

```
!git clone https://github.com/rslim087a/track
```

```
Cloning into 'track'...
remote: Enumerating objects: 12163, done.
remote: Total 12163 (delta 0), reused 0 (delta 0), pack-reused 12163
Receiving objects: 100% (12163/12163), 156.98 MiB | 32.38 MiB/s, done.
Checking out files: 100% (12160/12160), done.
```

```
!ls track
```

```
driving_log.csv  IMG
```

```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import keras
from keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import cv2
import pandas as pd
import ntpath
import random
```

```
datadir = 'track'
columns = ['center', 'left', 'right', 'steering', 'throttle', 'reverse', 'speed']
data = pd.read_csv(os.path.join(datadir, 'driving_log.csv'), names=columns)
pd.set_option('display.max_colwidth', -1)
data.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: FutureWarning: Pas
after removing the cwd from sys.path.
```

**center**

0	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_382.jpg	C:\Users\Ame
1	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_670.jpg	C:\Users\Ame
2	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_724.jpg	C:\Users\Ame
3	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_792.jpg	C:\Users\Ame
4	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_860.jpg	C:\Users\Ame

```
def path_leaf(path):
    head, tail = ntpath.split(path)
    return tail

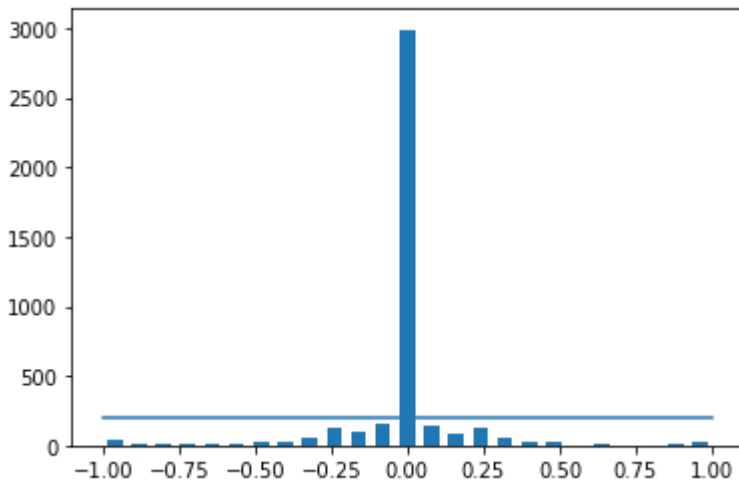
data['center'] = data['center'].apply(path_leaf)
data['left'] = data['left'].apply(path_leaf)
data['right'] = data['right'].apply(path_leaf)
```

```
data.head()
```

	center	left	
0	center_2018_07_16_17_11_43_382.jpg	left_2018_07_16_17_11_43_382.jpg	right_2018_07_
1	center_2018_07_16_17_11_43_670.jpg	left_2018_07_16_17_11_43_670.jpg	right_2018_07_
2	center_2018_07_16_17_11_43_724.jpg	left_2018_07_16_17_11_43_724.jpg	right_2018_07_
3	center_2018_07_16_17_11_43_792.jpg	left_2018_07_16_17_11_43_792.jpg	right_2018_07_
4	center_2018_07_16_17_11_43_860.jpg	left_2018_07_16_17_11_43_860.jpg	right_2018_07_

```
num_bins = 25
samples_per_bin = 200
hist,bins = np.histogram(data['steering'],num_bins)
center = (bins[:-1]+bins[1:]) * 0.5
plt.bar(center,hist,width=0.05)
plt.plot((np.min(data['steering']),np.max(data['steering'])),(samples_per_bin,samples_per_bin))
```

[<matplotlib.lines.Line2D at 0x7fa944aecc50>]

