

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

```
fatal: destination path 'track' already exists and is not an empty directory.
```

```
!ls track
```

```
driving_log.csv  IMG
```

```
!pip3 install imgaug
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: imgaug in /usr/local/lib/python3.7/dist-packages (0.4.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.15.0)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.21.6)
Requirement already satisfied: scikit-image>=0.14.2 in /usr/local/lib/python3.7/dist-packages (from imgaug) (0.18.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.7.3)
Requirement already satisfied: Shapely in /usr/local/lib/python3.7/dist-packages (from imgaug) (1.8.2)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from imgaug) (7.1.2)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-packages (from imgaug) (4.6.0.66)
Requirement already satisfied: imageio in /usr/local/lib/python3.7/dist-packages (from imgaug) (2.9.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from imgaug) (3.2.2)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (2.6.3)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (1.3.0)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image>=0.14.2->imgaug) (2021.11)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (1.4.4)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgau
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->imgaug) (0.11.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib->imgaug) (
```

```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as npimg
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
from imgaug import augmenters as iaa
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: Passing a negative integer is deprecated in version 1.0 and after removing the cwd from sys.path.

	center	left	right
0	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_382.jpg	C:\Users\Amer\Desktop\new_track\IMG\left_2018_07_16_17_11_43_382.jpg	C:\Users\Amer\Desktop\new_track\IMG\right_2018_07_16_17_11_43_382.jpg
1	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_670.jpg	C:\Users\Amer\Desktop\new_track\IMG\left_2018_07_16_17_11_43_670.jpg	C:\Users\Amer\Desktop\new_track\IMG\right_2018_07_16_17_11_43_670.jpg
2	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_724.jpg	C:\Users\Amer\Desktop\new_track\IMG\left_2018_07_16_17_11_43_724.jpg	C:\Users\Amer\Desktop\new_track\IMG\right_2018_07_16_17_11_43_724.jpg
3	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_792.jpg	C:\Users\Amer\Desktop\new_track\IMG\left_2018_07_16_17_11_43_792.jpg	C:\Users\Amer\Desktop\new_track\IMG\right_2018_07_16_17_11_43_792.jpg
4	C:\Users\Amer\Desktop\new_track\IMG\center_2018_07_16_17_11_43_860.jpg	C:\Users\Amer\Desktop\new_track\IMG\left_2018_07_16_17_11_43_860.jpg	C:\Users\Amer\Desktop\new_track\IMG\right_2018_07_16_17_11_43_860.jpg



```
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)

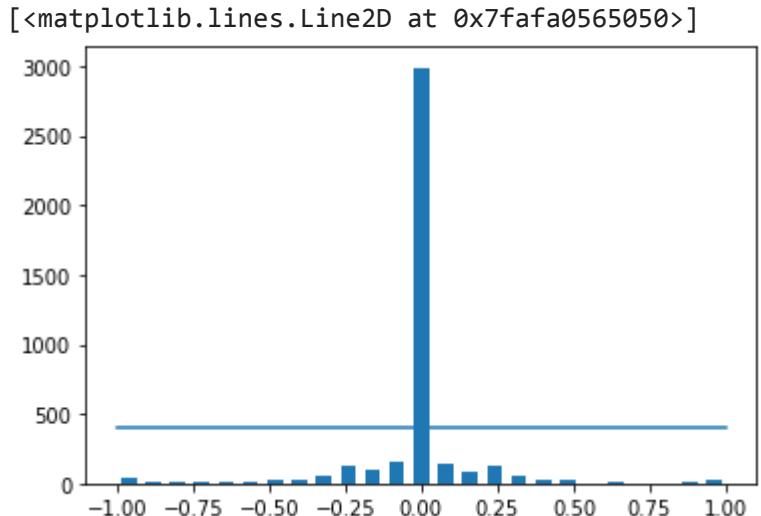
data.head()
```

	center	left	right	steering	throttle	reverse	spee
0	center_2018_07_16_17_11_43_382.jpg	left_2018_07_16_17_11_43_382.jpg	right_2018_07_16_17_11_43_382.jpg	0.0	0.0	0.0	0.64978
1	center_2018_07_16_17_11_43_670.jpg	left_2018_07_16_17_11_43_670.jpg	right_2018_07_16_17_11_43_670.jpg	0.0	0.0	0.0	0.62794
2	center_2018_07_16_17_11_43_724.jpg	left_2018_07_16_17_11_43_724.jpg	right_2018_07_16_17_11_43_724.jpg	0.0	0.0	0.0	0.62291
3	center_2018_07_16_17_11_43_792.jpg	left_2018_07_16_17_11_43_792.jpg	right_2018_07_16_17_11_43_792.jpg	0.0	0.0	0.0	0.61916
4	center_2018_07_16_17_11_43_860.jpg	left_2018_07_16_17_11_43_860.jpg	right_2018_07_16_17_11_43_860.jpg	0.0	0.0	0.0	0.61543

```

num_bins = 25
samples_per_bin = 400
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))

```



```

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_)

```

```

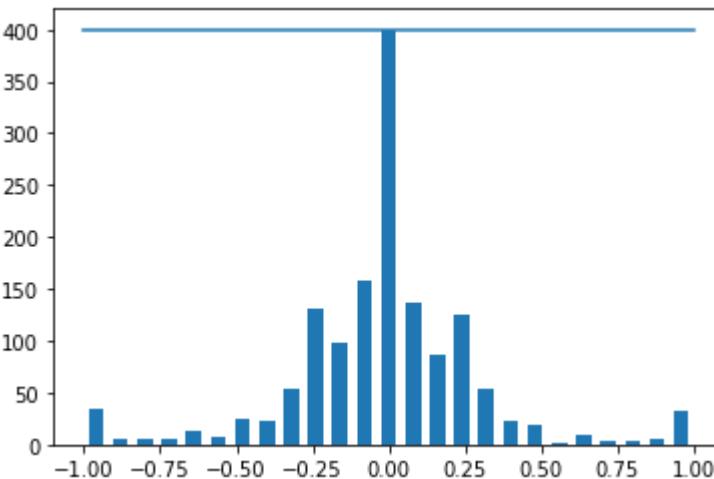
list_ = list_[samples_per_bin:]
remove_list.extend(list_)

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: 2590
 Remaining: 1463
 [`<matplotlib.lines.Line2D at 0x7fafaf043d9d0>`]



```

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)
```

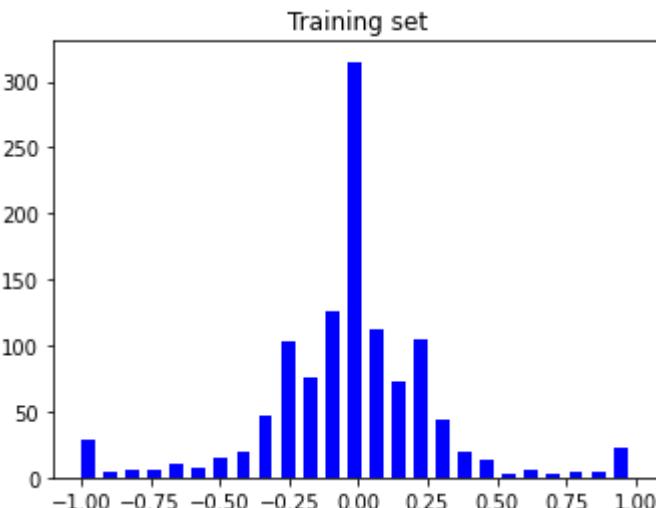
```
center      center_2018_07_16_17_11_44_413.jpg
left        left_2018_07_16_17_11_44_413.jpg
right       right_2018_07_16_17_11_44_413.jpg
steering    -0.05
throttle   0.642727
reverse     0.0
speed      1.434013
Name: 12, dtype: object
```

```
X_train,X_valid,Y_train,Y_valid = train_test_split(image_paths,steerings,test_size=0.2,random_state=6)
print('Training Samples: {} \nValid Samples: {}'.format(len(X_train),len(X_valid)))
```

