PROJECT 3

BEHAVIORAL CLONING

The project folder will contain the following……

modeltest.h5	To test if everything was connecting properly or not
modelLeNET.h5	Model using LeNET architecture for Track 1
modelNVIDIA.h5	Model using NVIDIA architecture for Track 1
model.h5	Generalized model using images from both the tracks

Then it contains the videos……

runTEST.mp4	Video of the car during the test
runLeNET.mp4	Video of the car during the LeNET architecture
runNVIDIA.mp4	Video of the car during the NVIDIA architecture
runmodelTRACK1.mp4	Video of the car on Track 1 using the generalized model

LeNET:

The architecture used was a very basic one containing-

- Lambda layer
- Cropping layer (output image
- 1st Convolutional layer (ReLU activation)
- Max Pooling layer
- 2nd convolutional layer (ReLU activation)
- Flattening layer
- Three fully connected layers

As seen from the video the car is able to go around the track without much trouble until the point it encounters the road with dusty boundaries. At this point the car leaves the road and goes on a journey of its own and then into the road side ditch, where it was then stuck.
In the end I don't want to near any self driven car employing this network.

The network though was quick and performed fairly well but cannot be employed and needs lots of improvements.

Suggested improvement-

- Use of a masking region (it will block all the things that might distract the car).
- Better pre-processing, instead of just normalizing we can employ other methods as well, such as grayscaling, HOG transformation etc.
- Using more data (it will also put more load on the system)
- Generate fake images.

Also since this model was created using the images from Track 1 only, it cannot be employed to track 2. As it will be biased to the turns of Track 1.

NVIDIA:

The architecture used was the one used by NVIDIA in their self-driving cars. Any changes in the architecture proved to be a factor of deterioration in the model and were henceforth removed.
The architecture is as follows-

- Lambda
- Cropping
- 1st Convolutional layer
- 2nd Convolutional layer
- 3rd Convolutional layer
- 4th Convolutional layer
- 5th Convolutional layer
- Flattening layer
- 4 fully connected layer

The network performed quite well. The car went around the whole track without going off road at any moment, though it came close sometimes. But at the ending phases of the track the car starts to swerve left and right, which is never a good sign, suggesting that the network isn't very accurate.

The performance can be improved through pre-processing and using more images to train the network with.

There was no problem in the network itself, it is a network used by NVIDIA to drive their autonomous cars, so why would there be any trouble…

Suggested improvements-

- Enhance pre-processing of the training images.
- Using more data.

GENERALIZED NETWORK (Modified NVIDIA):

This network also utilized the NVIDIA(modified) architecture it just used more images and the images were from both the tracks.
The difference being that there was batch normalization layer after every convolutional layer, helping in generalizing.

The car had no problem going around the first track and stayed in the centre of the road.
But on trying out for the 2nd Track, the car could not reach the first turn. Suggesting improvements we add a few more images from the second track.

- Lambda
- Cropping
- 1st Convolutional layer
- Batch Normalization
- 2nd Convolutional layer
- Batch Normalization
- 3rd Convolutional layer
- Batch Normalization
- 4th Convolutional layer
- Batch Normalization
- 5th Convolutional layer

