**Problem statement:**

In this problem, we need to design a deep learning neural network which will detect user’s hand gestures. There are 5 gestures which need to be identified.

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

The training data consists of a few hundred videos categorised into one of the five classes. Each video (typically 2-3 seconds long) is divided into a **sequence of 30 frames(images)**.

About the data:

|  |  |  |
| --- | --- | --- |
| Training | Class | Number of Videos (folders) |
| 0 | 136 |
| 1 | 137 |
| 2 | 130 |
| 3 | 137 |
| 4 | 123 |

|  |  |  |
| --- | --- | --- |
| Validation | CLASS | NUMBER OF VIDEOS (FOLDERS) |
| 0 | 18 |
| 1 | 23 |
| 2 | 22 |
| 3 | 21 |
| 4 | 16 |

We can see that the data is evenly distributed among the classes. The images in the dataset have two dimensions: **160X120 and 360X360**

We will resize the images to a common size before feeding to the model.

**We have croped all the 160X120 images to 120X120 in generator.**

**Experiments**

**Graph Colors**

Training Loss Validation Loss

Training Accuracy Validation Accuracy

**CNN (CONV-3D) Model**

|  |  |  |  |
| --- | --- | --- | --- |
| E# | Model | Performance | Decision & explanation |
| 1 | CNN model  CONV 3D: (8,16,32,64)  Dense layer: (64) | Training accuracy:21%  Validation accuracy: 23%. | **Explanation:**  The model did not run for 20 epochs as it was stopped using early stopping because there was no improvement in **validation loss (val\_loss).**  There was no significant change with epochs.  **Decision:**  So, we decided to try different optimizer. We tried ADAM in the next experiment. |
| 2 | CNN model  CONV 3D: (8,16,32,64)  Dense layer: (64) | Training accuracy:100%  Validation accuracy: 70%. | **Explanation:**  As we can see from the graph that the loss decreased significantly as the number of epochs increased. However, after around 6 epochs, the **training accuracy** started to increase and reached 1 signifying that the model was over fitting. The **validation accuracy** started to decrease.  **Decision:**  Thus, we decided to introduce batch normalization and dropout for our next experiment. |
| 3 | CNN model  CONV 3D: (8,16,32,64)  Dense layer: (64)  BN and DO | Training accuracy:45%  Validation accuracy: 55%. | **Explanation:**  As can we see from the graph, the performance (**validation accuracy**) of the model decreased after introducing batch normalization and dropout.  **Decision:**  So, we decided to increase the complexity of the model by increasing the number of parameters. We increased the number of dense layers.  We also increased batch size from 10 to 20. |
| 4 | CNN model  CONV 3D: (8,16,32,64)  Dense layer: (128,64)  BN and DO | Training accuracy:76%  Validation accuracy: 69%. | **Explanation:**  As we can see that the model is performing good (**validation accuracy**) with such less parameters.  **Decision:**  However, to improve further, we decided to introduce more regularization by increasing the dropouts on the conv and dense layers. |
| 5 | CONV 3D: (8,16,32,64)  Dense layer: (128,64)  BN and DO | Training accuracy:52%  Validation accuracy: 67%. | **Explanation:**  We can see that the performance of the model reduced. The **validation accuracy** is higher than **training accuracy** which implies that the model is underfitting.  **Decision:**  We decided to increase parameters by increasing the fully connected layer size.  We also introduced **l2 regularization on the second last** fully connected layer.  We also increased the epoch from **20 to 30** as we saw improvement in the loss when the epochs were increasing.  Learning rate: 0.002 |
| 6 | CONV 3D: (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization  **Learning rate: 0.002** | **Training accuracy: 84%**  **Validation accuracy: 81%.** | **Explanation:**  We can see that this model is giving very good **validation accuracy of 81%** with relatively small number of parameters **(910,437). This modal is good candidate for the final model.**  ***The size of the model is also small: 11MB***  **Decision:**  While seeing the logs, we found that the learning rate was being reduced by **ReduceLROnPlateau** to increase the learning. We decided to lower the learning rate to **0.0002.** |
| 7 | CONV 3D: (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization | Training accuracy:44%  Validation accuracy: 43%. | **Explanation:**  We can see that the model degraded severely when we reduced learning rate from **0.002 to 0.0002.** The validation accuracy and training accuracy both dropped as the model did not properly learn.  We also tried with **0.001, however, 0.002 gave better result.**  **Decision:**  We decided to change the learning rate back to **0.002.**  However, we decided to change the number of frames to check if we get any big degradation of the model. |
| 8 | CONV 3D: (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization  **With 24 equally spaced frames from 0** | Training accuracy:74%  Validation accuracy: 70% | **Explanation:**  We can see that the **validation accuracy and training accuracy** are good. However, reducing the number of frames resulted in less training time.  **Decision:**  As our aim is to find a model with best accuracy we decided to continue with ***Model#6*** for further experiments.  We started with reducing **kernel size from (3,3,3) to (2,2,2)** |
| 9 | CONV 3D: (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization  Kernel Size: (2,2,2) | Training accuracy: 50%  Validation accuracy: 40% | **Explanation:**  We can see that the model did not do well. **Both training accuracy and validation accuracy dropped.** The model stopped learning and was stopped well before 30 epochs by early stopping callback.  **Decision:**  As our aim is to find a model with best accuracy we decided to continue with ***Model#6*** for further experiments.  For the next experiment we decided to change image size from **(120X120) to (100X100)** |
| 10 | (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization  **Image -Size**  **(100X100)** | Training accuracy: 62%  Validation accuracy: 51% | **Explanation:**  We can see that the model did not do well. **Both training accuracy and validation accuracy dropped.** The model did not learn properly.  **Decision:**  As our aim is to find a model with best accuracy we decided to continue with ***Model#6*** for further experiments.  For the next experiment we decided to change activation function to **relu** to **sigmoid** |
| 11 | (8,16,32,64)  Dense layer: (256,128)  BN and DO with l2 regularization  Activation function: **sigmoid** | Training accuracy: 50%  Validation accuracy: 20% | **Explanation:**  We can see that the model performance: **validation accuracy and training accuracy,** degraded significantly. The model is not learning anything.  **Decision:**  As our aim is to find a model with best accuracy we decided to continue with ***Model#6.*** |
| Model#6 will be final model which we will consider for CNN  Training accuracy:84%  Validation accuracy: 81%. | | | |