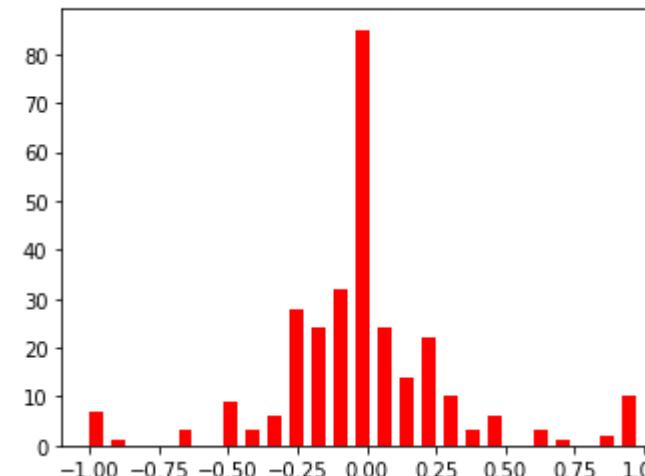
```
Training Samples: 1170
Valid Samples: 293
```

```
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')
```



Validation set



```
def zoom(image):
    zoom = iaa.Affine(scale=(1, 1.3))
    image = zoom.augment_image(image)
    return image
```

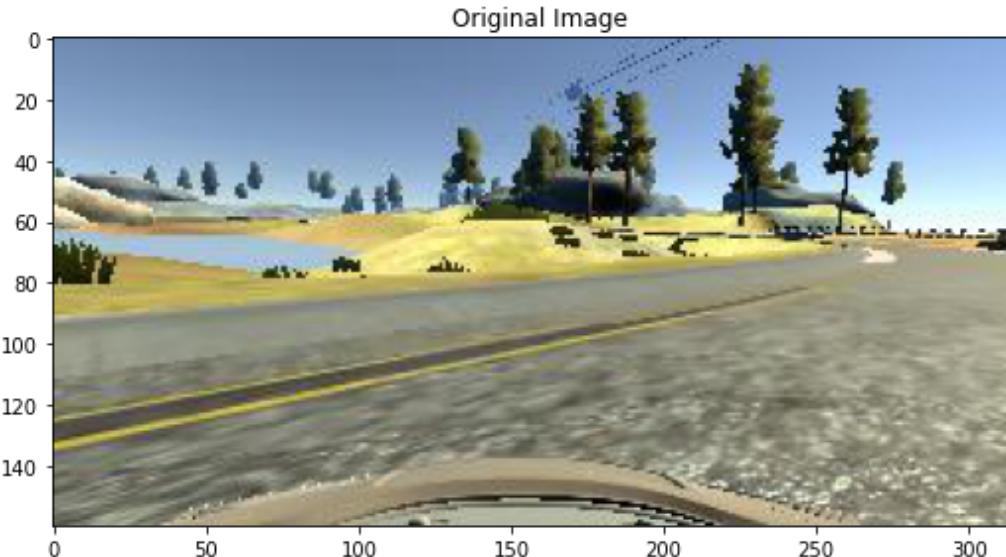
```
image = image_paths[random.randint(0, 1000)]
original_image = npimg.imread(image)
zoomed_image = zoom(original_image)

fig, axs = plt.subplots(1, 2, figsize=(15, 10))
fig.tight_layout()

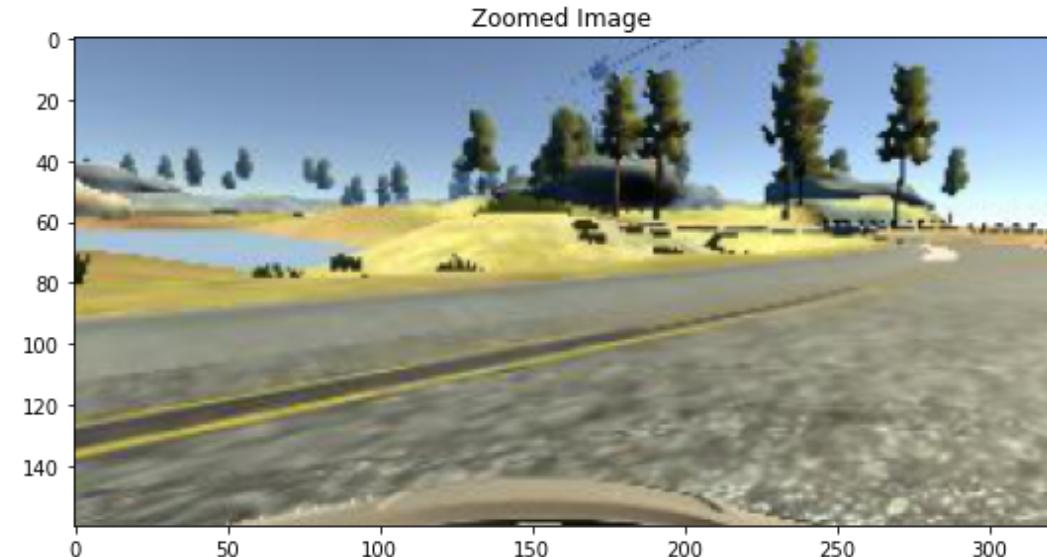
axs[0].imshow(original_image)
axs[0].set_title('Original Image')

axs[1].imshow(zoomed_image)
axs[1].set_title('Zoomed Image')
```

```
Text(0.5, 1.0, 'Zoomed Image')
```



Original Image



Zoomed Image

```
def pan(image):
    pan = iaa.Affine(translate_percent= {"x" : (-0.1, 0.1), "y": (-0.1, 0.1)})
```

```

image = pan.augment_image(image)
return image

image = image_paths[random.randint(0, 1000)]
original_image = npimg.imread(image)
panned_image = pan(original_image)

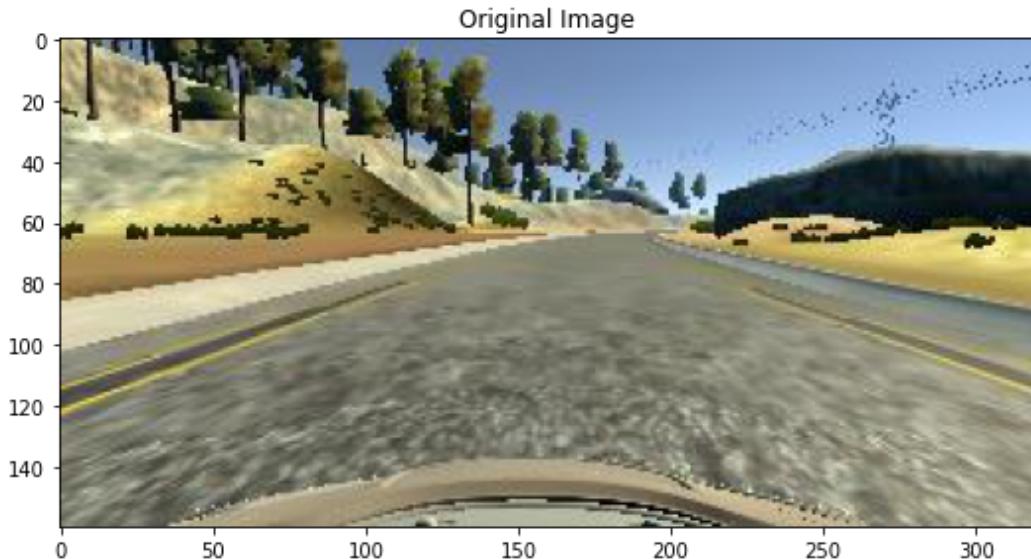
fig, axs = plt.subplots(1, 2, figsize=(15, 10))
fig.tight_layout()

axs[0].imshow(original_image)
axs[0].set_title('Original Image')

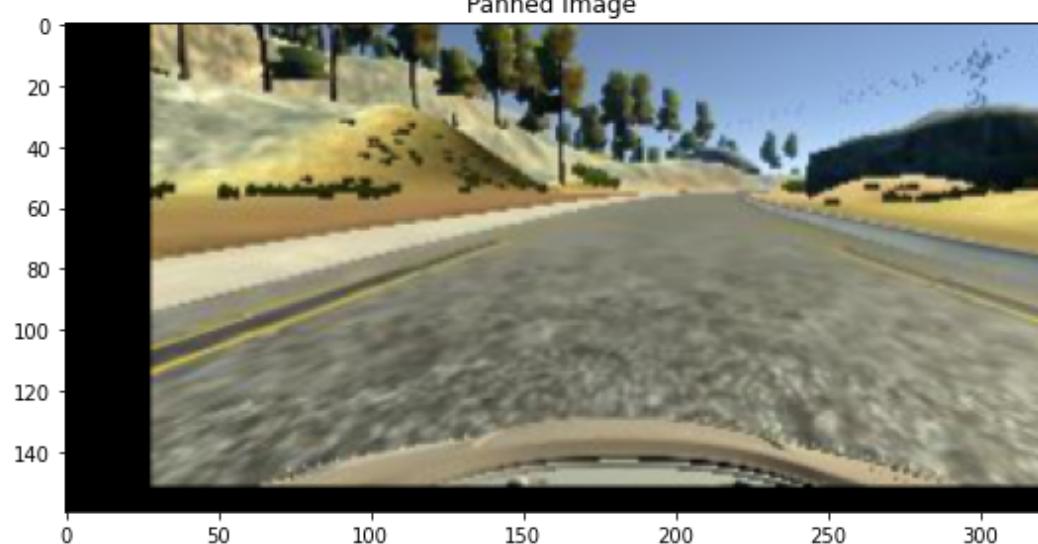
axs[1].imshow(panned_image)
axs[1].set_title('Panned Image')