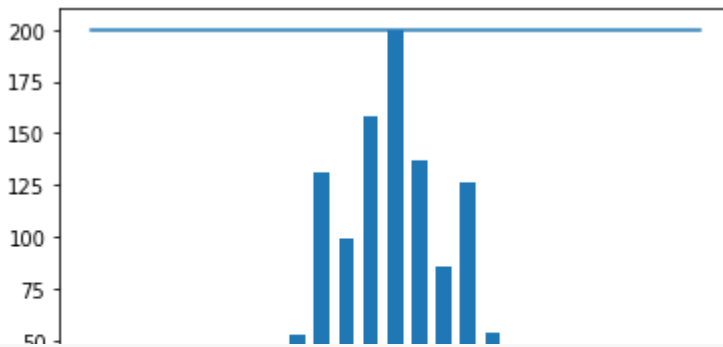


```
print('Total Data:',len(data))
remove_list = []
for j in range(num_bins):
    list_ = []
    for i in range(len(data['steering'])):
        if data['steering'][i] >= bins[j] and data['steering'][i] <= bins[j+1]:
            list_.append(i)
    list_ = shuffle(list_)
    list_ = list_[samples_per_bin:]
    remove_list.extend(list_)

print('Removed:',len(remove_list))
data.drop(data.index[remove_list],inplace=True)
print('Remaining:',len(data))

hist,_ = np.histogram(data['steering'],num_bins)
plt.bar(center,hist,width=0.05)
plt.plot((np.min(data['steering']),np.max(data['steering'])),(samples_per_bin,samples_per_bin))
```

Total Data: 4053  
 Removed: 2790  
 Remaining: 1263  
 [<matplotlib.lines.Line2D at 0x7fa94454bbd0>]



```
print(data.iloc[1])
def load_img_steering(datadir,df):
    image_path = []
    steering = []
    for i in range(len(data)):
        indexed_data = data.iloc[i]
        center,left,right = indexed_data[0],indexed_data[1],indexed_data[2]
        image_path.append(os.path.join(datadir,center.strip()))
        steering.append(float(indexed_data[3]))
    image_paths = np.asarray(image_path)
    steerings = np.asarray(steering)
    return image_paths, steerings
```

```
image_paths,steerings = load_img_steering(datadir + '/IMG',data)
```

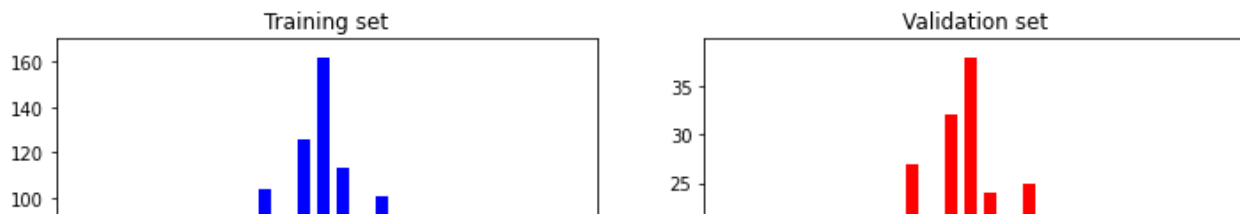
```
center      center_2018_07_16_17_11_44_413.jpg
left        left_2018_07_16_17_11_44_413.jpg
right       right_2018_07_16_17_11_44_413.jpg
steering     -0.05
throttle     0.642727
reverse      0.0
speed        1.434013
Name: 12, dtype: object
```

```
X_train,X_valid,Y_train,Y_valid = train_test_split(image_paths,steerings,test_size=0.2,random_state=
print('Training Samples: {}\nValid Samples: {}'.format(len(X_train),len(X_valid)))
```

```
Training Samples: 1010
Valid Samples: 253
```

```
fig,axes = plt.subplots(1,2,figsize=(12,4))
axes[0].hist(Y_train,bins=num_bins,width=0.05,color='blue')
axes[0].set_title('Training set')
axes[1].hist(Y_valid,bins=num_bins,width=0.05,color='red')
axes[1].set_title('Validation set')
```

Text(0.5, 1.0, 'Validation set')

