Behavioral Cloning for Udacity Drive Simulator

This Project was prepared for Udacity Self Driving Car ND - Term one. 12 Feb 2019

Notes:

Project Uses Nvidia model published in NVIDIA End To End Driving Paper - Thanks and Acknowledgements

There are two environments. One for Drive simulator Model building Environment (this file) and second environment is for simulation interface bridge for realtime simulation (drive.py)

I have used two different workstations, a mac and a tensorflow gpu on windows 10 pro Xeon W workstation on GPU 1080, unfortunately this environment broke down with pip issues when installing pip (12.0 - relased Feb 2019)along with eventlet and socket io which we use for drive.py. I switched between mac dual core mojave tensorflow cpu (conda based). I finished off in mac. Final movie is edited in apple imove. Music track is americal naitive pipe.

There are two ipnyb jupyter notebooks. One shows all the experiments from the start and this one is the second notebook where i finalised a model for final testing and submission

Date:12 Feb 2019

Change request for release : 1

Release no:

Prepared for : Udacity ND Term 1 BC

Prepared by : ARV

Design doc ref:

Release doc ref:

Tech doc ref:

Testing doc ref:

User Read me ref :

Admin and Dev read me ref:

In [45]:
```
!conda list
```

```
# packages in environment at /miniconda3/envs/udrivesimul:
#
# Name                    Version                   Build  Channel
absl-py                   0.7.0                    py36_1000    conda-forge
appnope                   0.1.0                py36hf537a9a_0
asn1crypto                0.24.0                   py36_1003    conda-forge
astor                     0.7.1                      py_0    conda-forge
backcall                  0.1.0                    py36_0
blas                      1.0                         mkl    anaconda
bleach                    3.1.0                    pypi_0    pypi
bzip2                     1.0.6                h1de35cc_1002    conda-forge
c-ares                    1.15.0               h1de35cc_1001    conda-forge
ca-certificates           2019.1.23                     0
cairo                     1.14.12              h9d4d9ac_1005    conda-forge
certifi                   2018.11.29               py36_0
cffi                      1.11.5           py36h342bebf_1001    conda-forge
click                     7.0                        py_0    conda-forge
cloudpickle               0.7.0                    pypi_0    pypi
cryptography              2.5                 py36ha12b0ac_0
cycler                    0.10.0                     py_1    conda-forge
dask                      1.1.1                    pypi_0    pypi
dbus                      1.13.6                h90a0687_0
decorator                 4.3.2                    py36_0
entrypoints               0.3                      py36_0
eventlet                  0.23.0                  py36_1000    conda-forge
expat                     2.2.6                 h0a44026_0
ffmpeg                    4.1                  heb45b42_1000    conda-forge
flask                     1.0.2                      py_2    conda-forge
fontconfig                2.13.1               h1e4e890_1000    conda-forge
freetype                  2.9.1                   hb4e5f40_0    anaconda
gast                      0.2.1.post0                py_0    conda-forge
gettext                   0.19.8.1             hcca000d_1001    conda-forge
giflib                    5.1.4                h1de35cc_1001    conda-forge
glib                      2.56.2               h67dad55_1001    conda-forge
gmp                       6.1.2                h0a44026_1000    conda-forge
gnutls                    3.6.5                h53004b3_1001    conda-forge
graphite2                 1.3.13               h2098e52_1000    conda-forge
greenlet                  0.4.13                   py36_0    conda-forge
grpcio                    1.16.1               py36h044775b_1
h5py                      2.8.0                py36h878fce3_3    anaconda
harfbuzz                  1.9.0                h9889186_1001    conda-forge
hdf5                      1.10.2                  hfa1e0ec_1    anaconda
```

```
icu                       58.2                 h0a44026_1000    conda-forge
idna                      2.8                      py36_1000    conda-forge
imageio                   2.5.0                    pypi_0    pypi
imgaug                    0.2.8                    pypi_0    pypi
intel-openmp              2019.1                     144
ipykernel                 5.1.0                py36h39e3cac_0
ipython                   7.2.0                py36h39e3cac_0
ipython_genutils          0.2.0                py36h241746c_0
ipywidgets                7.4.2                       py36_0
itsdangerous              1.1.0                         py_0    conda-forge
jasper                    1.900.1               h636a363_1006    conda-forge
jedi                      0.13.2                      py36_0
jinja2                    2.10                          py_1    conda-forge
jpeg                      9c                    h1de35cc_1001    conda-forge
jsonschema                2.6.0                py36hb385e00_0
jupyter                   1.0.0                       py36_7
jupyter_client            5.2.4                       py36_0
jupyter_console           6.0.0                       py36_0
jupyter_core              4.4.0                       py36_0
keras                     2.2.4                       py36_0    conda-forge
keras-applications        1.0.4                         py_1    conda-forge
keras-preprocessing       1.0.2                         py_1    conda-forge
kiwisolver                1.0.1             py36h04f5b5a_1002      conda-forge
libcxx                    7.0.0                    h2d50403_2    conda-forge
libffi                    3.2.1                 h0a44026_1005    conda-forge
libgfortran               3.0.1                    h93005f0_2
libgpuarray               0.7.6                 h1de35cc_1003    conda-forge
libiconv                  1.15                  h1de35cc_1004    conda-forge
libopenblas               0.3.3                    hdc02c5d_3
libpng                    1.6.36                   ha441bb4_0    anaconda
libprotobuf               3.6.1                 hd9629dc_1000    conda-forge
libsodium                 1.0.16                   h3efe00b_0
libtiff                   4.0.10                   hcb84e12_2    anaconda
libwebp                   1.0.2                    h801f6e5_1    conda-forge
libxml2                   2.9.8                 hf14e9c8_1005    conda-forge
llvm-meta                 7.0.0                            0    conda-forge
mako                      1.0.7                         py_1    conda-forge
markdown                  2.6.11                        py_0    conda-forge
markupsafe                1.1.0             py36h1de35cc_1000      conda-forge
matplotlib                3.0.2                     py36_1002    conda-forge
matplotlib-base           3.0.2             py36hf043ca5_1002      conda-forge
mistune                   0.8.4              py36h1de35cc_0
```

```
mkl                       2019.1                144
mkl_fft                   1.0.10             py36_0    conda-forge
mkl_random                1.0.2              py36_0    conda-forge
nbconvert                 5.3.1              py36_0
nbformat                  4.4.0           py36h827af21_0
ncurses                   6.1              h0a44026_1002    conda-forge
nettle                    3.4.1            h1de35cc_1002    conda-forge
networkx                  2.2                pypi_0    pypi
notebook                  5.7.4              py36_0
numpy                     1.15.4          py36hacdab7b_0    anaconda
numpy-base                1.15.4          py36h6575580_0    anaconda
olefile                   0.46               py36_0    anaconda
openblas                  0.3.3            hdc02c5d_1001    conda-forge
opencv                    3.4.1           py36h6fd60c2_1    anaconda
opencv-python             4.0.0.21           pypi_0    pypi
openh264                  1.8.0            hd9629dc_1000    conda-forge
openssl                   1.1.1a              h1de35cc_0
pandas                    0.24.1          py36h0a44026_0    anaconda
pandoc                    2.2.3.2                0
pandocfilters             1.4.2              py36_1
parso                     0.3.2              py36_0
pcre                      8.41             h0a44026_1003    conda-forge
pexpect                   4.6.0              py36_0
pickleshare               0.7.5              py36_0
pillow                    5.4.1           py36hb68e598_0    anaconda
pip                       19.0.1             py36_0
pixman                    0.34.0           h1de35cc_1003    conda-forge
prometheus_client         0.5.0              py36_0
prompt_toolkit            2.0.8                py_0
protobuf                  3.6.1         py36h0a44026_1001    conda-forge
ptyprocess                0.6.0              py36_0
pycparser                 2.19                 py_0    conda-forge
pygments                  2.3.1              py36_0
pygpu                     0.7.6         py36h917ab60_1000    conda-forge
pyopenssl                 19.0.0             py36_0    conda-forge
pyparsing                 2.3.1                py_0    conda-forge
pyqt                      5.9.2           py36h655552a_2
python                    3.6.8              haf84260_0
python-dateutil           2.7.5              py36_0
python-engineio           3.0.0                py_0    conda-forge
python-socketio           3.1.2                py_0    conda-forge
pytz                      2018.9             py36_0    anaconda
```

```
pywavelets                1.0.1                 pypi_0    pypi
pyyaml                    3.13           py36h1de35cc_1001    conda-forge
pyzmq                     17.1.2          py36h0a44026_2
qt                        5.9.7               h468cd18_1
qtconsole                 4.4.3                  py36_0
readline                  7.0             hcfe32e1_1001    conda-forge
scikit-image              0.14.2                 pypi_0    pypi
scikit-learn              0.20.2          py36h27c97d8_0    anaconda
scipy                     1.2.0           py36h1410ff5_0    anaconda
send2trash                1.5.0                  py36_0
setuptools                40.8.0                 py36_0    conda-forge
shapely                   1.6.4.post2            pypi_0    pypi
sip                       4.19.8          py36h0a44026_0
six                       1.12.0               py36_1000    conda-forge
sqlite                    3.26.0           h1765d9f_1000    conda-forge
tensorboard               1.10.0                 py36_0    conda-forge
tensorflow                1.10.0                 py36_0    conda-forge
termcolor                 1.1.0                    py_2    conda-forge
terminado                 0.8.1                  py36_1
testpath                  0.4.2                  py36_0
theano                    1.0.3                  py36_0    conda-forge
tk                        8.6.9            ha441bb4_1000    conda-forge
toolz                     0.9.0                  pypi_0    pypi
tornado                   5.1.1           py36h1de35cc_0
traitlets                 4.3.2           py36h65bd3ce_0
wcwidth                   0.1.7           py36h8c6ec74_0
webencodings              0.5.1                  py36_1
werkzeug                  0.14.1                   py_0    conda-forge
wheel                     0.32.3                 py36_0    conda-forge
widgetsnbextension        3.4.2                  py36_0
x264                      1!152.20180717    h1de35cc_1001    conda-forge
xz                        5.2.4             h1de35cc_1001    conda-forge
yaml                      0.1.7             h1de35cc_1001    conda-forge
zeromq                    4.3.1                h0a44026_3
zlib                      1.2.11            h1de35cc_1004    conda-forge
zstd                      1.3.7                h5bba6e5_0    anaconda
```

