

Gesture Recognition Write up

Problem Statement: Imagine you are working as a data scientist at a **home electronics company** which manufactures state of the art smart televisions. You want to develop a cool feature in the **smart-TV** that can recognize **five different gestures** performed by the user which will help users control the TV without using a remote.

Experiment Number	Model	Result	Decision + Explanation
1	Conv3D	Train accuracy = 0.6516 Validation accuracy = 0.22 Epochs = 21	Highly overfitting model. Even train model isn't performing that well so we first target train accuracy and will later target validation accuracy.
2	Conv3D	Train accuracy = 1.00 Validation accuracy = 0.78 Epochs = 32	We added extra dense layer to the model. Since, features from CNN were not being learned properly. Model still overfitted.
3	Conv3D	Train accuracy = 0.92 Validation accuracy = 0.74 Epochs = 21	We tried adding batch normalization layer and dropout layers to reduce overfitting.
4	Conv3D	Train accuracy = 1.00 Validation accuracy = 0.72 Epochs = 21	Previously batch normalization wasn't added in dense layers so tried doing it to see if the overfitting is handled.
5	Conv3D	Train accuracy = 1.00 Validation accuracy = 0.75 Epochs = 32	Reducing the CNN layers and unnecessary dense layers to reduce overfitting but couldn't manage overfitting.
6	CNN (Transfer learning) + LSTM	Train accuracy = 0.44 Validation accuracy = 0.46 Epochs = 32	Overfitting is addressed but the model is underfitting.
7	CNN (Transfer learning) + GRU	Train accuracy = 0.65 Validation accuracy = 0.61 Epochs = 21	Model is slightly better then random model and addresses overfitting as well.

Since, the models are overfitting or is underfitting we download weights of model which gave least overfitting and highest accuracy from experiment number 3 which are attached in zip file.