
Self-Driving Car

Behavioural Cloning

NVIDIA Model

Training set:

- Self-driving car simulator, Open-Source by Udacity.
- Preprocessing techniques used are

>RGB to YUV (recommended by Nvidia model architects)

>Gaussian blur to smoothing the image and to reduce noise within the image.

>Resize of our image to 200 by 66.

(Input Size in the CNN)

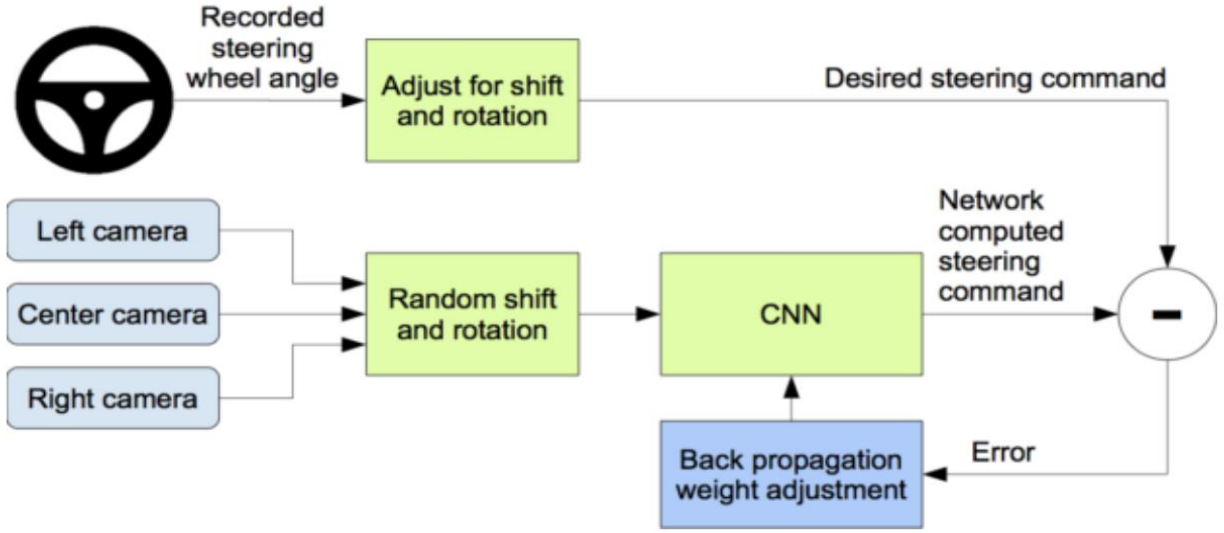


Figure 3: Training the neural network.