```

Text(0.5, 1.0, 'Panned Image')



Panned Image



```

def img_random_brightness(image):
    brightness = iaa.Multiply((0.2, 1.2))
    image = brightness.augment_image(image)
    return image

```

```

image = image_paths[random.randint(0, 1000)]
original_image = npimg.imread(image)
brightness_altered_image = img_random_brightness(original_image)

```

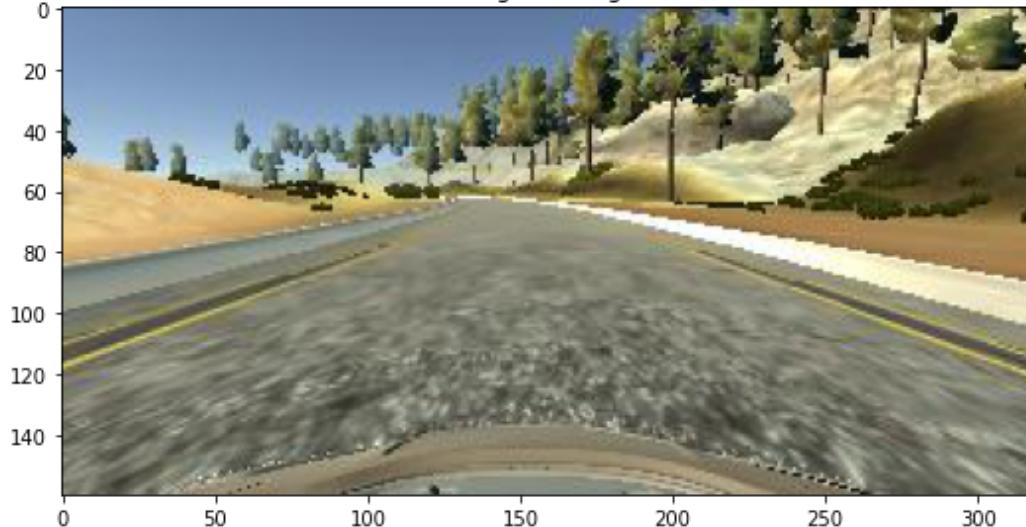
```
fig, axs = plt.subplots(1, 2, figsize=(15, 10))
fig.tight_layout()
```

```
axs[0].imshow(original_image)
axs[0].set_title('Original Image')
```

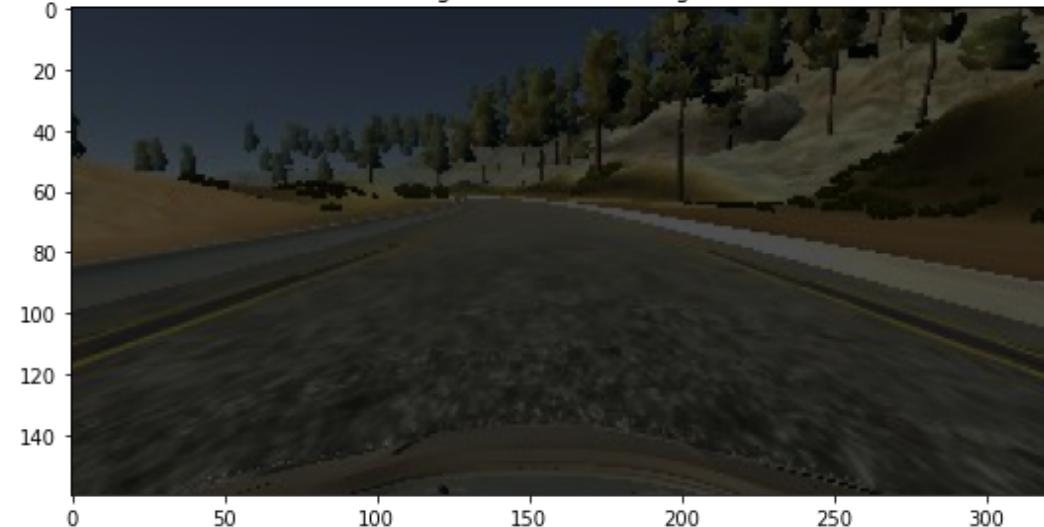
```
axs[1].imshow(brightness_altered_image)
axs[1].set_title('Brightness altered image ')
```

```
Text(0.5, 1.0, 'Brightness altered image ')
```

Original Image



Brightness altered image



```
def img_random_flip(image, steering_angle):
    image = cv2.flip(image,1)
    steering_angle = -steering_angle
    return image, steering_angle
```

```
random_index = random.randint(0, 1000)
image = image_paths[random_index]
steering_angle = steerings[random_index]
```

```
original_image = npimg.imread(image)
flipped_image, flipped_steering_angle = img_random_flip(original_image, steering_angle)
```

```

fig, axs = plt.subplots(1, 2, figsize=(15, 10))
fig.tight_layout()

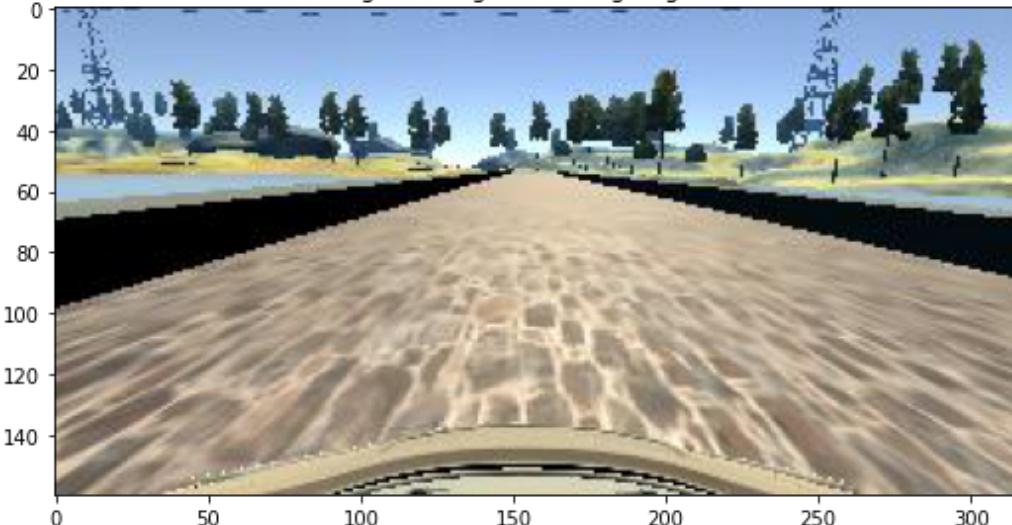
axs[0].imshow(original_image)
axs[0].set_title('Original Image - ' + 'Steering Angle:' + str(steering_angle))

axs[1].imshow(flipped_image)
axs[1].set_title('Flipped Image - ' + 'Steering Angle:' + str(flipped_steering_angle))

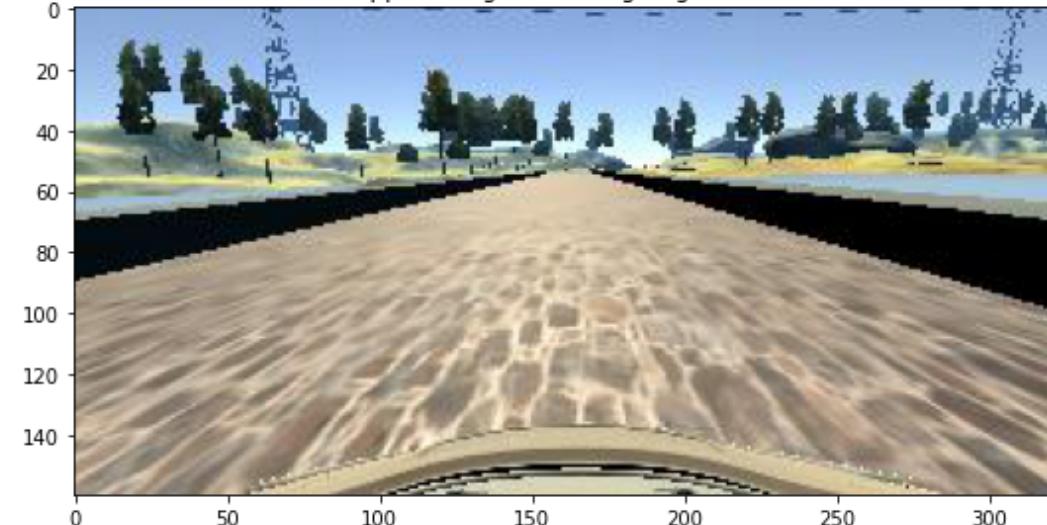
```