Let us do the imports. All the imports done were collected and collated here

In [46]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import keras
from keras.models import Sequential
from keras.optimizers import Adam
from keras.layers import Convolution2D, 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
```

Handle the CSV file containing ref to snapshot of manual driving files simulating real cameras on a car

In [47]:
```
datadir = 'Trail1_combined'
columns =['center','left','right','steering','throttle','reverse','speed']
data=pd.read_csv('Trail1_Combined.csv',names=columns)
data.head
```

```
Out[47]: <bound method NDFrame.head of                                                 center  \
0     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
1     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
2     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
3     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
4     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
5     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
7     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
8     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
9     C:\Users\Vijy\Desktop\new_track\IMG\center_201...
10    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
11    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
12    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
13    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
14    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
15    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
16    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
17    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
18    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
19    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
20    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
21    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
22    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
23    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
24    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
25    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
26    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
27    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
28    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
29    C:\Users\Vijy\Desktop\new_track\IMG\center_201...
...                                                 ...
6704  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6705  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6706  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6707  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6708  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6709  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6710  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6711  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6712  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6713  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
```

```
6714  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6715  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6716  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6717  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6718  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6719  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6720  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6721  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6722  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6723  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6724  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6725  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6726  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6727  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6728  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6729  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6730  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6731  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6732  C:\Users\Vijy\Desktop\new_track\IMG\center_201...
6733  C:\Users\Vijy\Desktop\new_track\IMG\center_201...

                                                  left  \
0      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
1      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
2      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
3      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
4      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
5      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
7      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
8      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
9      C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
10     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
11     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
12     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
13     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
14     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
15     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
16     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
17     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
18     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
19     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
```

```
20     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
21     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
22     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
23     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
24     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
25     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
26     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
27     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
28     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
29     C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
...                                              ...
6704   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6705   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6706   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6707   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6708   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6709   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6710   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6711   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6712   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6713   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6714   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6715   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6716   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6717   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6718   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6719   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6720   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6721   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6722   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6723   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6724   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6725   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6726   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6727   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6728   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6729   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6730   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6731   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6732   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
6733   C:\Users\Vijy\Desktop\new_track\IMG\left_2019...
```

	right	steering	throttle	\
0	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.158175	
1	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
2	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
3	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
4	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
5	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
6	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
7	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
8	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.592151	0.294393	
9	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.767385	0.439335	
10	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.466801	0.468232	
11	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.342845	0.447736	
12	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.014343	0.403693	
13	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.401168	
14	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.401168	
15	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.110154	0.401168	
16	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.095912	0.435303	
17	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.147981	0.452559	
18	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.353507	0.437413	
19	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.307278	0.400076	
20	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.294807	0.368201	
21	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.279360	0.252157	
22	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.137997	
23	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.149300	
24	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.028396	0.152427	
25	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
26	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
27	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
28	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000	
29	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	-0.505005	0.183360	
...	...	...	...	...
6704	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.135398	
6705	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.162449	
6706	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.165991	
6707	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.169382	
6708	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.169382	
6709	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.061928	0.169382	
6710	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.202349	0.172999	
6711	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.214744	0.172999	
6712	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.214744	0.172999	
6713	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.081181	0.161846	

6714	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.078091
6715	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.064754
6716	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.062192
6717	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.062192
6718	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.062192
6719	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.095988	0.065809
6720	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.232942	0.069237
6721	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.205740	0.072930
6722	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.001080	0.070594
6723	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.050474
6724	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.025985
6725	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.023460
6726	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.134569	0.041545
6727	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.198732	0.063247
6728	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.198732	0.066751
6729	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.193080	0.066751
6730	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000
6731	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.000000
6732	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.201654
6733	C:\Users\Vijy\Desktop\new_track\IMG\right_201...	0.000000	0.165450

	reverse	speed
0	0.0	0.928941
1	0.0	0.921869
2	0.0	0.910790
3	0.0	0.899844
4	0.0	0.889031
5	0.0	0.878348
6	0.0	0.869542
7	0.0	0.859088
8	0.0	1.054672
9	0.0	1.493697
10	0.0	2.278393
11	0.0	2.952424
12	0.0	3.511507
13	0.0	4.051748
14	0.0	4.496209
15	0.0	4.933930
16	0.0	5.477810
17	0.0	6.067337
18	0.0	6.719277
19	0.0	7.140470

20	0.0	7.771864
21	0.0	8.053087
22	0.0	8.212673
23	0.0	8.305655
24	0.0	8.458338
25	0.0	8.384179
26	0.0	8.300022
27	0.0	8.183642
28	0.0	8.085201
29	0.0	8.009748
...	...	...
6704	0.0	13.447340
6705	0.0	13.535550
6706	0.0	13.081500
6707	0.0	13.701100
6708	0.0	13.786840
6709	0.0	13.305300
6710	0.0	13.347960
6711	0.0	13.464280
6712	0.0	14.060590
6713	0.0	14.157290
6714	0.0	14.112980
6715	0.0	13.474890
6716	0.0	13.959670
6717	0.0	13.882770
6718	0.0	13.243220
6719	0.0	13.167570
6720	0.0	13.636270
6721	0.0	13.008220
6722	0.0	13.502640
6723	0.0	12.899170
6724	0.0	13.311720
6725	0.0	13.186840
6726	0.0	13.051590
6727	0.0	12.405240
6728	0.0	12.324380
6729	0.0	12.828130
6730	0.0	12.725410
6731	0.0	12.142730
6732	0.0	12.043820
6733	0.0	12.669170