**RNN (CONV-2D-LSTM) Model**

|  |  |  |  |
| --- | --- | --- | --- |
| E# | Model | Performance | Decision & explanation |
| 12 | RNN model (LSTM)  CONV 2D: (8,16,32,64)  Dense layer: (64)  LSTM: (64) | Training accuracy: 99%  Validation accuracy: 70% | **Explanation:**  As we can see from the graph that the loss decreased significantly as the number of epochs increased. However, after around 4 epochs, the **training accuracy** started to increase and reached 1 signifying that the model was over fitting. The **validation accuracy** started to decrease.  **Decision:**  Thus, we decided to introduce batch normalization and dropout for our next experiment. |
| 13 | RNN model (LSTM)  CONV 2D: (8,16,32,64)  Dense layer: (64)  LSTM: (64)  With dropout and batch normalization | Training accuracy: 46%  Validation accuracy: 56% | **Explanation:**  We can see in the graph that after introducing batch normalization and dropout the performance dropped.  **Decision:**  Thus, we decided to increase the complexity of the model by increasing number of layers. |
| 14 | RNN model (LSTM)  CONV 2D: (16,32,64,128)  Dense layer: (128)  LSTM: (256)  With dropout and batch normalization with L2 regularization | Training accuracy: 79%  Validation accuracy: 74% | **Explanation:**  We can see that the model **accuracy** increased to 74%.  However, the number of parameters is too big.  **Decision:**  **As, the model size was too big and did not suite our use case of less model size. We decided to stop experiments with CNN-LSTM and chose CNN-CONV3D model for the submission.** |

**Final Model**

After going through multiple experiments with different parameters and model architectures, we decided to with **CNN-CONV3D** model as it has less memory usage and suitable for the use case of being installed on a web cam. **The best model *model-00024-0.63102-0.84917-0.75914-0.81000.h5* is also being submitted with this write-up.** The best model has

**Training accuracy:84%**

**Validation accuracy: 81%**

A picture containing screenshot

Description automatically generated