Text(0.5, 1.0, 'Flipped Image - Steering Angle:-0.05')

Original Image - Steering Angle:0.05



Flipped Image - Steering Angle:-0.05



```

def random_augment(image, steering_angle):
    image = npimg.imread(image)
    if np.random.rand() < 0.5:
        image = pan(image)
    if np.random.rand() < 0.5:
        image = zoom(image)
    if np.random.rand() < 0.5:
        image = img_random_brightness(image)
    if np.random.rand() < 0.5:
        image, steering_angle = img_random_flip(image, steering_angle)
    return image, steering_angle

```

```

ncol = 2
nrow = 10

```

```
fig, axs = plt.subplots(nrows, ncols, figsize=(15, 50))
fig.tight_layout()

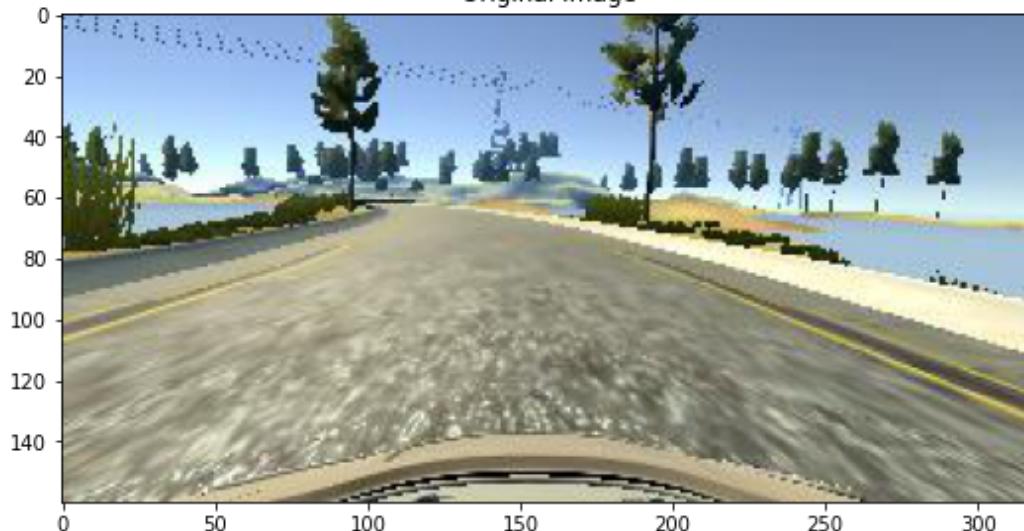
for i in range(10):
    randnum = random.randint(0, len(image_paths) - 1)
    random_image = image_paths[randnum]
    random_steering = steerings[randnum]

    original_image = npimg.imread(random_image)
    augmented_image, steering = random_augment(random_image, random_steering)

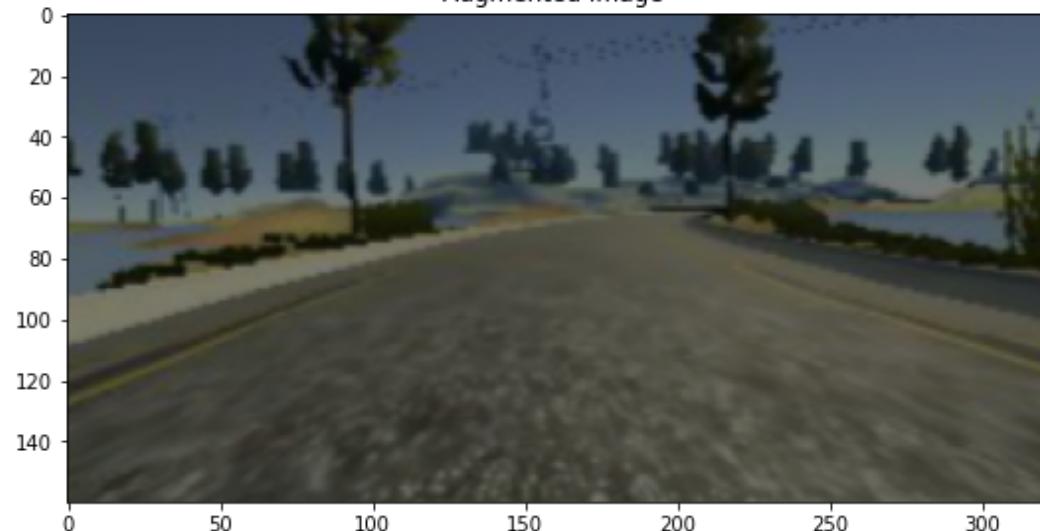
    axs[i][0].imshow(original_image)
    axs[i][0].set_title("Original Image")

    axs[i][1].imshow(augmented_image)
    axs[i][1].set_title("Augmented Image")
```