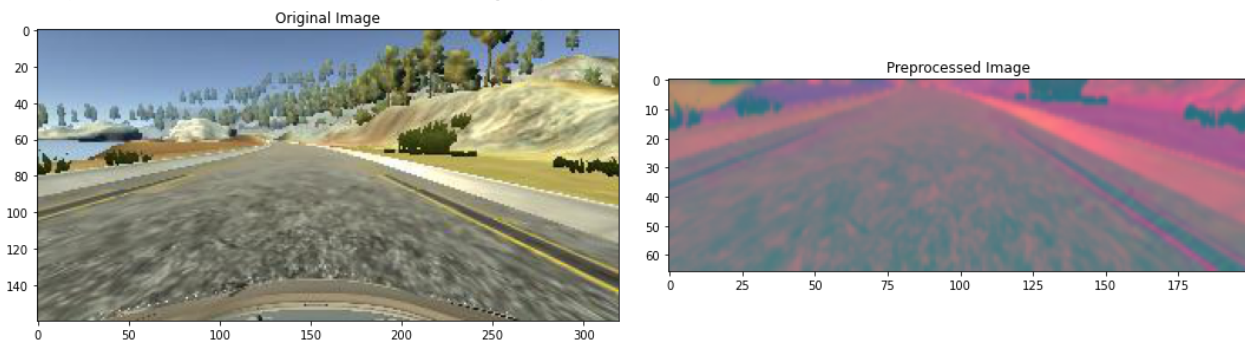


```
def img_preprocess(img):  
    img = npimg.imread(img)  
    img = img[60:135,:,:]  
    img = cv2.cvtColor(img,cv2.COLOR_RGB2YUV)  
    img = cv2.GaussianBlur(img,(3,3),0)  
    img = cv2.resize(img,(200,66))  
    img = img/255  
    return img
```

```
image = image_paths[100]  
original_image = npimg.imread(image)  
preprocessed_image = img_preprocess(image)
```

```
fig,axs = plt.subplots(1,2,figsize=(15,10))  
fig.tight_layout()  
axs[0].imshow(original_image)  
axs[0].set_title('Original Image')  
axs[1].imshow(preprocessed_image)  
axs[1].set_title('Preprocessed Image')
```

Text(0.5, 1.0, 'Preprocessed Image')



```
X_train = np.array(list(map(img_preprocess,X_train)))  
X_valid = np.array(list(map(img_preprocess,X_valid)))
```

```
plt.imshow(X_train[random.randint(0,len(X_train)-1)])  
plt.axis('off')  
print(X_train.shape)
```

(1010, 66, 200, 3)



```
from keras.layers.convolutional import Convolution2D
from keras import activations
def nvidia_model():
    model = Sequential()
    model.add(Conv2D(24,(5,5), strides=(2, 2), input_shape=(66, 200, 3), activation='elu'))
    model.add(Conv2D(36, (5,5), strides=(2, 2), activation='elu'))
    model.add(Conv2D(48, (5,5), strides=(2, 2), activation='elu'))
    model.add(Conv2D(64, (3,3), activation='elu'))
    model.add(Conv2D(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(learning_rate=1e-3)
    model.compile(loss='mse', optimizer=optimizer)
    return model
```

```
model = nvidia_model()
print(model.summary())
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
=====		
conv2d_5 (Conv2D)	(None, 31, 98, 24)	1824
conv2d_6 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_7 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_8 (Conv2D)	(None, 3, 20, 64)	27712
conv2d_9 (Conv2D)	(None, 1, 18, 64)	36928
dropout_4 (Dropout)	(None, 1, 18, 64)	0
flatten_1 (Flatten)	(None, 1152)	0
dense_4 (Dense)	(None, 100)	115300
dropout_5 (Dropout)	(None, 100)	0
dense_5 (Dense)	(None, 50)	5050
dropout_6 (Dropout)	(None, 50)	0
dense_6 (Dense)	(None, 10)	510
dropout_7 (Dropout)	(None, 10)	0

```
=====
Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0
```

---

None

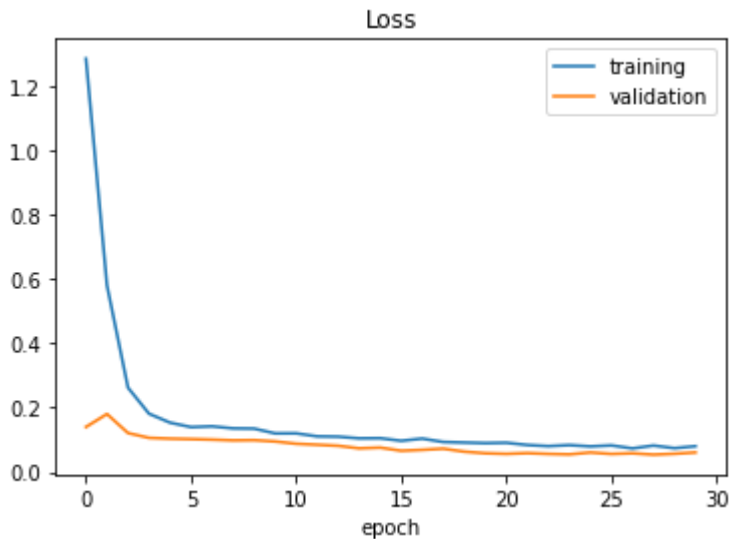
```
history = model.fit(X_train,Y_train,epochs=30,validation_data=(X_valid,Y_valid),batch_size=100,verbo
```