```
[6734 rows x 7 columns]>
```

Handle path data and verifing first few lines with 'head'

In [48]:
```
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()
```

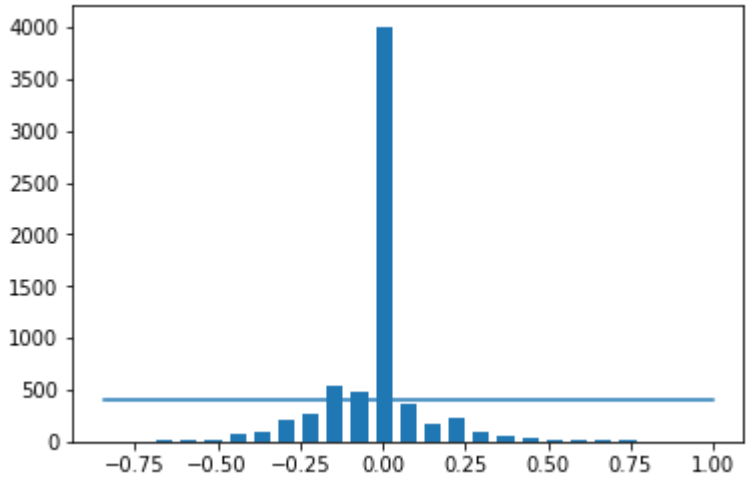
Out[48]:

	center	left	right	steering	throttle	reve
0	center_2019_02_12_19_16_37_349.jpg	left_2019_02_12_19_16_37_349.jpg	right_2019_02_12_19_16_37_349.jpg	0.0	0.158175	0.0
1	center_2019_02_12_19_16_37_466.jpg	left_2019_02_12_19_16_37_466.jpg	right_2019_02_12_19_16_37_466.jpg	0.0	0.000000	0.0
2	center_2019_02_12_19_16_37_574.jpg	left_2019_02_12_19_16_37_574.jpg	right_2019_02_12_19_16_37_574.jpg	0.0	0.000000	0.0
3	center_2019_02_12_19_16_37_701.jpg	left_2019_02_12_19_16_37_701.jpg	right_2019_02_12_19_16_37_701.jpg	0.0	0.000000	0.0
4	center_2019_02_12_19_16_37_819.jpg	left_2019_02_12_19_16_37_819.jpg	right_2019_02_12_19_16_37_819.jpg	0.0	0.000000	0.0

Plotting steering angle as bins between values of -1 and +1, against number of occurances in the training

In [49]:
```
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))
```

Out[49]:
```
[<matplotlib.lines.Line2D at 0x1c3c3f4f28>]
```



Data study for feature enineering and data wrangling. Removing too much bias for 0 deg straight angle and attemting at generalising beyond this predominatly straight track road

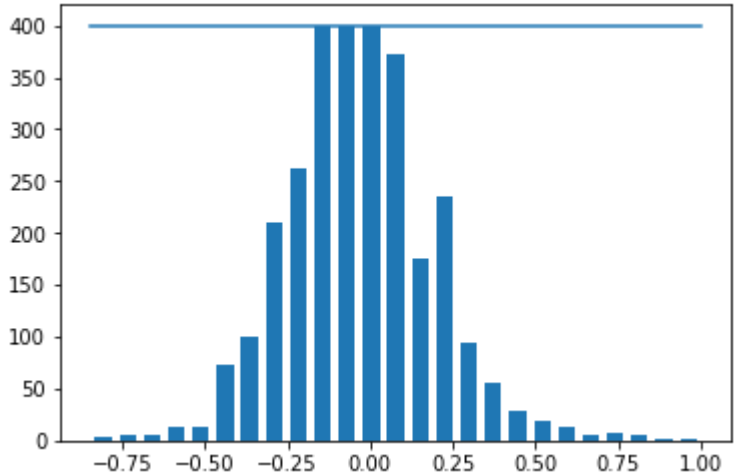
In [50]:
```
print('total data:', len(data))
remove_list = []
for j in range(num_bins):
  list_ = []
  for i in range(len(data['steering'])):
    if data['steering'][i] >= bins[j] and data['steering'][i] <= bins[j+1]:
      list_.append(i)
  list_ = shuffle(list_)
  list_ = list_[samples_per_bin:]
  remove_list.extend(list_)

print('removed:', len(remove_list))
data.drop(data.index[remove_list], inplace=True)
print('remaining:', len(data))
```

```
total data: 6734
removed: 3831
remaining: 2903
```

Histogramming steering after removal

In [51]:
```
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))
```

Out[51]:
```
[<matplotlib.lines.Line2D at 0x1c42b1c978>]
```



Printing a sample and verifying data pattern

In [52]:
```
print(data.iloc[1])
```

```
center      center_2019_02_12_19_16_38_271.jpg
left          left_2019_02_12_19_16_38_271.jpg
right        right_2019_02_12_19_16_38_271.jpg
steering                             -0.592151
throttle                              0.294393
reverse                                      0
speed                                  1.05467
Name: 8, dtype: object
```

Function for picking up images and returning image paths and steering angle value from csv file

In [53]:
```
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]))
    # left image append
    image_path.append(os.path.join(datadir,left.strip()))
    steering.append(float(indexed_data[3])+0.15)
    # right image append
    image_path.append(os.path.join(datadir,right.strip()))
    steering.append(float(indexed_data[3])-0.15)
  image_paths = np.asarray(image_path)
  steerings = np.asarray(steering)
  return image_paths, steerings
```

Adding real file system point for dataset. Spliting training and validation set from collected data

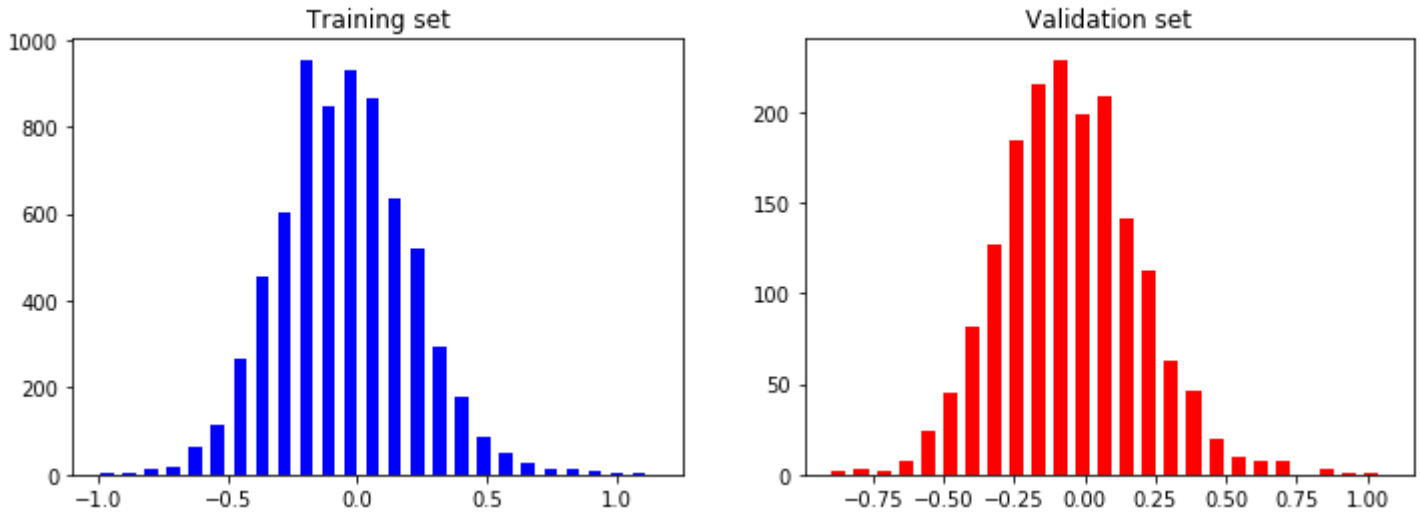
In [54]:
```
image_paths, steerings = load_img_steering(datadir + '/IMG', data)
X_train, X_valid, y_train, y_valid = train_test_split(image_paths, steerings, test_size=0.2, random_state=6)
```

Printing out our training and validation split set, plotting histogram to visualise steering angles

In [55]:
```
print('Training Samples: {}\nValid Samples: {}'.format(len(X_train), len(X_valid)))
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')
```

