1.5M)- 1,455,749**

**Epochs- 15, Image size - 120\*120, Number of images per video – 22, Batch Size- 20, Filter Size – (3,3,3)**

**Training accuracy- 98% , Validation-accuracy- 93%**

**Final Model Architecture-**

class Model3DConv\_16(ModelGenerator):

def define\_model(self,filter=(3,3,3),dense\_neurons=64,dropout=0.25):

input\_shape\_model = (self.num\_images,self.image\_width,self.image\_height,3)

model = Sequential()

model.add(Conv3D(16, filter,input\_shape=input\_shape\_model))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Conv3D(32, filter))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Conv3D(64, filter))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Flatten())

model.add(Dense(dense\_neurons,activation='relu'))

model.add(Dropout(dropout))

model.add(Dense(5,activation='softmax'))

optimiser = 'adam'

model.compile(optimizer=optimiser, loss='categorical\_crossentropy', metrics=['categorical\_accuracy'])

#K.set\_value(model.optimizer.learning\_rate,0.0002)

return model

# Model Parameters-

conv\_3dconv16\_9=Model3DConv\_16()

conv\_3dconv16\_9.image\_size(image\_height=120,image\_width=120)

conv\_3dconv16\_9.other\_parameters(num\_images=22,epochs=15,batch\_size=20)

conv\_3dconv16\_9\_model=conv\_3dconv16\_9.define\_model(filter=(3,3,3),dense\_neurons=128,dropout=0.5)

conv\_3dconv16\_9\_model.summary()

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

Layer (type) Output Shape Param #

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

conv3d\_24 (Conv3D) (None, 20, 118, 118, 16) 1312

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

activation\_24 (Activation) (None, 20, 118, 118, 16) 0

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

max\_pooling3d\_24 (MaxPooling (None, 10, 59, 59, 16) 0

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

conv3d\_25 (Conv3D) (None, 8, 57, 57, 32) 13856

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

activation\_25 (Activation) (None, 8, 57, 57, 32) 0

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

max\_pooling3d\_25 (MaxPooling (None, 4, 28, 28, 32) 0

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

conv3d\_26 (Conv3D) (None, 2, 26, 26, 64) 55360

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

activation\_26 (Activation) (None, 2, 26, 26, 64) 0

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

max\_pooling3d\_26 (MaxPooling (None, 1, 13, 13, 64) 0

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

flatten\_5 (Flatten) (None, 10816) 0

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

dense\_10 (Dense) (None, 128) 1384576

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

dropout\_5 (Dropout) (None, 128) 0

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

dense\_11 (Dense) (None, 5) 645

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

Total params: 1,455,749

Trainable params: 1,455,749

Non-trainable params: 0

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

# Model Run-

print("Total Params:", conv\_3dconv16\_9\_model.count\_params())

history\_conv\_3dconv16\_9 = conv\_3dconv16\_9.train\_model(train\_path,val\_path,train\_doc,val\_doc,conv\_3dconv16\_9\_model)

# Result-

Epoch 00015: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.

34/34 [==============================] - 109s 3s/step - loss: 0.0818 - categorical\_accuracy: 0.9759 - val\_loss: 0.2575 - val\_categorical\_accuracy: 0.9300

# Model Plot-

