COMPUTER VISION ASSIGNMENT - 4

AKSHAY KUMAR PANSARI - 2015ME20713
ANKIT KUMAR - 2017MT10727

Hand Gesture Recognition

INTRODUCTION:

The purpose of this assignment is to train a Convolutional Neural Network to recognise the hand gestures. The hand gestures could be termed as Previous, Next and Stop signifying the sign used to control a Music Player.

MODEL:

We chose to use Keras due to our familiarity with it and also because there were few things which were beneficial such as callbacks.

We decided to use 3 Conv Layer with Channel Size of (16, 32, 64) and one dense layer with size of 400 which was chosen after tuning the hyperparameter. Also we chose the optimiser over ADAM over RMSPROP as shown in the model below.

Preprocessing Steps:

Without PreProcessing:

In the first iteration, we were not doing any pre-processing. We thought that the model should be able to predict as we had a lot of data. Then as we were testing it, we realised that it was not general enough and it was giving biased output based on the background with which it was trained on. For eg it was giving result also based on the colour of the t-shirt or the face of the version.

Pre-processing Step:

So, we thought to remove this. There were many things that we tried. First was using dilation and erosion, but it didn’t work out. Then we tried to just run on the model on the hand. We then extracted only those colours which were representative of the skin. So during training we applied this function to each training images and we found that the model was working better than before and on the validation set we found the accuracy to be 93%.

Data Augmentation:

The next thing that we tried was data augmentation. In this we added zoom factor, vertical flip and rotation of images by certain degree. But then we noticed that it was getting relatively poor accuracy, and our model was able to generalise over various transformations very well without this itself. So, we decided to remove this.

HYPERPARAMETERS:

We optimised our model to get the best parameters. We realised that ADAM had the best accuracy with dense connected layer value to be 400. This was surprising as we thought that as we will increase the number of layers the accuracy should increase but this was not the case.

We also varied learning rate(0.01 to 0.0001) and dropout values and it turned out that the hyperparameter that were best are reported below.

Accuracy on VALIDATION set with different hyperparameter .

Dense Connected Layer	RMS	ADAM
200	0.8966	0.9060
400	0.89600	0.9234
500	0.9113	0.9078
700	0.9051	0.890
1000	0.8850	0.9086
1200	0.9095	0.9144

TABLE1 Hyperparameter tuning

Learning Rate: 0.001

Optimizer: [Adam ~~vs Rmsprop]~~

Kernel Size: 3*3 ~~vs 5*5~~

Flatten:~~ 200,~~ 400, ~~500, 700, 1000, 1200~~

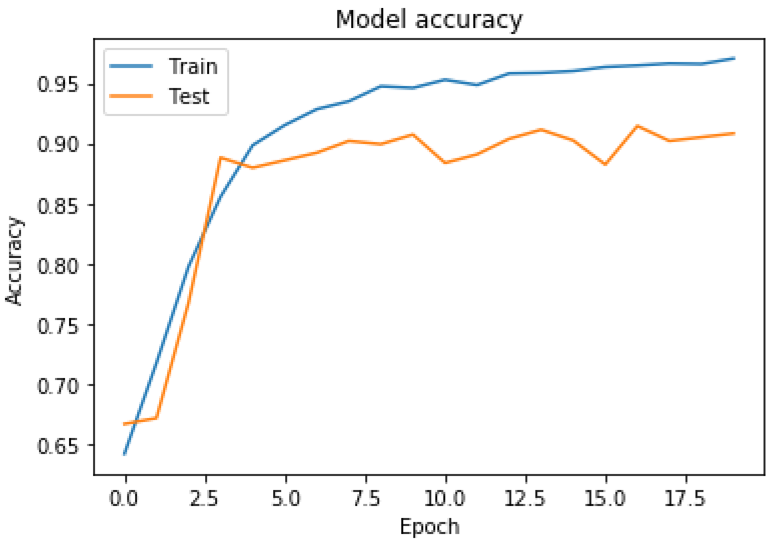
Conv Layer vs Dense Layer: 3 Conv, 2 Dense

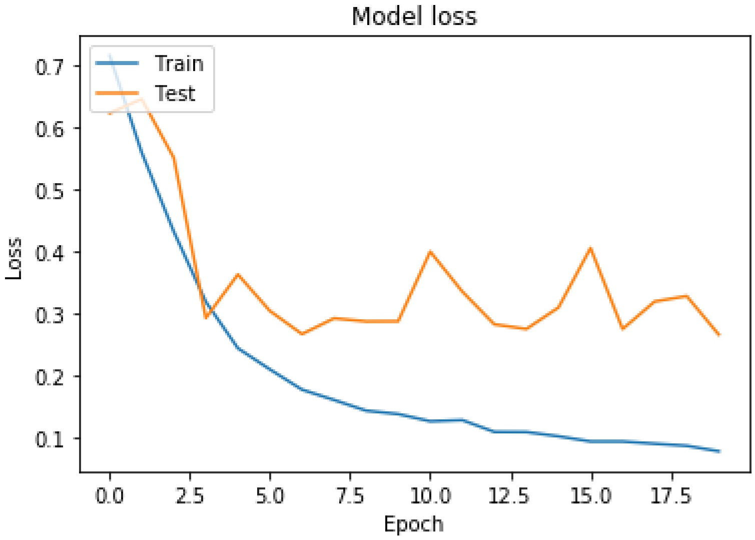
Dropout Value : 0.5 ~~vs 0.2~~

Patience: 2

Activation Function : First 4 layer RELU, last layer Softmax

PLOTS





WEIGHTS

REFERENCES:

https://keras.io/preprocessing/image/

https://keras.io/callbacks/

https://stackoverflow.com/questions/47266383/save-and-load-weights-in-keras

https://stackoverflow.com/questions/35074549/how-to-load-a-model-from-an-hdf5-file-in-keras

https://stackoverflow.com/questions/36952763/how-to-return-history-of-validation-loss-in-keras