```
Training Samples: 6967
Valid Samples: 1742
```

Out[55]: Text(0.5, 1.0, 'Validation set')



Augmentation Techniques

We attempt various augmentation methods to try and generalise trained model from the collected training data.

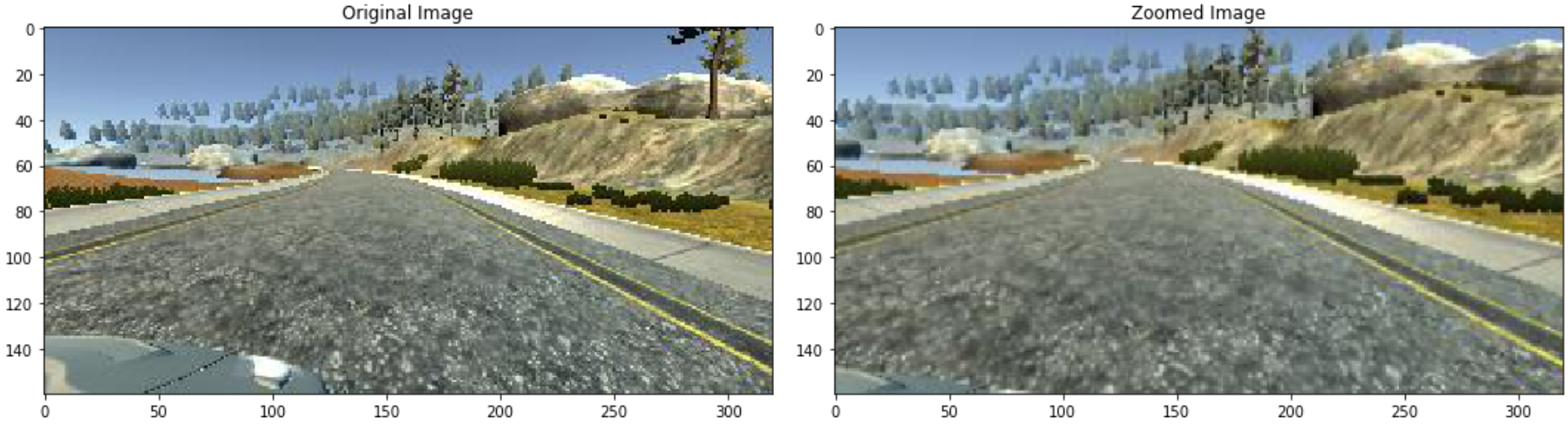
Zoom affine augmentation

In [56]:
```
def zoom(image):
  zoom = iaa.Affine(scale=(1, 1.3))
  image = zoom.augment_image(image)
  return image
```

In [57]:
```
# print(image)
```

In [58]:
```
image = image_paths[random.randint(0, 1000)]
original_image = mpimg.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')
```

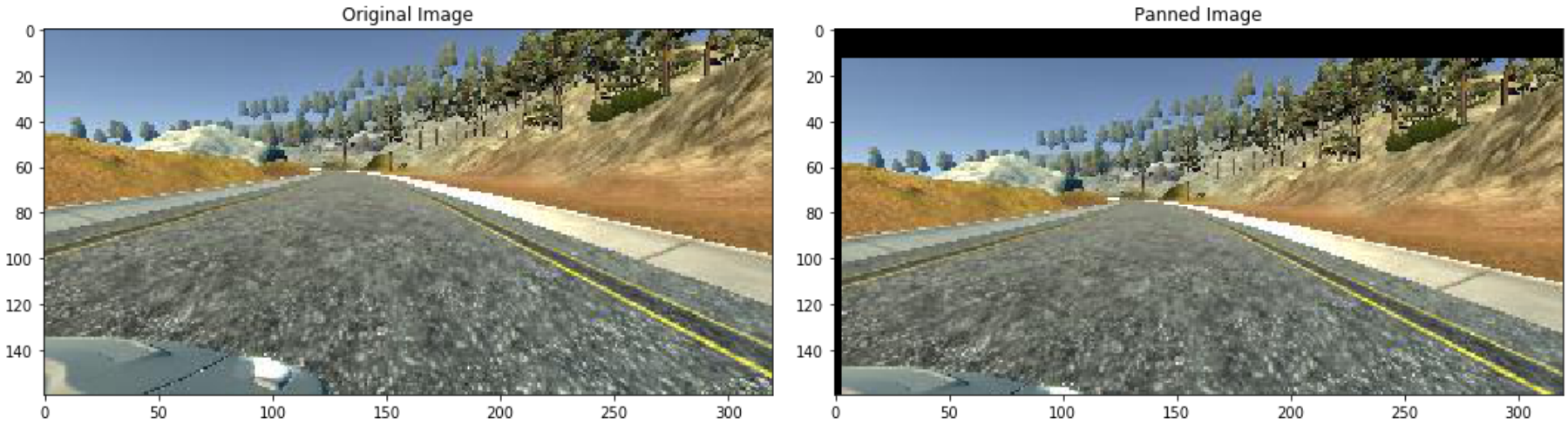
Out[58]: Text(0.5, 1.0, 'Zoomed Image')