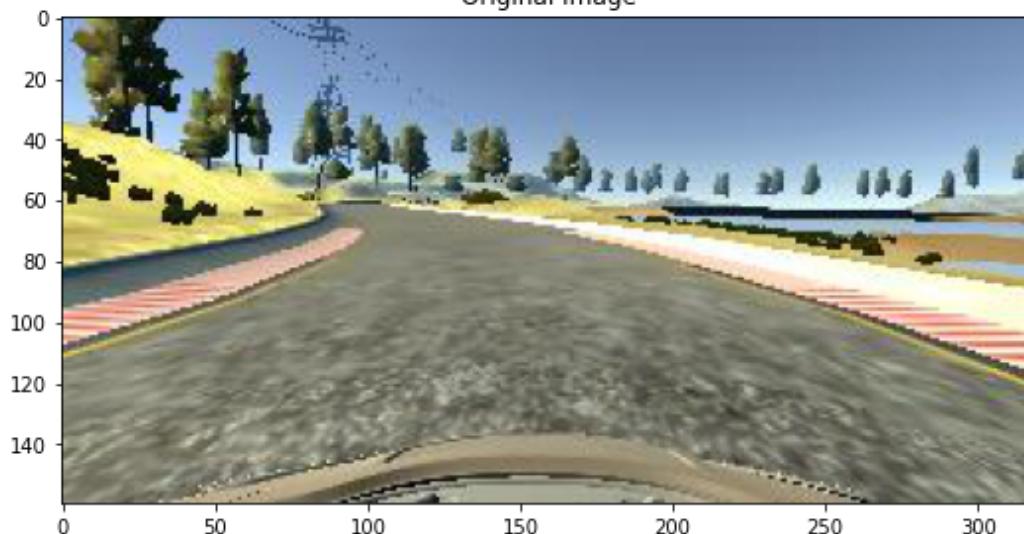
Original Image



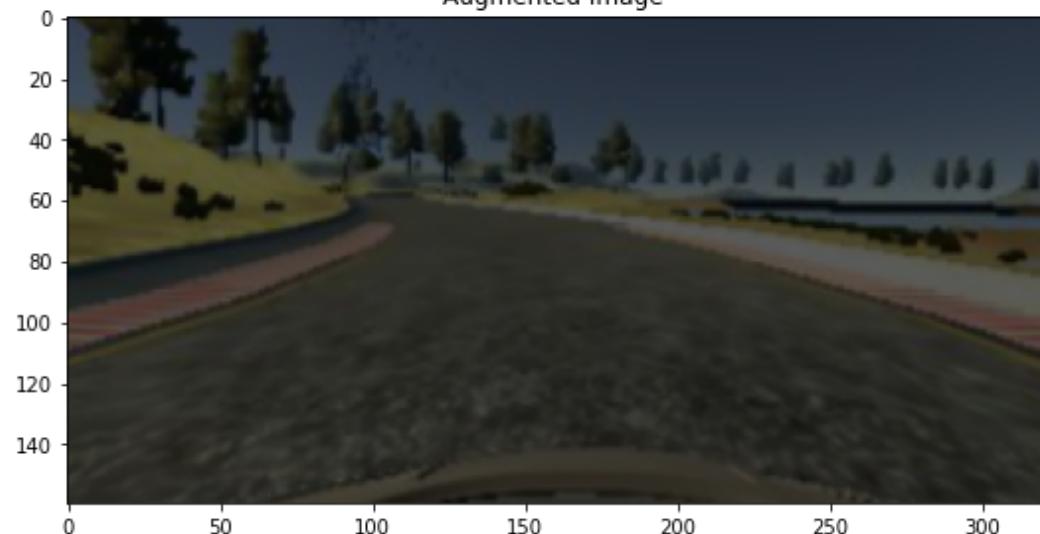
Augmented Image



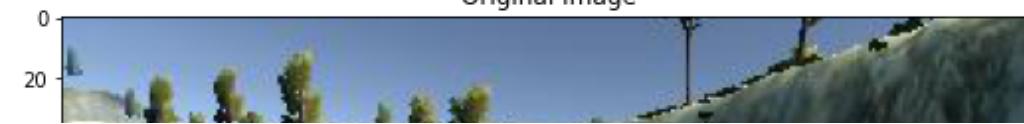
Original Image



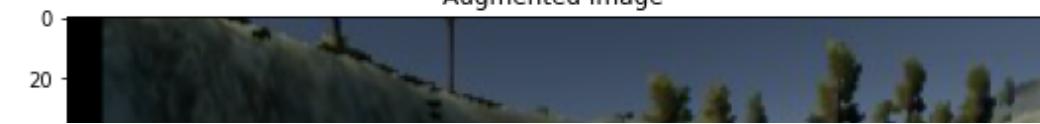
Augmented Image

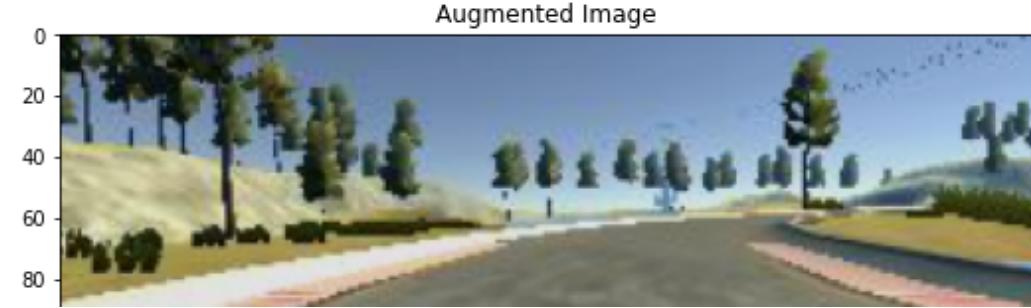
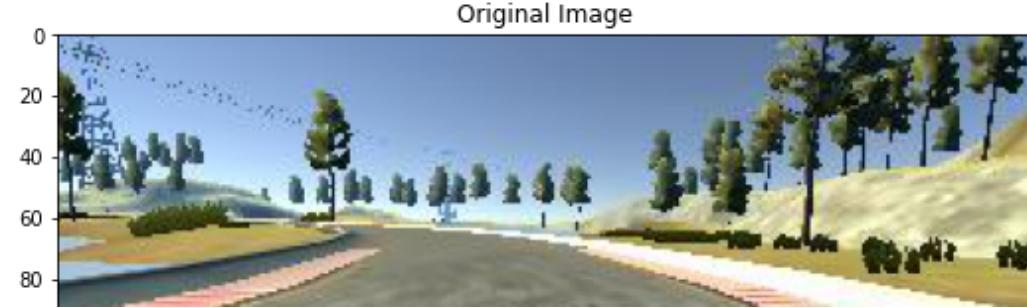
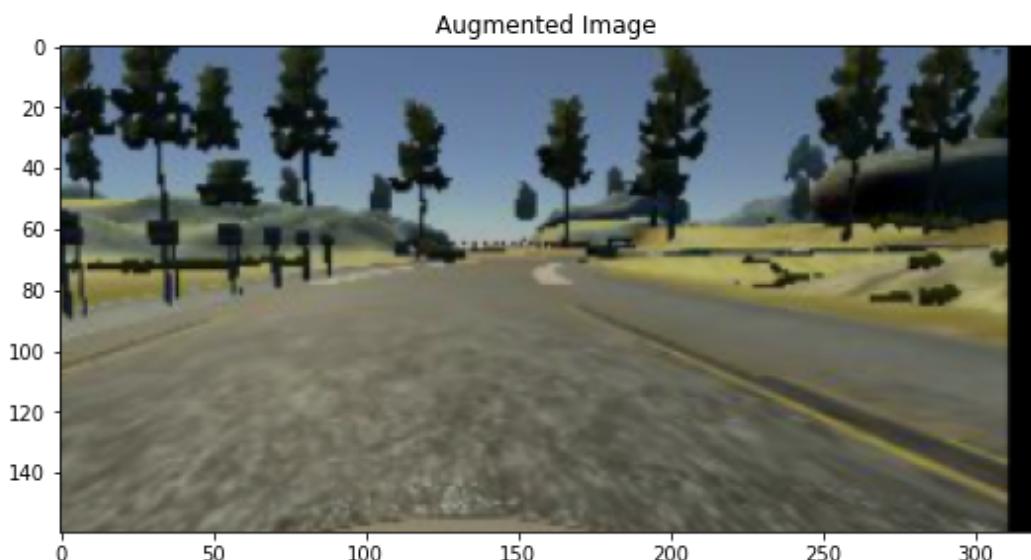
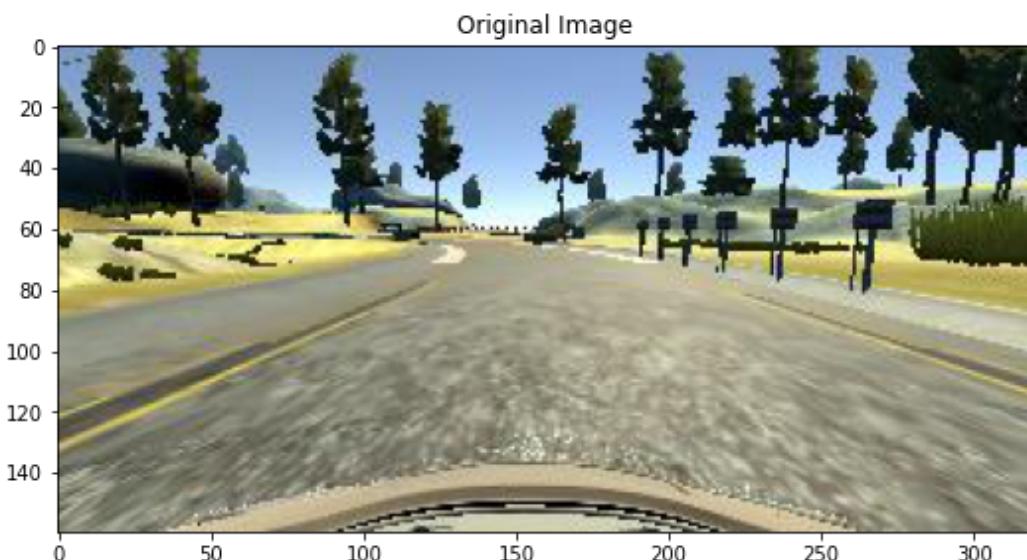
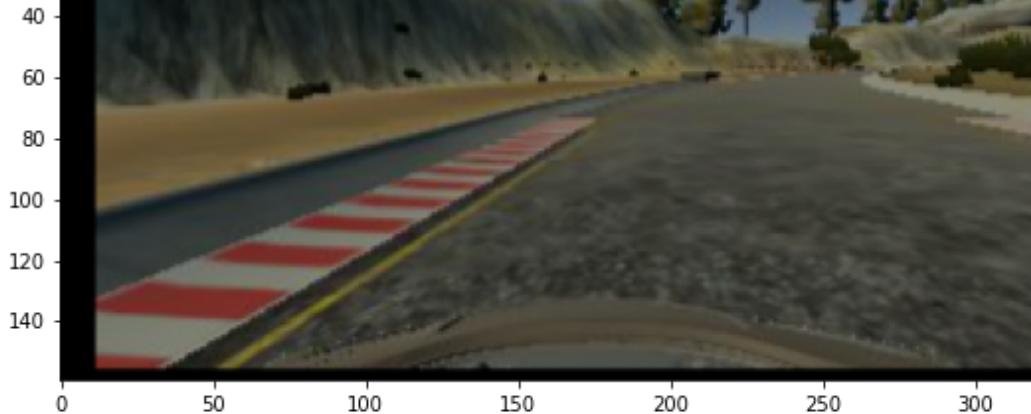
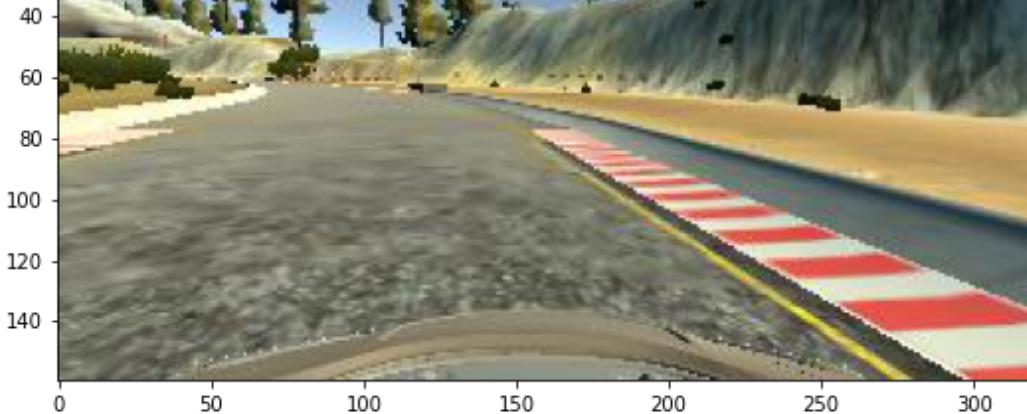