- Batch Normalization
- Flattening layer
- 4 fully connected layer

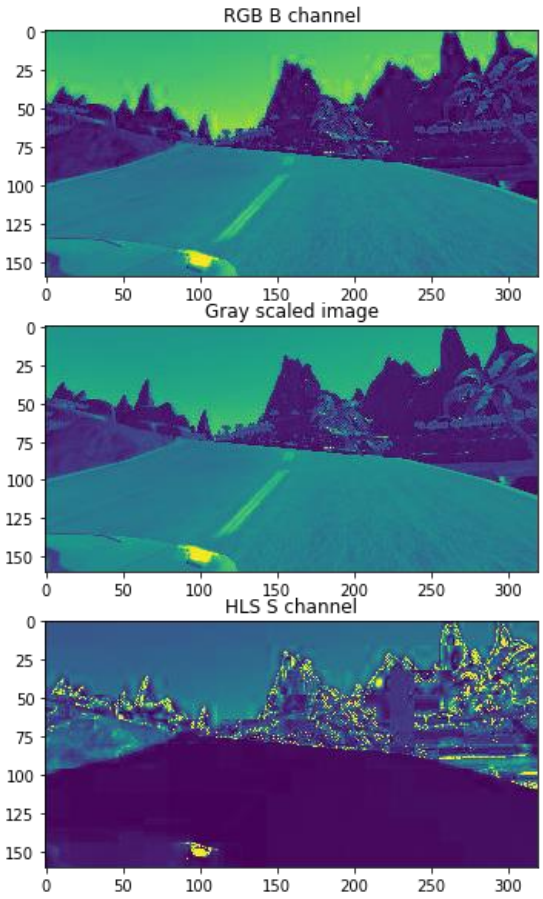
But still I could not achieve the desired result. So I failed to complete the challenge problem and submit the project with the code written, compiled and video for TRACK 1 only, however I will continue to work on the 2nd track.

PRE-PROCESSING Methods employed:

- The methods used in pre-processing are simple and basic, the first being normalization of the images.
- The second being cropping out the sky and hood of the car from the images.
- The other method being flipping the centre images, it will the model generalize better, as for very image telling the car to go right there will be an image telling the car go left.

(**NOTE: This isn't necessarily a pre-processing step but rather a way to increase the data, but you can also interpret it as a way of generating images [which is a pre-processing step]).**

- I tried grayscaling the images but the loss increased from around 0.002 to 1.005, so it was not an option. I also came to know from a fellow classmate on slack that grayscaling was only reducing the performance of the car and not enhancing it. WHICH IS TRUE CONSIDERING THAT GRAYSCALING ESSENTIALY REMOVES INFORMATION FROM THE IMAGE BY CONVERTING OUR **3D IMAGE** INTO A **2D IMAGE.**
- Here are some of the examples of how the images look in the different colour spaces.