Pan augmentation

In [59]:
```
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 = mpimg.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')
```

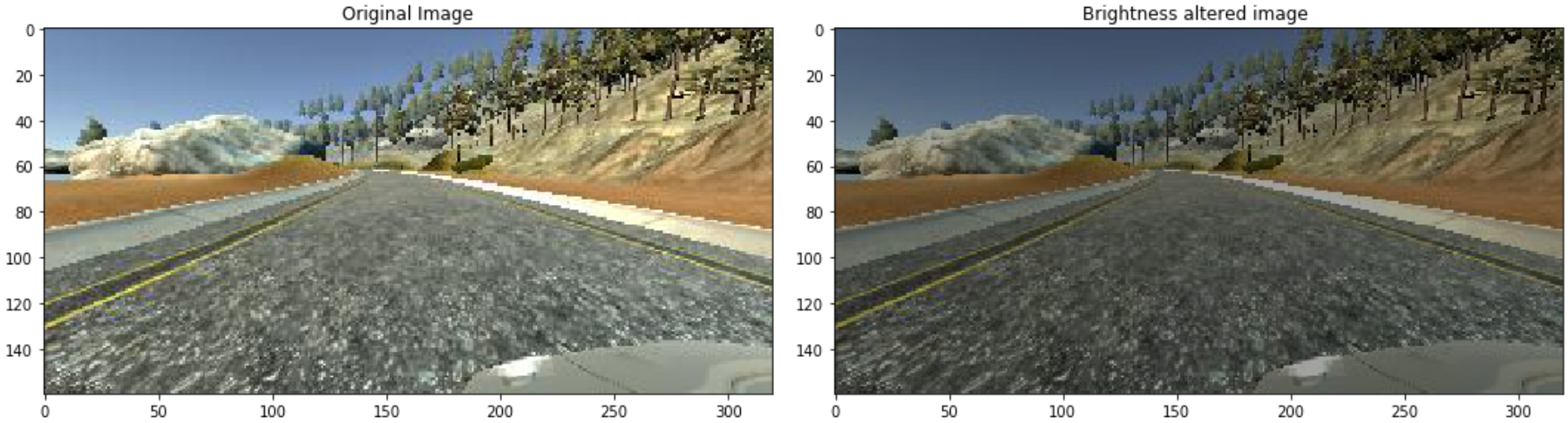
Out[59]: Text(0.5, 1.0, 'Panned Image')



Altered Brightess augmentation

In [60]:
```
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 = mpimg.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 ')
```

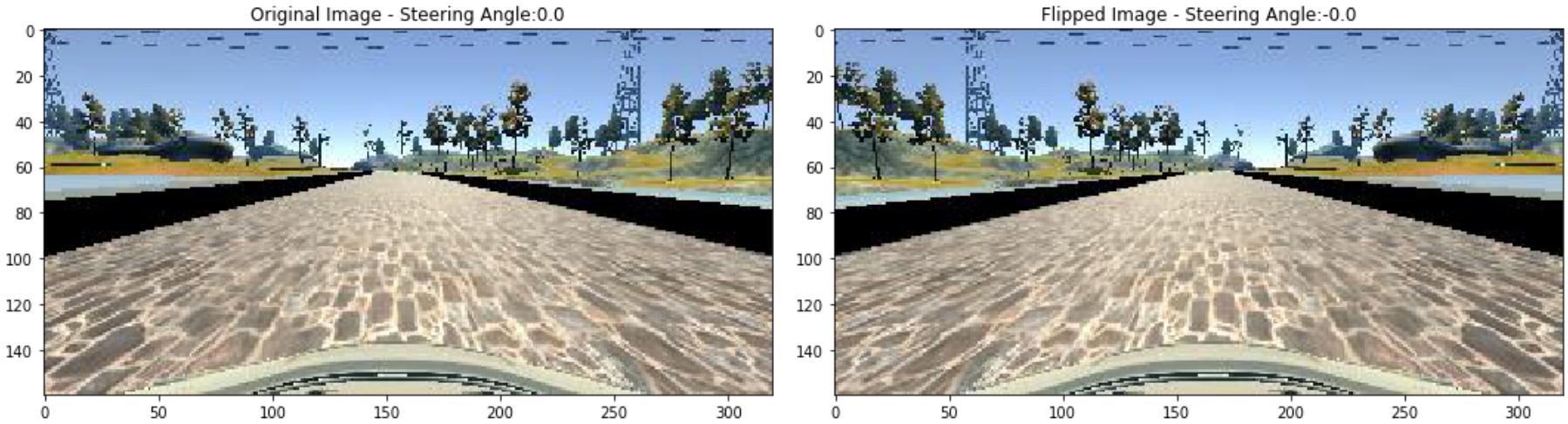
Out[60]: Text(0.5, 1.0, 'Brightness altered image ')



Random Flip Augmentation

In [61]:
```
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 = mpimg.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))