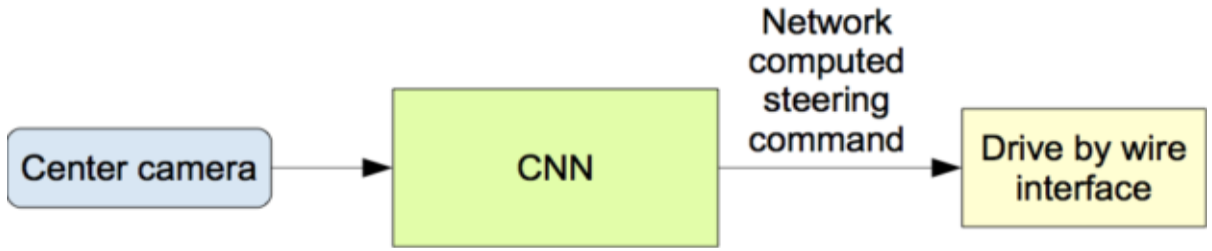
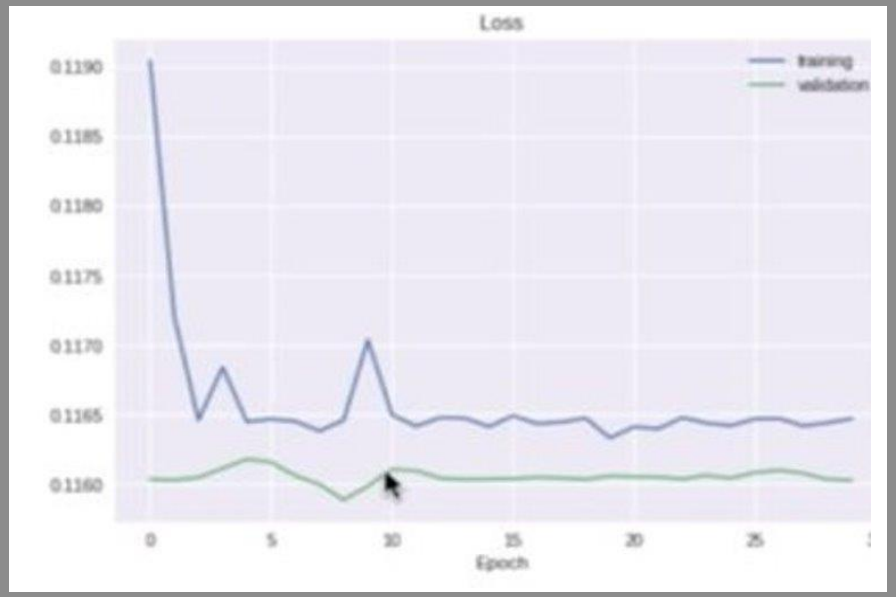
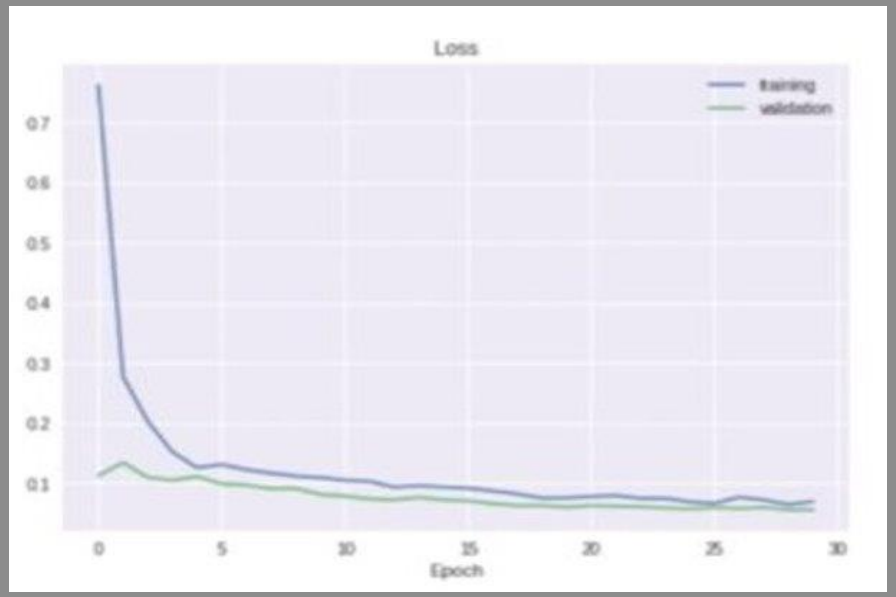


Figure 4: The trained network is used to generate steering commands from a single front-facing center camera.





NVIDIA Model

```
model = Sequential()
model.add(Convolution2D(24, 5, 5, subsample=(2, 2), input_shape=(66, 200, 3), activation='elu'))
model.add(Convolution2D(36, 5, 5, subsample=(2, 2), activation='elu'))
model.add(Convolution2D(48, 5, 5, subsample=(2, 2), activation='elu'))
model.add(Convolution2D(64, 3, 3, activation='elu'))

model.add(Convolution2D(64, 3, 3, activation='elu'))
#   model.add(Dropout(0.5))


model.add(Flatten())

model.add(Dense(100, activation = 'elu'))
#   model.add(Dropout(0.5))

model.add(Dense(50, activation = 'elu'))
#   model.add(Dropout(0.5))

model.add(Dense(10, activation = 'elu'))
#   model.add(Dropout(0.5))

model.add(Dense(1))

optimizer = Adam(lr=1e-4)
```

