Write – Up (Gesture Recognition)

1. Con3d

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Con3d1**  **(image size = 160x160, batch\_size=8, no.of epoch=1)**  **Data Augmentation = False** | **Trainable parameters: 17,35,525**  **Categorical\_accuracy**   1. **training = 36 %** 2. **validation = 16 %** | **To verify data generator code and making sure that code is correct.** |
| **2** | **Con3d1**  **(image size = 100x100, batch\_size=4, no.of epochs=3)**  **Filter size = 3x3x3 for first layer and 2x2x2 for layer 2,3 and 4. Dense layer = 128 neurons, dropout = 50 % and another layer of 64 neurons with 25 % dropout**  **Data Augmentation = False** | **Total parameters: 6,87,813**  **Categorical\_accuracy**   1. **training = 34.84 %** 2. **validation = 22 %** | **Reduction in batch size and increase in epochs resulted in better validation accuracy even with compressed image size.** |
| **3** | **Con3d1**  **(image size = 100x100, batch\_size=8, no.of epochs=3)**  **Filter size = 3x3x3 for first layer and 2x2x2 for layer 2,3 and 4. Dense layer = 128 neurons, dropout = 50 % and another layer of 64 neurons with 25 % dropout**  **Data Augmentation = False** | **Total parameters: 6,87,813**  **Categorical\_accuracy**   1. **training = 50 %** 2. **validation = 18 %** | **Increase in batch size contributed to training accuracy but resulted in decrease of validation accuracy compared to experiment 2.** |
| **4** | **Con3d1**  **(image size = 160x160, batch\_size=32, no.of epochs=3)**  **Filter size = 3x3x3 for first layer and 2x2x2 for layer 2,3 and 4. Dense layer = 128 neurons, dropout = 50 % and another layer of 64 neurons with 25 % dropout neurons with 25 % dropout**  **Data Augmentation = False** | **Total parameters: 17,36,389**  **Categorical\_accuracy**   1. **training = 71 %** 2. **validation = 25 %** | **Higher image information has contributed to the slight overfitting compared to experiment 3.** |
| **5** | **Con3d1**  **(image size = 160x160, batch\_size=8, no.of epochs=30)**  **Filter size = 3x3x3 for for layer 1,2,3 and 4. Dense layer = 64 neurons, dropout = 25 % and another layer of 64 neurons with 25 % dropout**  **Data Augmentation = False** | **Total parameters: 11,17,061**  **Categorical\_accuracy**   1. **training = 78.73 %** 2. **validation = 73 %** | **Increase in number of epochs and changes in model architecture given accuracy is achieved.** |
| **6** | **Con3d2**  **(image size = 160x160, batch\_size=8, no.of epochs=30)**  **Filter size = 3x3x3 for for layer 1,2,3 and 4. Dense layer = 256 neurons, dropout = 50 % and another layer of 256 neurons with 50 % dropout**  **Data Augmentation = True** | **Total parameters: 36,38,981**  **Categorical\_accuracy**   1. **training = 83.33%** 2. **validation = 83 %** | **Increase in the dense layer neurons resulted in better accuracy as compared to experiment 5.** |
| **7** | **Con3d3**  **(image size = 120x120, batch\_size=8, no.of epochs=30)**  **Filter size = 2x2x2 for for layer 1,2,3 and 4. Dense layer = 256 neurons, dropout = 50 % and another layer of 256 neurons with 50 % dropout** | **Total parameters: 17,62,613**  **Categorical\_accuracy**   1. **training = 74.43 %** 2. **validation = 82 %** | **Reduction in image size even with smaller filter size, resulting in decrease of accuracy.** |
| **8** | **Con3d4**  **(image size = 120x120, batch\_size=8, no.of epochs=30)**  **Filter size = 3x3x3 for for layer 1,2,3,4,5,6. Dense layer = 256 neurons, 2 dense layers of 256 neurons with 50 % dropout**  **Data Augmentation = True** | **Total parameters: 25,56,533**  **Categorical\_accuracy**   1. **training = 87.48 %** 2. **validation = 84 %** | **Compared to experiment 7, increase in filter size resulted in better accuracy.** |
| **9** | **Con3d5**  **(image size = 120x120, batch\_size=8, no.of epochs=30)**  **Filter size = 3x3x3 for for layer 1,2,3,4,5,6. Dense layer = 256 neurons, dropout = 25 % between 2 and 3, 4 and 5 and after 6. 2 dense layers of 256 neurons with 50 % dropout**  **Data Augmentation = True** | **Total parameters: 25,56,533**  **Categorical\_accuracy**   1. **training = 88.89 %** 2. **validation = 81 %** | **We got almost similar result as experiment 8 with reduced trainable parameters.** |
| **10** | **Con3d6**  **(image size = 100x100, batch\_size=4, no.of epochs=3)**  **Filter size = 3x3x3 for first layer and 2x2x2 for layer 2,3 and 4. Dense layer = 128 neurons, dropout = 25 % and another layer of 128 neurons with 25 % dropout**  **Data Augmentation = True** | **Total parameters: 6,96,645**  **Categorical\_accuracy**   1. **training = 84.46 %** 2. **validation = 77 %** | **We got decent results with very less parameters, compressed image 100x100.** |
| **11** | **Con3d7**  **(image size = 120x120, batch\_size=4, no.of epochs=3)**  **Filter size = 3x3x3 for first layer and 2x2x2 for layer 2,3 and 4. Dense layer = 64 neurons, dropout = 25 % and another layer of 64 neurons with 25 % dropout**  **Data Augmentation = True** | **Total parameters: 5,04,709**  **Categorical\_accuracy**   1. **training = 75.72 %** 2. **validation = 75 %** | **Reducing dense layer resulted in lesser training accuracy.** |

