Gesture Recognition Case Study

This document contains all the experiments we have done with the architecture of our neural network and different algorithms which we tried to achieve good accuracies and reduce loss on our models.

We tried 2 approaches in this case study and all our experiments revolve around these 2 approaches.

1. 3DCNN
2. Transfer Learning + RNN

3DCNN Models

Model	Accuracy	Params	Architectural Changes
3DCNN	Train –98% Val – 96%	Epochs-20 Image-140x140 Frames-15	We used Batch Normalization here and did not use any dropouts. We noticed that even though we are not using dropouts our model did not overfit.
3DCNN	Train –21% Val – 20%	Epochs-20 Image-120x120 Frames-15	We Introduced some dropouts and as a result our model did underfit, due to the loss of information.
3DCNN	Train –60% Val – 59%	Epochs-20 Image-120x120 Frames-15	We reduced the number of dropouts in the network and kept it only in the last layer at 0.25 value. We got significantly better results but not the best.
3DCNN	Train –98% Val – 94%	Epochs-20 Image-140x140 Frames-10	This Model Among all our 3D CNN models performed the best. Although the frame size was 10 but we did not add any dropouts in the model.

Transfer Learning Models

Model	Accuracy	Params	Architectural Changes
RESNET+GRU	Train -78% Val – 92%	Epochs-20 Image-140x140 Frames-10	Here we tried resnet and gru with 10 frames in the hope that it might work well with 10 frames just like it did for 3DCNN, but it did not.
RESNET+GRU	Train -96% Val – 98%	Epochs-20 Image-140x140 Frames-15	We Ran the same model as above but with 15 frames this time. And we got significantly good results with more number of frames.
Xception+GRU	Train -84% Val – 88%	Epochs-20 Image-120x120 Frames-15	Kept the network similar to above and replaced RESNET transfer learning to Xception transfer learning.

RESNET+LSTM	Train -99% Val – 99%	Epochs-20 Image-140x140 Frames-15	As our model performed well with RESNET and GRU, as we are aware LSTM has an extra gate in every RNN unit which we train. So, this might help us shoot up our accuracies even more. And it did. This was the best model we achieved so far.

Conclusion

1. 3DCNN
    a. 3DCNN gave us exceptional results when we used only 10/30 images from a video to train our model.
    b. 3DCNN tend to gave better results when we removed dropouts, when we added dropouts to the model the accuracy dropped miserably leading to an underfitting model.
2. Transfer Learning
    a. RESNET+GRU gave us good results, when we reduced the number of dropouts in this model we observed that the accuracy went up and we achieved around 96% accuracy on the val data
    b. XCEPTION+GRU gave us good results but <u>3DCNN</u> and <u>RESNET+GRU</u> proved to be better models for our dataset.
    c. RESNET+LSTM proved out be the best model for us, as the accuracy we achieved on the training and val set was approximately 99% on both.