```

Out[61]: Text(0.5, 1.0, 'Flipped Image - Steering Angle:-0.0')

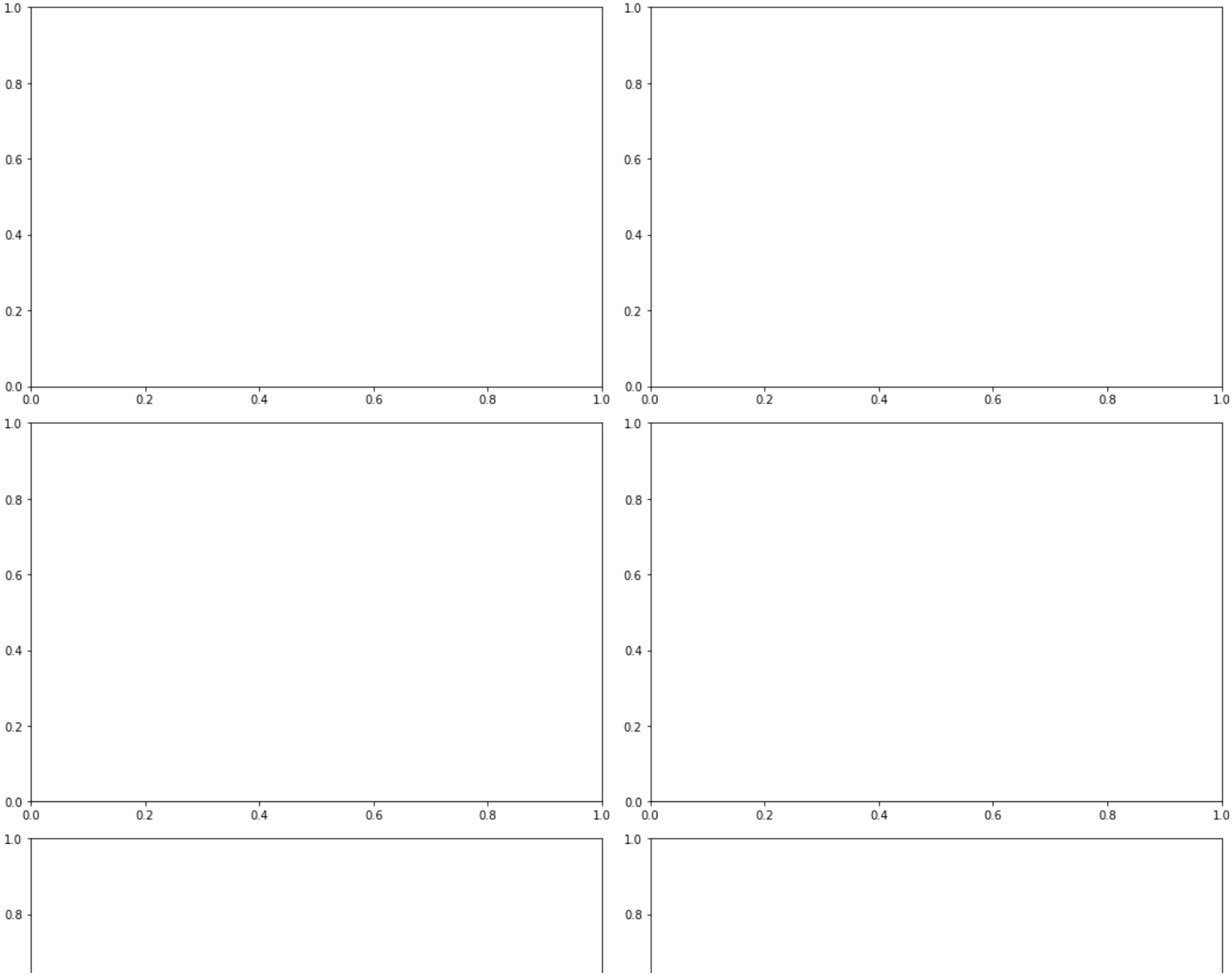


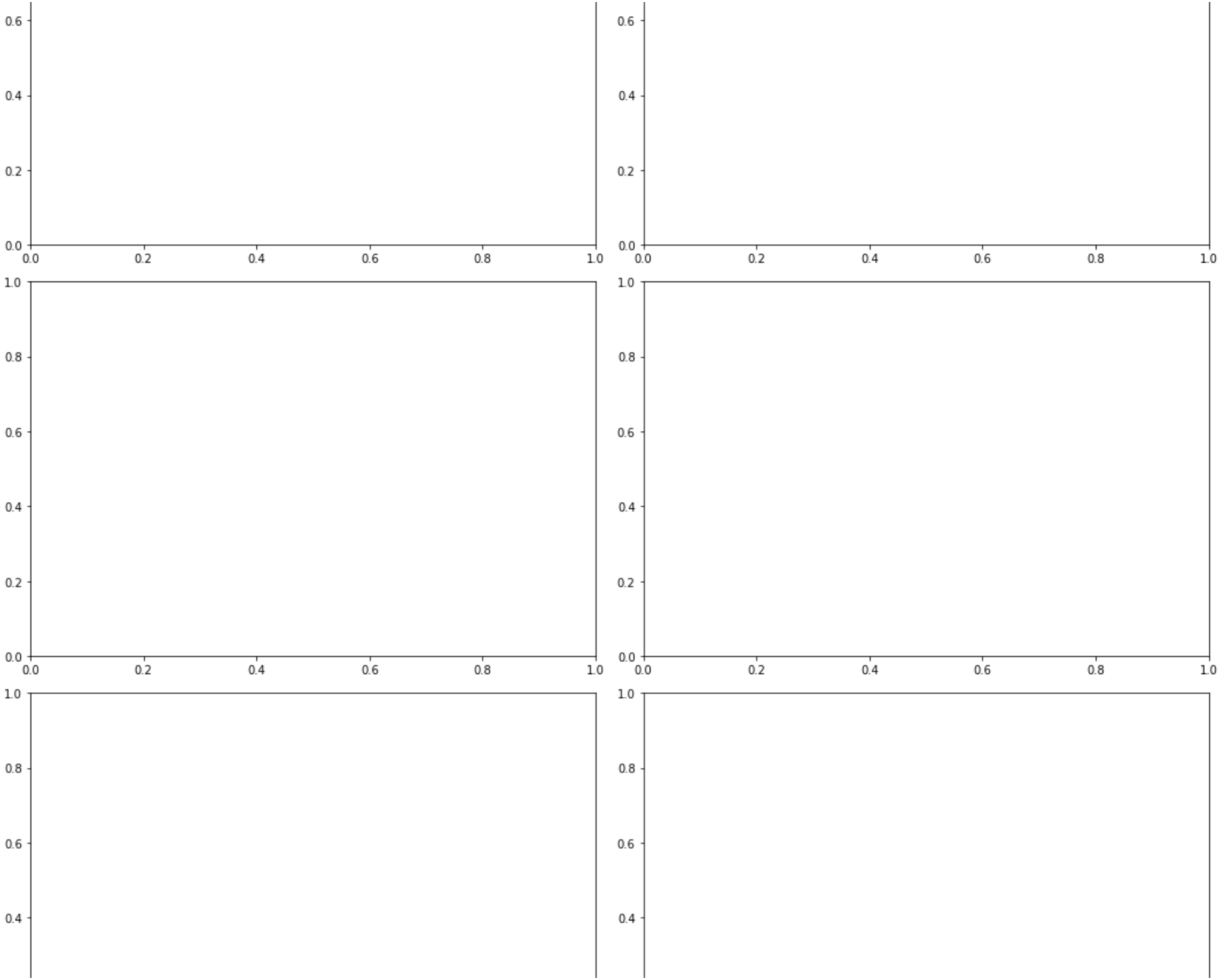
We do a 50% probablity for selective augmentation on various images

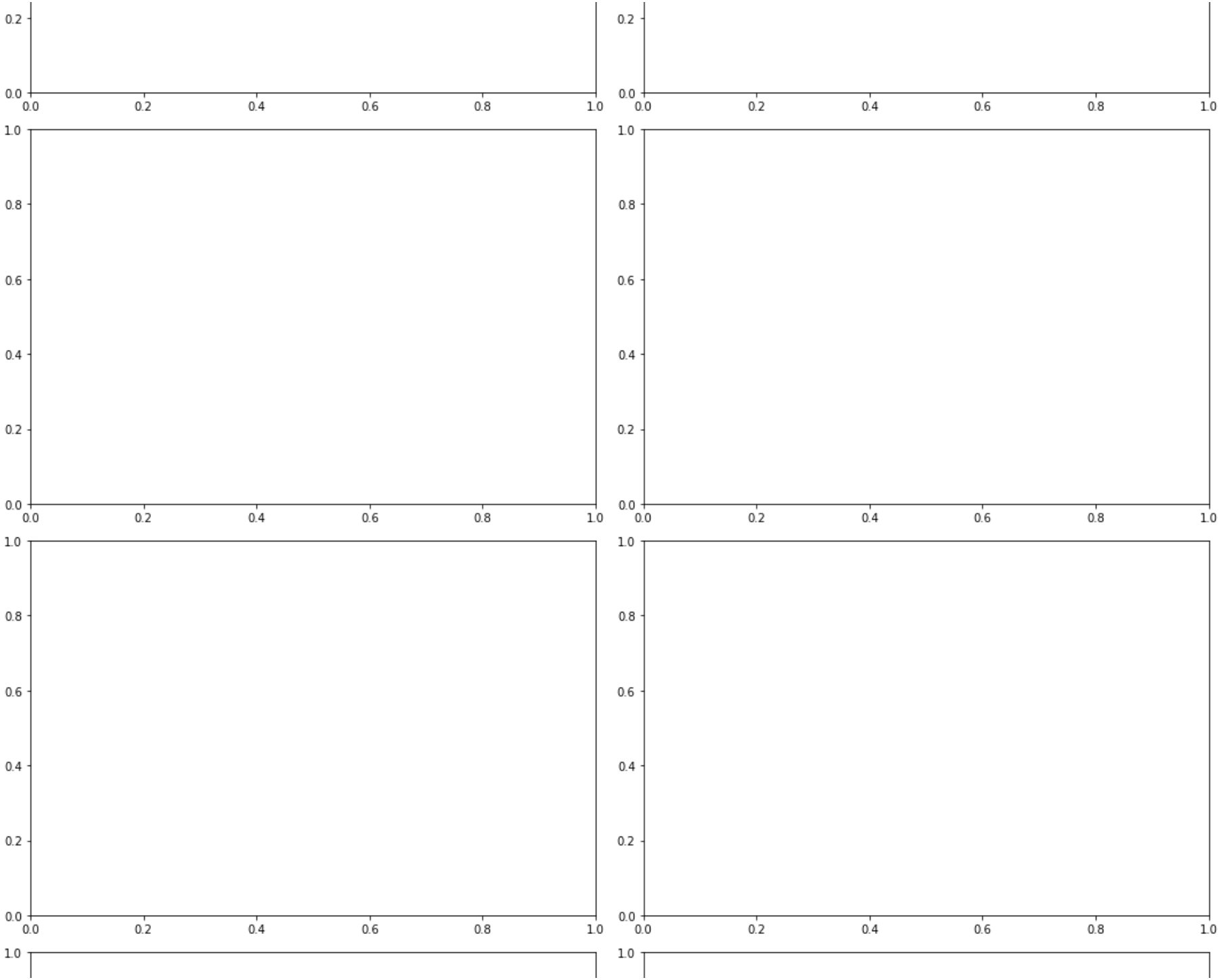
In [62]:
```
def random_augment(image, steering_angle):
    image = mpimg.imread(image)
    if np.random.rand() < 0.5:
      image = pan(image)
    if np.random.rand() < 0.5:
      image = zoom(image)
    if np.random.rand() < 0.5:
      image = img_random_brightness(image)
    if np.random.rand() < 0.5:
      image, steering_angle = img_random_flip(image, steering_angle)