```
Epoch 1/30
11/11 [=====] - 1s 53ms/step - loss: 1.2854 - val_loss: 0.1399
Epoch 2/30
11/11 [=====] - 0s 27ms/step - loss: 0.5805 - val_loss: 0.1804
Epoch 3/30
11/11 [=====] - 0s 24ms/step - loss: 0.2623 - val_loss: 0.1212
Epoch 4/30
11/11 [=====] - 0s 23ms/step - loss: 0.1813 - val_loss: 0.1061
Epoch 5/30
11/11 [=====] - 0s 22ms/step - loss: 0.1534 - val_loss: 0.1037
Epoch 6/30
11/11 [=====] - 0s 23ms/step - loss: 0.1397 - val_loss: 0.1023
Epoch 7/30
11/11 [=====] - 0s 23ms/step - loss: 0.1420 - val_loss: 0.1007
Epoch 8/30
11/11 [=====] - 0s 23ms/step - loss: 0.1356 - val_loss: 0.0983
Epoch 9/30
11/11 [=====] - 0s 23ms/step - loss: 0.1349 - val_loss: 0.0990
Epoch 10/30
11/11 [=====] - 0s 22ms/step - loss: 0.1201 - val_loss: 0.0949
Epoch 11/30
11/11 [=====] - 0s 22ms/step - loss: 0.1202 - val_loss: 0.0878
Epoch 12/30
11/11 [=====] - 0s 23ms/step - loss: 0.1109 - val_loss: 0.0848
Epoch 13/30
11/11 [=====] - 0s 23ms/step - loss: 0.1099 - val_loss: 0.0815
Epoch 14/30
11/11 [=====] - 0s 23ms/step - loss: 0.1046 - val_loss: 0.0735
Epoch 15/30
11/11 [=====] - 0s 24ms/step - loss: 0.1051 - val_loss: 0.0758
Epoch 16/30
11/11 [=====] - 0s 29ms/step - loss: 0.0970 - val_loss: 0.0660
Epoch 17/30
11/11 [=====] - 0s 28ms/step - loss: 0.1038 - val_loss: 0.0689
Epoch 18/30
11/11 [=====] - 0s 29ms/step - loss: 0.0933 - val_loss: 0.0725
Epoch 19/30
11/11 [=====] - 0s 26ms/step - loss: 0.0915 - val_loss: 0.0632
Epoch 20/30
11/11 [=====] - 0s 26ms/step - loss: 0.0899 - val_loss: 0.0583
Epoch 21/30
11/11 [=====] - 0s 29ms/step - loss: 0.0909 - val_loss: 0.0564
Epoch 22/30
11/11 [=====] - 0s 24ms/step - loss: 0.0840 - val_loss: 0.0587
Epoch 23/30
11/11 [=====] - 0s 23ms/step - loss: 0.0805 - val_loss: 0.0562
Epoch 24/30
11/11 [=====] - 0s 22ms/step - loss: 0.0837 - val_loss: 0.0548
Epoch 25/30
11/11 [=====] - 0s 22ms/step - loss: 0.0796 - val_loss: 0.0607
Epoch 26/30
```

```
11/11 [=====] - 0s 23ms/step - loss: 0.0827 - val_loss: 0.0562
Epoch 27/30
11/11 [=====] - 0s 22ms/step - loss: 0.0734 - val_loss: 0.0582
Epoch 28/30
11/11 [=====] - 0s 23ms/step - loss: 0.0823 - val_loss: 0.0540
Epoch 29/30
11/11 [=====] - 0s 26ms/step - loss: 0.0741 - val_loss: 0.0567
```

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training', 'validation'])
plt.title('Loss')
plt.xlabel('epoch')
```

Text(0.5, 0, 'epoch')



```
model.save('model.h5')
```

```
from google.colab import files
files.download('model.h5')
```