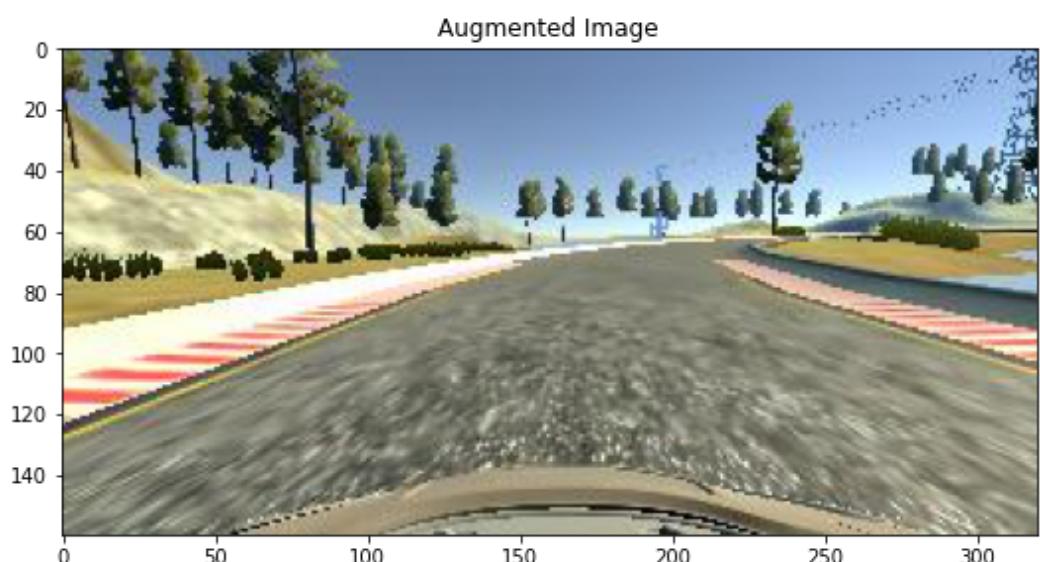
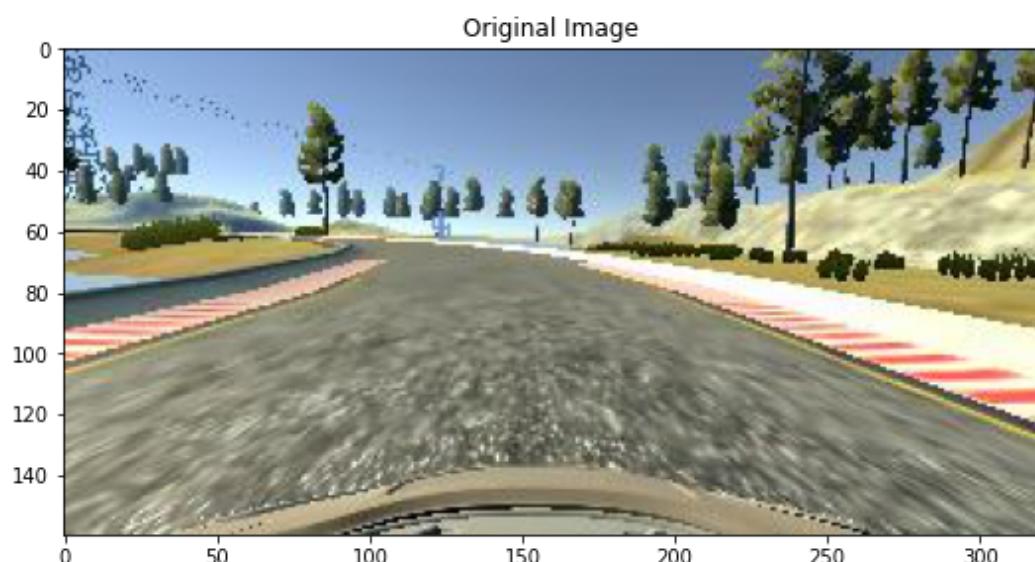
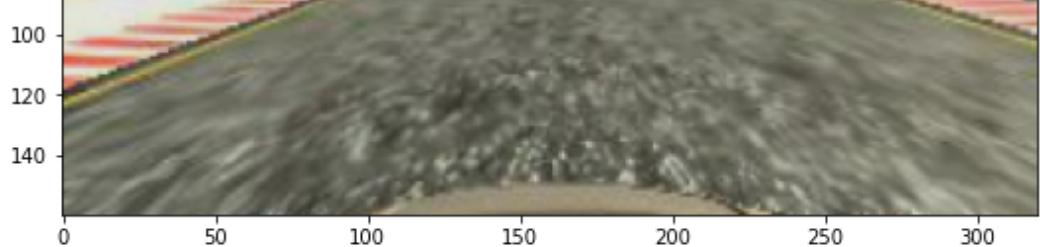
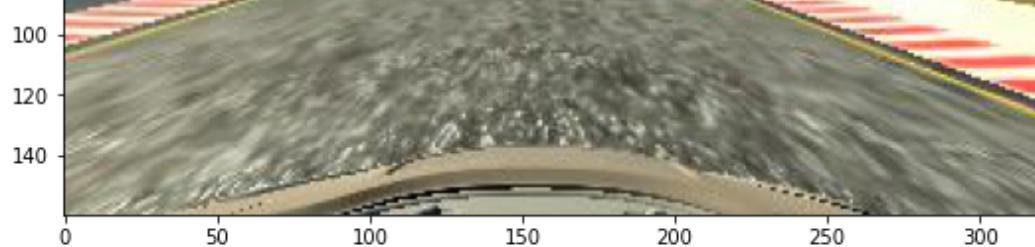
Original Image