    return image, steering_angle
```

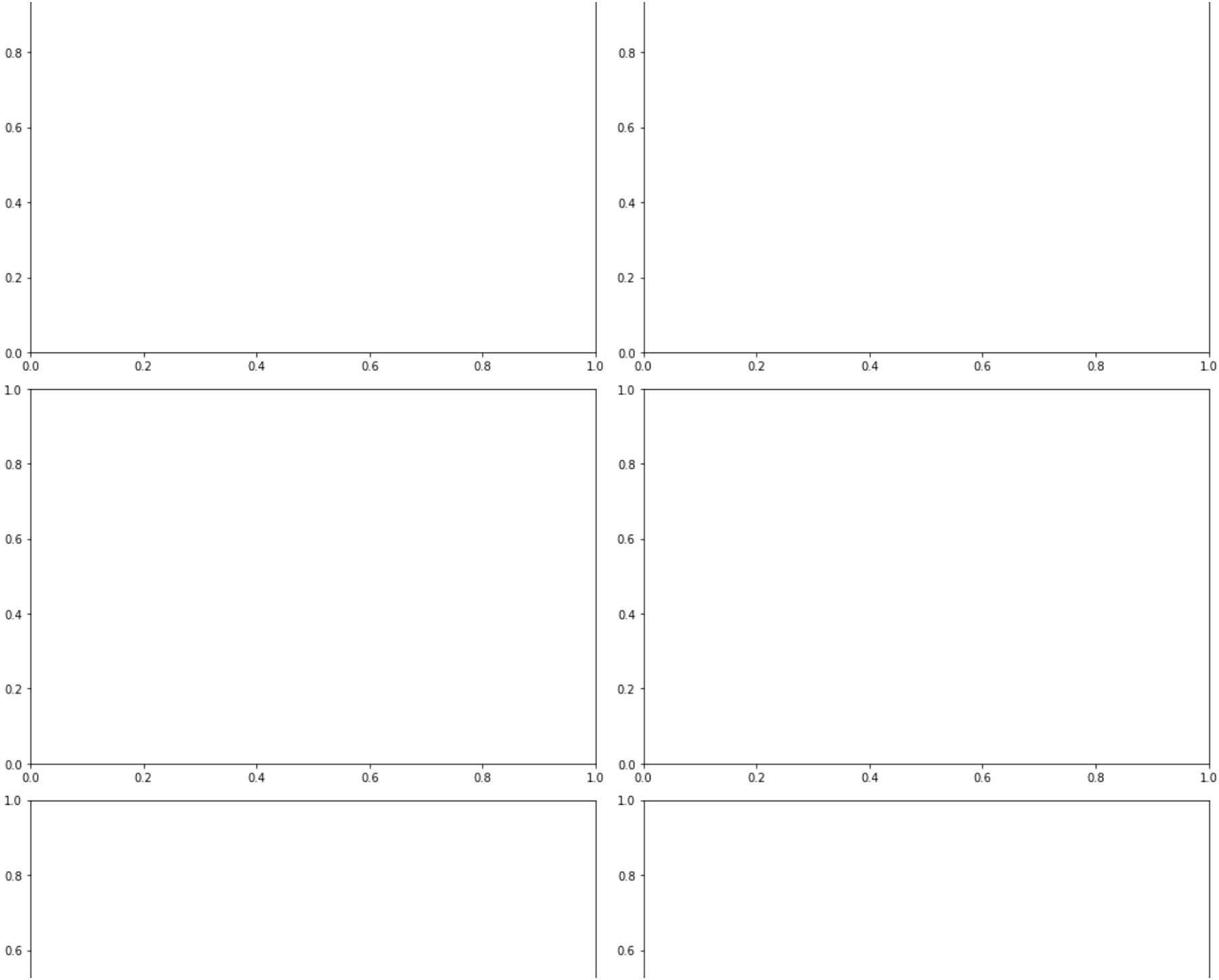
Preparing for plot areas to visualise augmentations

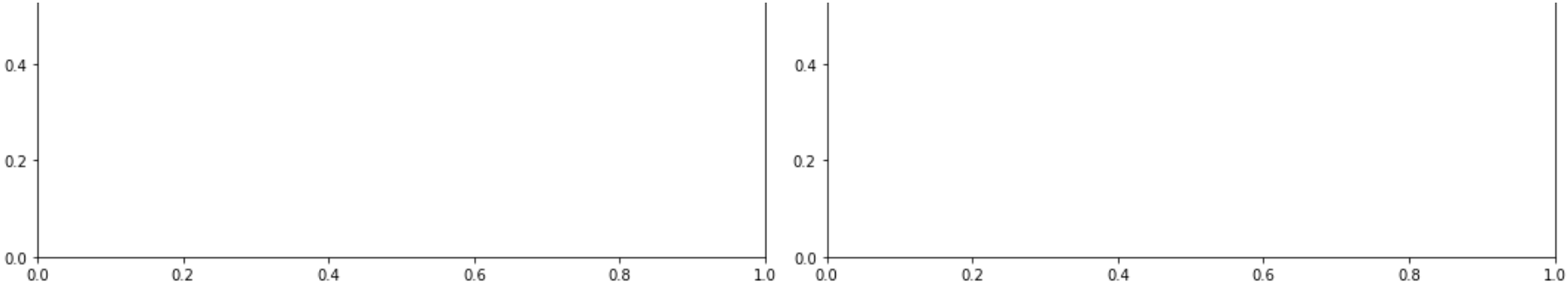
In [63]:
```
ncol = 2
nrow = 10