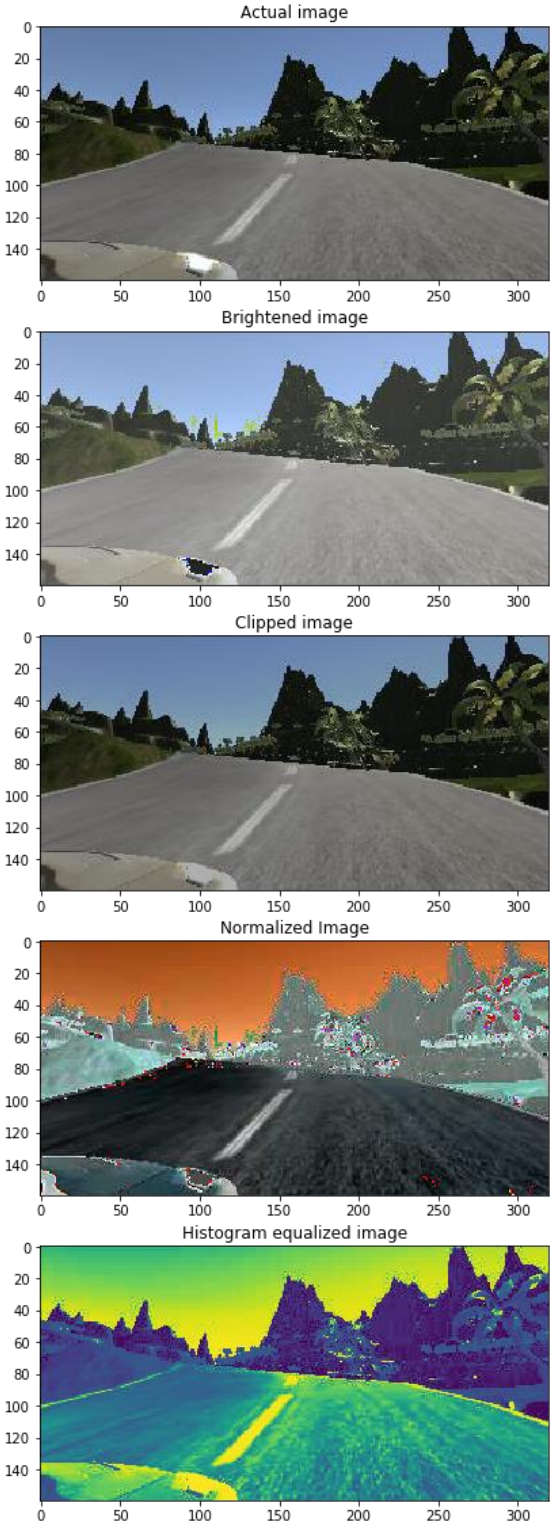


THOUGH HLS SPACE MAY HAVE BEEN A GOOD APPROACH IN ADVANCED LANE FINDINGS IT DOESN'T WORK HERE.

Now as seen from the three images RGB B channel and the grayscaled image look almost the same, but using the RGB image will be more beneficial for us, as this image carries more data (being the third layer in an image), whereas the gray scaled image carries much lesser information.

And the results of the HLS S space are easily visible, this due the fact that S space is the saturation space and the saturation of the colors in the middle regions is a lot. Thus essentialy making the image look as if someone has a complete bottle of purple colour on the image.

- Here are a few more examples of how the images look through different methods of normalization.



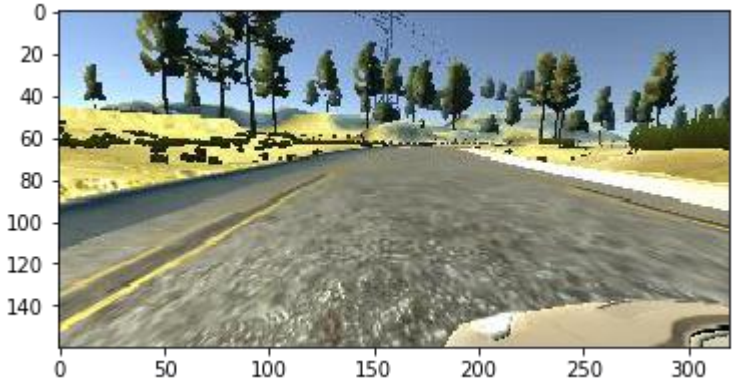
I increased the brightness of the images, because the jungle was quite dark compared to the lake track, which is making the steering difficult.

As can be seen in the normalized image the details are visible quite well. Compared to the image which was normalized without brightening.

Histogram equalization was disregarded because of the fact that it takes grayscale images as input and as we know grayscaling removes a lot of data from the images.

OTHER IMPORTANT POINTS:

- The other important thing is the correction factor**.** This allows the images taken from the left and right cameras of the car provide and important input to the car and help keep the car in the centre of the road.
  The way it is done correction factor is either added or subtracted from the steering labels if the images is taken from the left or right camera of the car respectively.



Here is an image taken from the left side of the car

- The other important thing is the memory and the data that is generated during these cloning operations is quite large going as high as 10GB (I ran into a lot of memory allocation problems due to this).
  This can be resolved in several ways (I used the following two methods)-
  1. Using the garbage collector function gc.collect().
     This allows all the redundant memory locations to be freed up allowing more space to be available to the larger datasets. This did the work in the most cases but when the data set for images became as high as 20000 images it was of no use.

  2. Here was the time to use the generators. What is does is, it provides the data when required and doesn't store in the memory all at once. Thereby making it more memory efficient.
  3. I decided to use both the images from track 1 and 2 so that my network, could help generalize better (because the lake track is quite bright and the jungle track is rather dim).

There is important improvement and probably the most important and that is, we are currently using only the steering angles to drive our car (that is a single label), we can enhance the performance a lot by using two labels for a single image like throttle and steering angle, this will improve the performance a lot , I didn't do it because it was too much for me, (I am relatively new to python so I didn't know the appropriate changes to make in the drive.py file to achieve this. I will be very glad if you could guide me on this).

Thank you very much, you made it to the end of this boring report.