**CASE STUDY**

**Neural Networks**

**Gesture recognition**

***Prepared By***

**Priyanshu** (priyanshu.cs17@gmail.com)

**Ankur Pandey**(matrixbeigns@gmail.com)

**Cohort**

Machine Learning Cohort 9

**Goal:**

Architect/develop a neural network to recognize following 5 hand gestures from a given clip of a video:

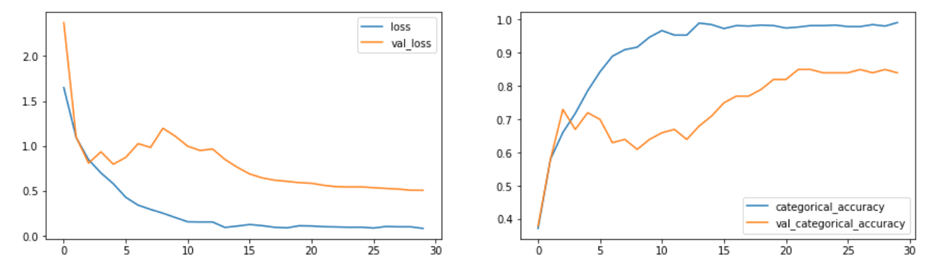
1. Stop
2. Thumbs Up
3. Thumbs Down
4. Swipe Left
5. Swipe Right

**Experiment # 1: 3DConv Model**

**Model details:** A slight variation of model given in starter code.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 30 | 30 | 40 | 98.47% | 85% | 0.5235 |



**Explanation:**

Model giving the best training accuracy of 98% and validation accuracy of 85% with **9 lakh** parameterswith **sample frame=30, Conv layer outputs=(16,32,64,128), batch=40, filter=(3X3X3),** **image pixels= 120X120** with **no dense and dropout.**

**Experiment # 2: 3DConv Model**

**Model details:** Trying to reduce number of parameters.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 20 | 20 | 40 | 97.94% | 79% | 0.5530 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the best training accuracy of 97.94% and validation accuracy of 79% with **6.99 lakh** parameterswith **sample frame=20, Conv layer outputs=(16,32,64,128), batch=40, image pixels= 120X120, dense=64 and dropout=0.25.**

**Experiment # 3: 3DConv Model**

**Model details:** Trying different filter size and dense layer value. Also reducing number of frame samples.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 30 | 16 | 30 | 90.27% | 85% | 0.5749 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the best training accuracy of 90.27% and validation accuracy of 85% with **1.7 million** parameterswith **sample frame=16, Conv layer outputs=(16,32,64,128), filter=(2X2X2), batch=30, dense=256, dropout=0.5 and image pixels= 120X120.**

**Experiment # 4: 3DConv Model**

**Model details:** Reducing the dense layer value and dropout values. Also the batch size is reduced.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 30 | 20 | 20 | 93.75% | 82% | 0.5634 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the best training accuracy of 93.75% and validation accuracy of 82% with **9 lakh** parameterswith **sample frame=20, Conv layer outputs=(16,32,64,128), filter=(2X2X2), batch=20, dense=128, dropout=0.25 and image pixels= 120X120.**

**Experiment # 5: 3DConv Model**

**Model details:** Changing the layer output filters at different layers. Dense layer output is also reduced.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 30 | 20 | 20 | 94.63% | 78% | 0.6540 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the best training accuracy of 94.63% and validation accuracy of 78% with **2.2 lakh** parameterswith **sample frame=20, Conv layer outputs=(8,16,32,64), filter=(2X2X2), batch=20, dense=64, dropout=0.25 and image pixels= 120X120.**

So far we have tried multiple version and combinations of 3D convolution neural networks. Building other models with **CNN + GRU** and **CNN + LSTM.**

**Experiment # 6: CNN + GRU**

**Model details:** A model with 6 layers and one GRU layer before network output

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 30 | 20 | 20 | 71.89% | 64% | 0.9102 |

A screenshot of a map

Description automatically generated

**Explanation:**

GRU didn’t work very well for us. Model giving the highest training accuracy of 71.89% and validation accuracy of 64% with **4 lakh** parameterswith **sample frame=20, Conv layer outputs=(8,16,16,32,64,128), filter= (3X3), batch=20, dense=128, dropout=0.25 and image pixels= 120X120.**

**Experiment # 7: CNN + LSTM**

**Model details:** A model with 6 layers and one LSTM layer before dense layers with minimum number of parameters.

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 20 | 20 | 20 | 94.41% | 77% | 0.6474 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the highest training accuracy of 94.41% and validation accuracy of 77% with **4 lakh** parameterswith **sample frame=20, Conv layer outputs=(16,16,32,32,128,256), filter= (3X3), batch=20, LSTM=128, dense=128, dropout=0.25 and image pixels= 120X120.**

Now building a model with transfer learning.

**Experiment # 8: MobileNet + GRU**

**Model details:** using mobile net as base network for transfer learning and putting a GRU layer before dense layers

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# of Epochs** | **# of Frame Sample** | **Batch size** | **Training Accuracy** | **Validation Accuracy** | **Validation Loss Value** |
| 20 | 20 | 10 | 98.81% | 96% | 0.1 |

A close up of a map

Description automatically generated

**Explanation:**

Model giving the best training accuracy of 98.81% and validation accuracy of 96% with **3.69 million** parameterswith **sample frame=20, batch=10, GRU=128, dense=128, dropout=0.25 and image pixels= 120X120.**

**NOTE:**

Due to size limitation we have uploaded the files on Google Drive.

|  |  |
| --- | --- |
| Model | Link |
| Model 4 | https://drive.google.com/file/d/1y5cLJtDps20oJUV1mCHm84QDH1QBEQAi/view?usp=sharing |
| Model 12 | https://drive.google.com/file/d/1WZaj-f1hZNevG03NfDCyWbRUOPAp4nWf/view?usp=sharing |

**Conclusion:**

Among various models that were built Model4 with validation accuracy 85% will be given as model of choice if we want to use Conv3D Model as our solution.

Having said that we also observed that a case of transfer learning where mobile net with GRU layers significantly outperformed all of the other models with validation accuracy of 96%. Therefore it’ll be our model of choice for gesture recognition solution.