**Gesture Recognition Experiment writeup**

**Model Architecture**

There are 3 different deep learning architectures which can be used to build model for gesture recognition. Let’s see those architectures below.

**CNN + RNN**

This is the standard architecture for processing videos. In this architecture, video frames are passed through a CNN layer which extracts features from the images and then these feature

vectors are fed to an RNN network to simulate sequence behavior of the video. Output of RNN is regular SoftMax function/

1. We can use transfer learning in 2D CNN layer instead of training own network.

2. LSTM or GRU can be used in RNN

**3D Convolution Network or Conv3D**

3D convolutions are a natural extension to the 2D convolutions. Just like in 2D conv, we move the filter in two directions (x and y), in 3D conv, the filter is moved in three directions

(x, y and z). In this case, the input to a 3D conv is a video (which is a sequence of 30 RGB images). If we assume that the shape of each image is 100x100x3, for example, the video becomes a 4-D

tensor of shape 100x100x3x30 which can be written as (100x100x30)x3 where 3 is the number of channels. Hence, deriving the analogy from 2-D convolutions where a 2-D kernel/filter (a square filter)

is represented as (fxf)xc where f is filter size and c is the number of channels, a 3-D kernel/filter (a 'cubic' filter) is represented as (fxfxf)xc (here c = 3 since the input images

have three channels). This cubic filter will now '3D-convolve' on each of the three channels of the (100x100x30) tensor

**Transfer Learning**

We will use some pre-trained models and try to use their knowledge to classify our cases correctly. These pre-trained models which are trained on millions of images may prove vital

in solving our problem efficiently. For transfer learning -

• We have done our experiments with the following pre-trained model using MobileNet

• We experimented several such models in the 2D CNN layer.

• We experimented with GRU units as well as LSTM units.

1. We experimented to train the model 11 times. For all the models we used 32 batch size and with 25 epochs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Number** | | **Model** | **Result** | **Decision + Explanation** | |
| **1** | | **Conv2D + RNN** | **Accuracy: 0.99 Validation accuracy: 0.27** | **batch size = 32 epochs = 25**  **Total params: 921,189**  **Trainable params: 920,197**  **Non-trainable params: 992**  **Accuracy is 99% but the validation accuracy is only 27%, Model is overfitted and not learned much.** | |
| **2** | | **Conv2D + RNN using TimeDistributed with GRU** | **Accuracy: 0.9789 Validation accuracy: 0.25** | **Same as experiment – 1 , model is overfitted. batch size = 32 , epochs = 25**  **Total params: 856,421**  **Trainable params: 855,429**  **Non-trainable params: 992** | |
| **3** | | Conv3D along with MaxPooling3D and 4 Convolution Layers | **Accuracy: 0.7104 Validation accuracy: 0.24** | **Accuracy is reduced compared the experiment 2 and validation accuracy remains the same. Batch size and epochs are same.**  **Training Accuracy: 71.04**  **Validation Accuracy: 24.0** | |
| **4** | | **Conv3D along with MaxPooling3D and 5 Convolution Layers** | **Accuracy: 0.8205 Validation accuracy: 0.24** | **Validation accuracy is still not good as experiment – 3. batch size = 32,Epochs 25** | |
| **5** | | **Conv3D** | **Accuracy: 0.6576 Validation accuracy: 0.35** | **Almost Same as experiment –4 , Validation accuracy is the increased a little.**  **Training Accuracy: 65.76**  **Validation Accuracy: 35.0** | |
| **6** | | **Conv3D: Adding Dropout after each Conv3D Layer** | **Accuracy: 0.4434 Validation accuracy: 0.2700** | **Training accuracy is dropped drastically. epochs, batch size remains the same.**  **Training Accuracy: 44.34**  **Validation Accuracy: 27.0**  **Total params: 1,287,989**  **Trainable params: 1,286,773**  **Non-trainable params: 1,216** | |
| **7** | | **Conv3D : Adding L2 Regularizer in Fully Connected layer in Conv3D Network** | **Accuracy: 0.6848 Validation accuracy: 0.39** | **Model remains same. Validation accuracy increased.batch size = 32**  **epochs=25**  **Training Accuracy: 68.48**  **Validation Accuracy: 39.0**  **Total params: 699,269**  **Trainable params: 698,533**  **Non-trainable params: 736** | |
| **8** | **Conv3D : L2 Regularization, Adding MaxPooling3D after every two similar Conv3D Layers** | | **Accuracy: 0.7134 Validation accuracy: 0.26** | **batch size = 32**  **epochs=25**  **Training Accuracy: 71.34**  **Validation Accuracy: 26.0**  **Total params: 1,287,989**  **Trainable params: 1,286,773**  **Non-trainable params: 1,216**  **Model validation accuracy is less.** |
| **9** | **Conv3D :** **Adding more L2 Regularization along with MaxPooling3D after every two similar Conv3D Layers** | | **Accuracy: 0.8733 Validation accuracy: 0.32** | **Training accuracy is increased but the Validation accuracy is still only 32%.**  **batch size = 32**  **epochs=25**  **Training Accuracy: 87.33**  **Validation Accuracy: 32.0**  **Total params: 5,782,261**  **Trainable params: 5,781,045**  **Non-trainable params: 1,216** |
| **10** | Transfer Learning using MobileNet Model with LSTM | | **Accuracy: 100 Validation accuracy: 0.7900** | **Model training accuracy and validation**  **accuracy is reached maximum compared to the previous models.**  **Training Accuracy: 100.0 , Validation Accuracy: 79.0**  **Model is having training and validation accuracy as 99% and 79% which shows model is good but can become better and able to learn the behavior.**  **batch size = 32**  **epochs=25** |
| **11** | Transfer Learning using MobileNet Model with GRU | | **Accuracy: 1.000 Validation accuracy:**  **0.9800** | **After all above model experiment, we are going to choose "Model 11 - Transfer Learning using MobileNet Model with GRU" as good model which performed well.**  **Training Accuracy: 100.0**  **Validation Accuracy: 98.0**  **batch size = 32**  **No of epochs=25**  **Total params: 3,446,725**  **Trainable params: 3,422,789**  **Non-trainable params: 23,936**  Best Model file: model-00022-0.00800-0.99849-0.16037-0.96000.h5 |

**Summary**

In this case study, we experimented on 3 different types of model architectures for gesture

recognition:

1. CNN (Conv2D) + RNN

2. CNN (Conv3D)

3. CNN (Transfer Learning) + RNN

Based on the model performance we selected our top model out of the 11 models we

tested. A high-level summary of top model is written out below.

Training Accuracy: 100.0

Validation Accuracy: 98.0

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

Graphical user interface

Description automatically generated

Final Verdict

From the summary above we see that our transfer learning model built using Transfer Learning using MobileNet Model with GRU pre-trained model achieves the best validation accuracy of 98% without overfitting on the training data. The training accuracy stands at 100% with a very minimal loss of information.

Thus, we choose Model 11 as the best model for the task of gesture recognition