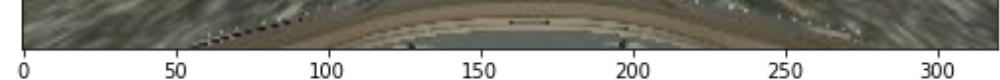
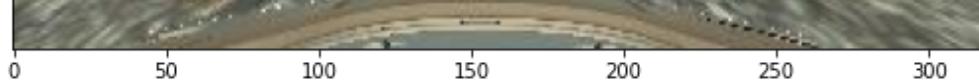


Augmented Image

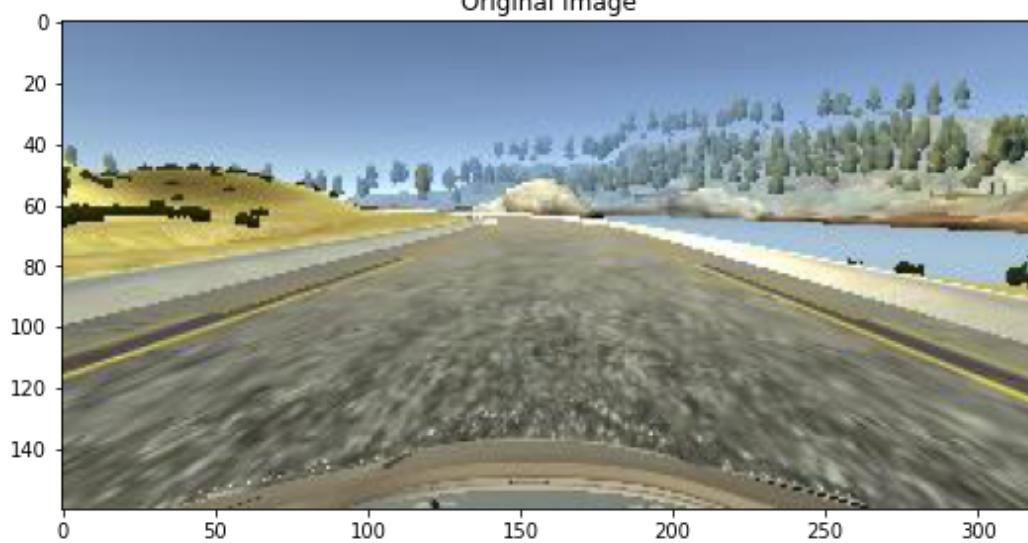




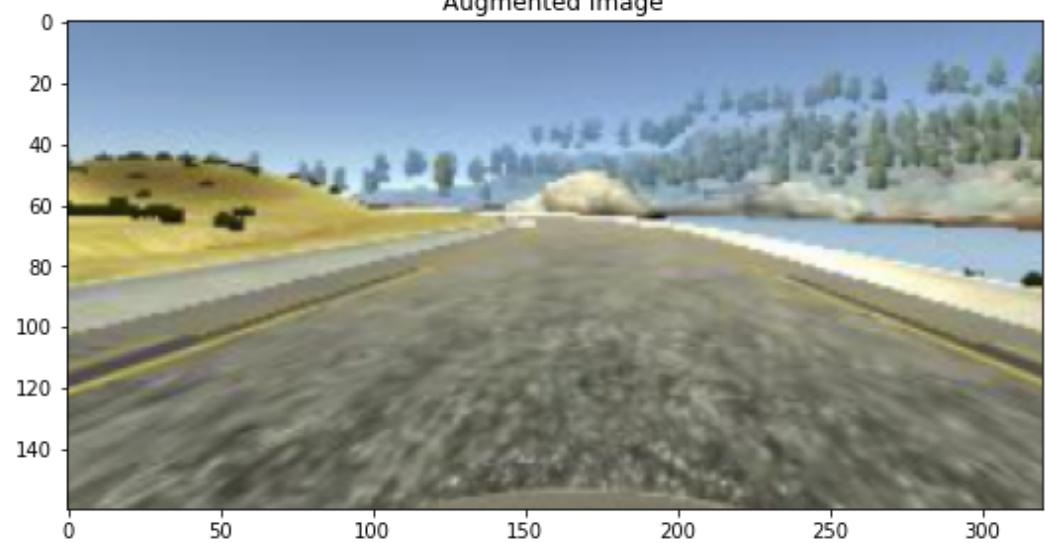




Original Image



Augmented Image



Original Image



Augmented Image



```
def img_preprocess(img):
    img = img[60:135,:,:]
    img = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
```

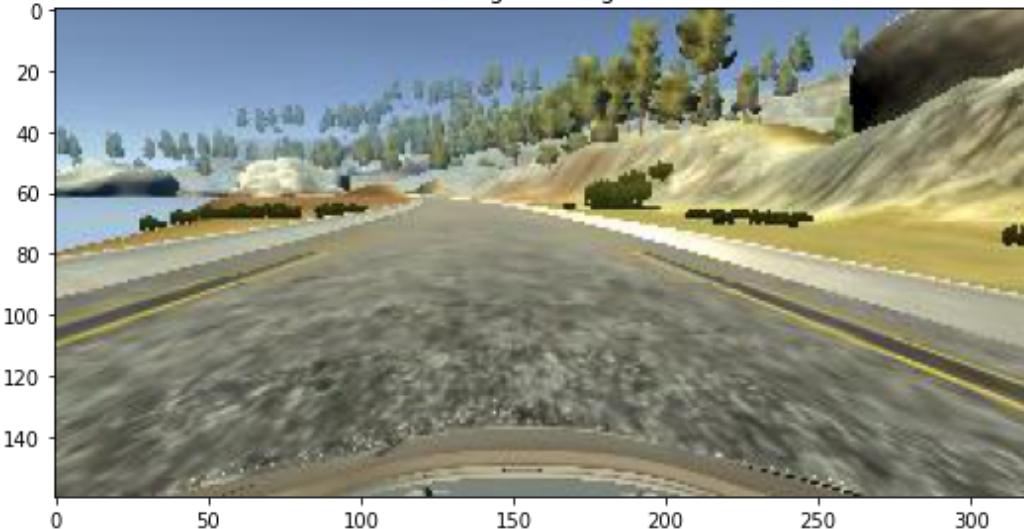
```
img = cv2.GaussianBlur(img,(3,3),0)
img = cv2.resize(img,(200,66))
img = img/255
return img
```

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

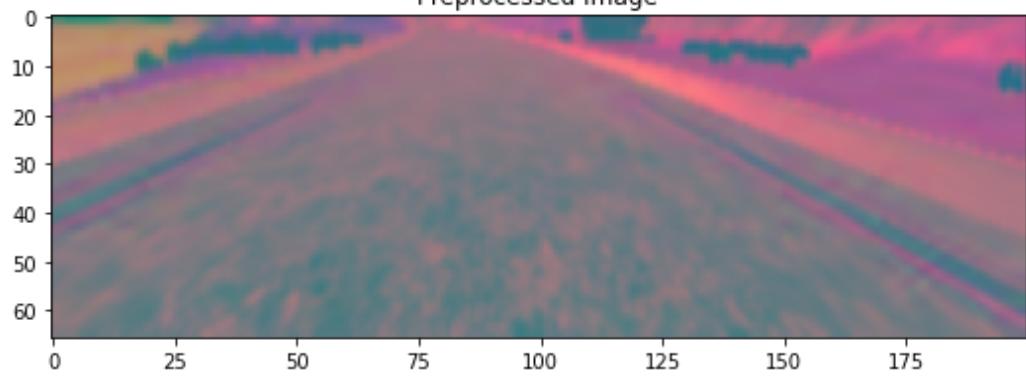
```
fig,axs = plt.subplots(1,2,figsize=(15,10))
fig.tight_layout()
axs[0].imshow(original_image)
axs[0].set_title('Original Image')
axs[1].imshow(preprocessed_image)
axs[1].set_title('Preprocessed Image')
```