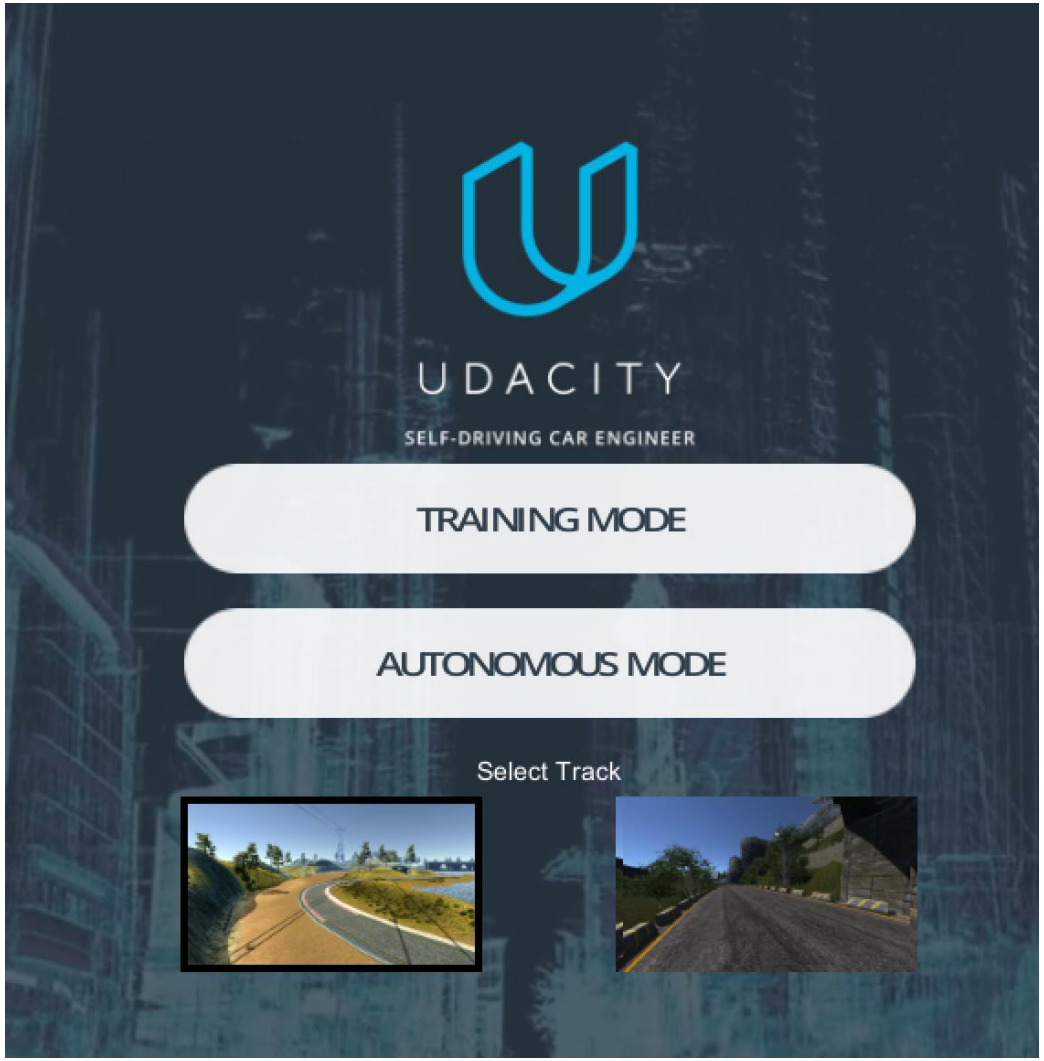
AUGMENTATION TECHNIQUES

Zooming, panning, random brightness, flip.

BATCH GENERATOR:

A memory efficient method to create small batches of images at the time only when the generator is called.

AND NOW....



References

- https://arxiv.org/pdf/1604.07316v1.pdf (End to End learning for self-driving cars)
- https://devblogs.nvidia.com/deep-learning-self-driving-cars/
- Learning Robotics using Python-Lentin Joseph

THANK YOU!

Special Thanks:
Bharghava Rajaram
Gopinath GR

Nithin Madadi

Lokesh Kondapaneni

Himaja Chanumaolu

Jahnavi Vikrama