8/17/2022

Write Up

1. Results of All the model experiments which was done for two architectures
    a. Conv3D
    b. Con2D with RNN
        i. RNN – GRU
        ii. RNN – LSTM

Gesture Recognition – Deep Learning

Problem Statement

You want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote

The gestures are continuously monitored by the webcam mounted on the TV. 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

Model No	Model Name	Result	Decision + Explanation
Architecture 1 – Using Conv3D			
1 & 2	**Conv3D**	**Model 1** threw out of memory error for batch size 128. So, we decided to stick with batch size 32 and 64. **Model 2** Throws Generator error	Batch size to be used as 32. Image resolution would be used at 120*120 to see the image size has clear visibility & improves accuracy.
3	**Conv3D with Epoch = 20**	**Model 3** is the 1st model which got results. Train Accuracy - 0.90 Validation Accuracy – 0.27	Model is overfitting now. So going to increase the epoch size to 50 and train the model
4	**Conv3D with Epoch = 50**	**Model 4** is the 2nd model which got results better than model 3 Train Accuracy - 0.90 Validation Accuracy – 0.71	Model is still overfitting, but the validation accuracy is increased due to increase in epoch. So, let us try with adding some dropouts
5	**Conv3D with Dropout = 0.2**	**Model 5** Train Accuracy - 0.98 Validation Accuracy – 0.76	So, by adding some dropouts as 0.2, both the train and validation accuracy is increased. Again, going to try with increase in dropouts' values
6	**Conv3D with Dropout = 0.5**	**Model 6** Train Accuracy - 0.97 Validation Accuracy – 0.51	By increasing the dropouts to 0.5, it proved that it will not be suited. Both the values are decreased and still the model is overfitting. Now going to use average polling instead of flatten layer.
7	**Conv3D using Global Average**	**Model 7** Train Accuracy - 0.95 Validation Accuracy – 0.90	Now it is the best model in architecture 1 (using CONV3D). But the training parameters are little high. So let's try with different architectures using Conv 2D and RNN.

Architecture 2 – Using Conv2D & RNN			
8	**Time Distributed Conv 2D + GRU (without Dropouts)**	**Model 8** Train Accuracy - 0.95 Validation Accuracy – 0.82	The model is performing good with less parameters. Let us add some dropouts so that we can make both the train and validations scores close to each other
9	**Time Distributed Conv 2D + GRU (with Dropouts)**	**Model 9** Train Accuracy - 0.87 Validation Accuracy – 0.60	Validation Accuracy is again lowered, and overfitting happened. So, we are going to remove the dropouts and add Dense layer instead of GRU
10	**Time Distributed Conv 2D + Dense (without Dropouts)**	**Model 10** Train Accuracy - 0.87 Validation Accuracy – 0.87	This is the good model with both the values with less parameters like (128517). Again, trying with different parameters like LSTM instead of GRU
11	**Time Distributed + ConvLSTM2D**	**Model 11** Train Accuracy - 0.96 Validation Accuracy – 0.93	**This is the final model with good amount of train accuracy and validation accuracy and even the parameters are very less (13589).**