Text(0.5, 1.0, 'Preprocessed Image')

Original Image



Preprocessed Image



```
def batch_generator(image_paths, steering_ang, batch_size, istraining):

while True:
    batch_img = []
    batch_steering = []

    for i in range(batch_size):
```

```

random_index = random.randint(0, len(image_paths) - 1)

ifistraining:
    im, steering = random_augment(image_paths[random_index], steering_ang[random_index])

else:
    im = npimg.imread(image_paths[random_index])
    steering = steering_ang[random_index]

im = img_preprocess(im)
batch_img.append(im)
batch_steering.append(steering)
yield (np.asarray(batch_img), np.asarray(batch_steering))

```

```

x_train_gen,y_train_gen = next(batch_generator(X_train,Y_train,1,1))
x_valid_gen,y_valid_gen = next(batch_generator(X_valid,Y_valid,1,0))

```

```

fig, axs = plt.subplots(1, 2, figsize=(15, 10))
fig.tight_layout()

```

```

axs[0].imshow(x_train_gen[0])
axs[0].set_title('Training Image')

```

```

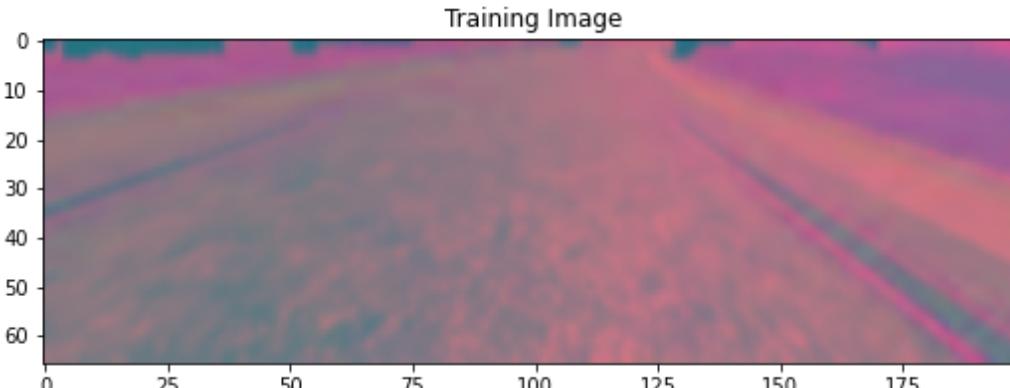
axs[1].imshow(x_valid_gen[0])
axs[1].set_title('Validation Image')

```

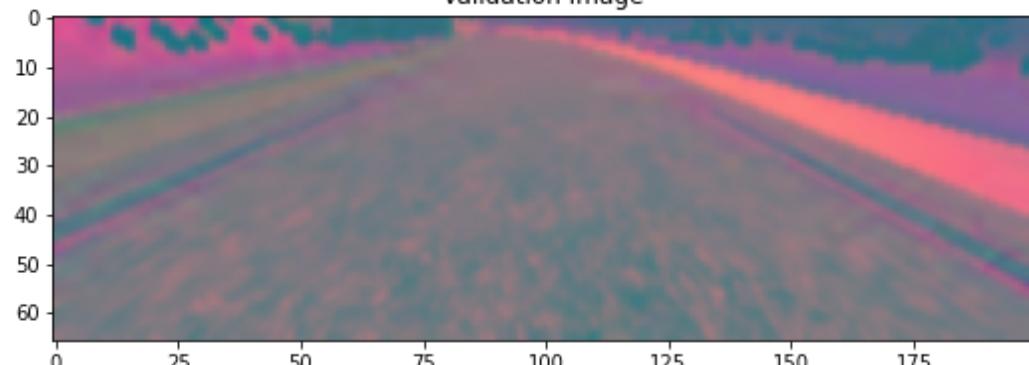
```

Text(0.5, 1.0, 'Validation Image')

```



Validation Image



```

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(Flatten())
model.add(Dense(100, activation = 'elu'))
model.add(Dense(50, activation = 'elu'))
model.add(Dense(10, activation = 'elu'))
model.add(Dense(1))

optimizer = Adam(learning_rate=1e-4)
model.compile(loss='mse', optimizer=optimizer)
return model

```

```

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

```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 31, 98, 24)	1824
conv2d_1 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_2 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_3 (Conv2D)	(None, 3, 20, 64)	27712
conv2d_4 (Conv2D)	(None, 1, 18, 64)	36928
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 100)	115300
dense_1 (Dense)	(None, 50)	5050
dense_2 (Dense)	(None, 10)	510
dense_3 (Dense)	(None, 1)	11

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

```
None
```

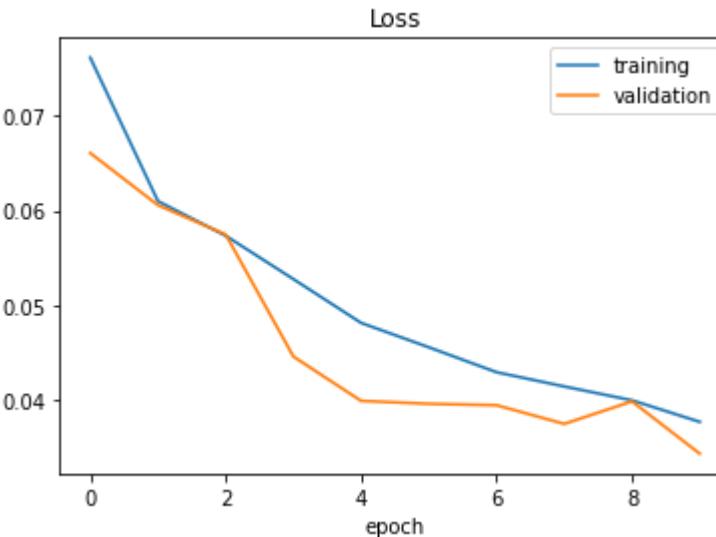
```
history = model.fit_generator(batch_generator(X_train,Y_train,100,1),
                               steps_per_epoch=300,
                               epochs=10,
                               validation_data=batch_generator(X_valid,Y_valid,100,0),
                               validation_steps=200,
                               verbose=1,
                               shuffle=1)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: UserWarning: `Model.fit_generator` is deprecated and will be removed in a
    import sys
Epoch 1/10
300/300 [=====] - 155s 477ms/step - loss: 0.0761 - val_loss: 0.0661
Epoch 2/10
300/300 [=====] - 126s 421ms/step - loss: 0.0610 - val_loss: 0.0606
Epoch 3/10
300/300 [=====] - 127s 425ms/step - loss: 0.0574 - val_loss: 0.0575
Epoch 4/10
300/300 [=====] - 127s 424ms/step - loss: 0.0528 - val_loss: 0.0446
Epoch 5/10
300/300 [=====] - 126s 423ms/step - loss: 0.0482 - val_loss: 0.0399
Epoch 6/10
300/300 [=====] - 127s 426ms/step - loss: 0.0456 - val_loss: 0.0396
Epoch 7/10
300/300 [=====] - 127s 424ms/step - loss: 0.0430 - val_loss: 0.0395
Epoch 8/10
300/300 [=====] - 127s 426ms/step - loss: 0.0415 - val_loss: 0.0375
Epoch 9/10
300/300 [=====] - 126s 420ms/step - loss: 0.0400 - val_loss: 0.0399
Epoch 10/10
300/300 [=====] - 125s 419ms/step - loss: 0.0377 - val_loss: 0.0344
```

```
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('finalmodel.h5')
```

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

✓ 0s completed at 8:23 AM



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.