Conclusions:

* For the applications of low memory model from experiment 10 could be used
* For applications with sufficient memory, experiment 8 could be used.
* To get the best result, it is recommended to use image data augmentation with 30 epochs for training.
* 120x120 seems to reasonable image compression size.

1. RNN

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| 1 | RNN\_CNN1  Image size = 120x120, batch size = 8, epochs = 30.  3x3 convolution layer with 16 kernels, 3x3 convolution layer with 32 kernels, 3x3 convolution layer with 64 kernels, 3x3 convolution layer with 128 kernels, 3x3 convolution layer with 256 kernels, 1 layer with 128 LSTM cells, dense layer with 128 neurons with 25 % dropout.  Data Augmentation = False | **Total parameters: 16,57,445**  **Categorical\_accuracy**   1. **training = 89.29 %** 2. **validation = 79 %** | Time series model has helped to achieve good accuracy with lesser parameters compared to the experiment 8 of conv3d |
| 2 | RNN\_CNN1  Image size = 120x120, batch size = 8, epochs = 30.  3x3 convolution layer with 16 kernels, 3x3 convolution layer with 32 kernels, 3x3 convolution layer with 64 kernels, 3x3 convolution layer with 128 kernels, 3x3 convolution layer with 256 kernels, 1 layer with 128 LSTM cells, dense layer with 128 neurons with 25 % dropout.  Data Augmentation = True | **Total parameters: 16,57,445**  **Categorical\_accuracy**   1. **training = 87.41 %** 2. **validation = 81 %** | Data Augmentation has helped us achieved better validation score. |
| 3 | RNN\_CNN2  Image size = 120x120, batch size = 8, epochs = 30.  3x3 convolution layer with 16 kernels, 3x3 convolution layer with 32 kernels, 3x3 convolution layer with 64 kernels, 3x3 convolution layer with 128 kernels, 1 layer with 128 GRU cells, dense layer with 128 neurons with 25 % dropout.  Data Augmentation = True | **Total parameters: 25,73,925**  **Categorical\_accuracy**   1. **training = 99.4 %** 2. **validation = 81 %** | The GRU cells has overfitted on the training data. |
| 4 | RNN\_CNN\_TL  Image size = 120x120, batch size = 8, epochs = 30.  Mobilenet con2d (trainable parameters = False)  LSTM layer with 128 cells  Dense layer of 128 with 25 % dropout.  Data Augmentation = True | **Total parameters: 38,40,453**  **Categorical\_accuracy**   1. **training = 99.47 %** 2. **validation = 79 %** | Failed experiment it has overfitted model with more parameters compared to experiment 3. |
| 5 | RNN\_CNN\_TL2  Image size = 120x120, batch size = 8, epochs = 30.  Mobilenet con2d (trainable parameters = TRUE)  GRU layer with 128 cells  Dense layer of 128 with 25 % dropout.  Data Augmentation = True | **Total parameters: 36,93,253**  **Categorical\_accuracy**   1. **training = 1 %** 2. **validation = 97 %** | Since we have allowed mobilenet weights to be learnt by model we got better accuracy in validation. |

Conclusions:

* For high computation high accuracy application, experiment 5 of RNN model is recommended.
* For low computation reasonable accuracy application like T.V. experiment 10 of Con3d is recommended.