fig, axs = plt.subplots(nrow, ncol, figsize=(15, 50))
fig.tight_layout()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4

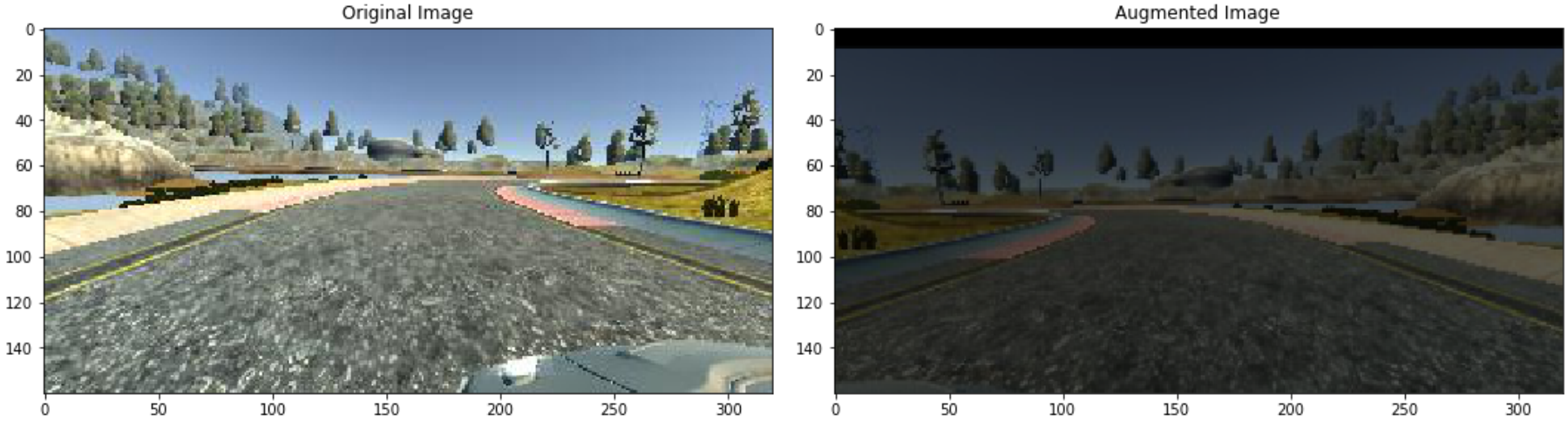

0.0
0.2
0.4
0.6
0.8
1.0

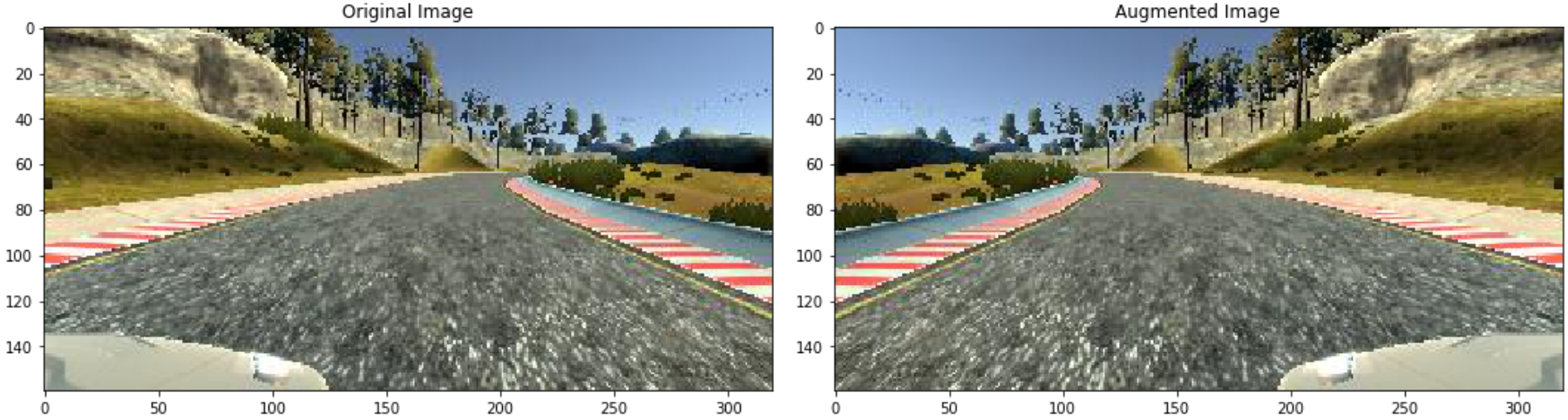

0.0
0.2
0.4
0.6
0.8
1.0

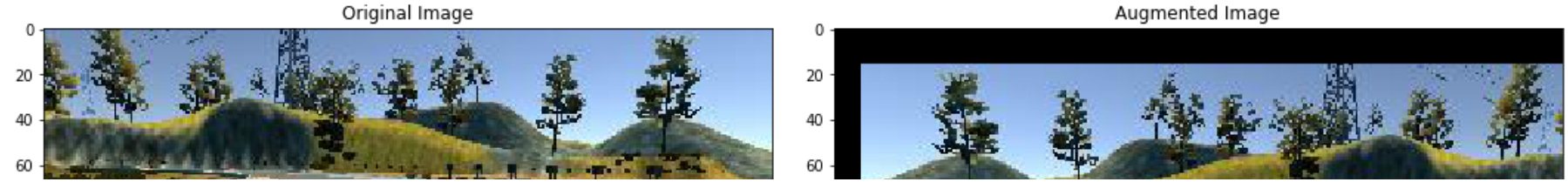


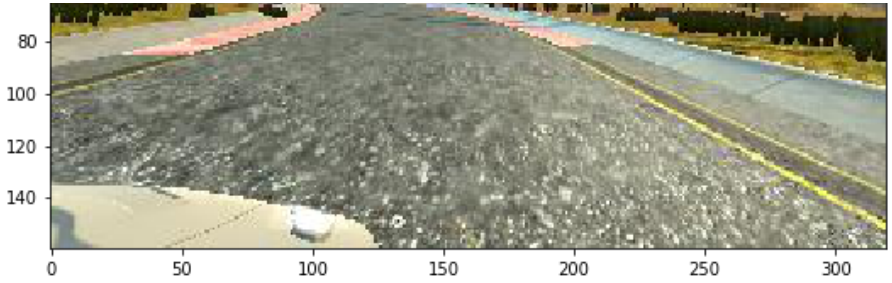
Augmentations visualised

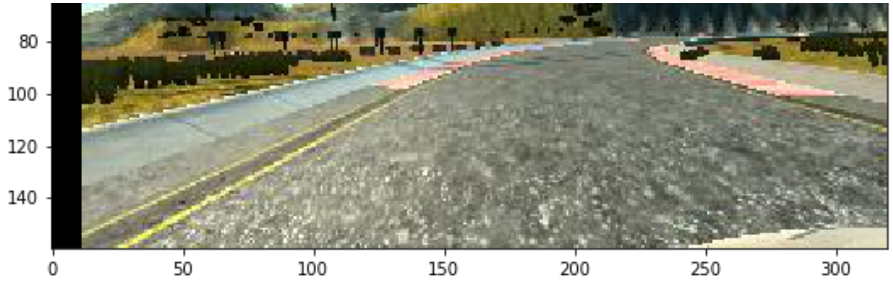
In [64]:
```
ncol = 2
nrow = 10

fig, axs = plt.subplots(nrow, ncol, 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 = mpimg.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")
```

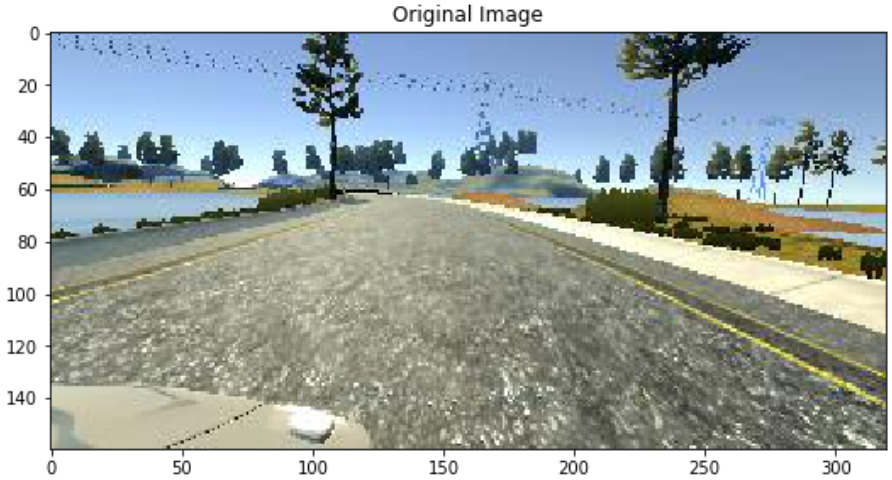

Original Image
Augmented Image

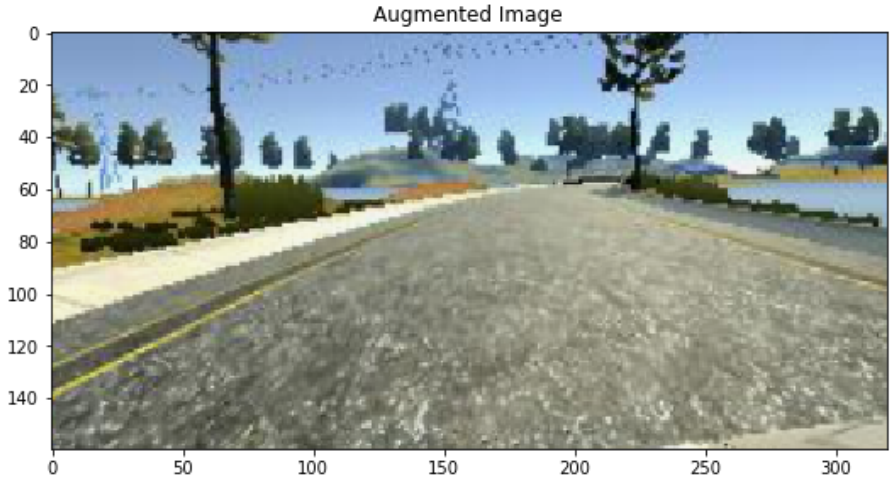

Original Image
Augmented Image


Original Image
Augmented Image


80
100
120
140
0
50
100
150
200
250
300

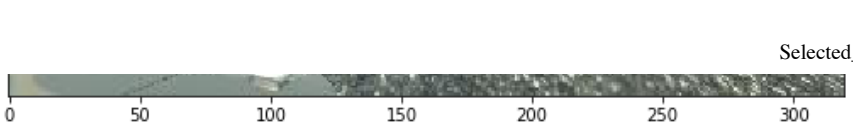

80
100
120
140
0
50
100
150
200
250
300

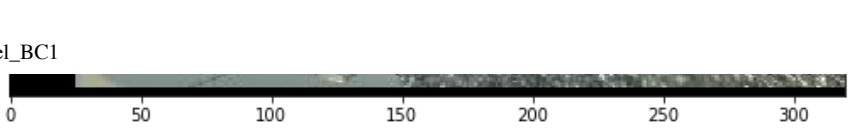

Original Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

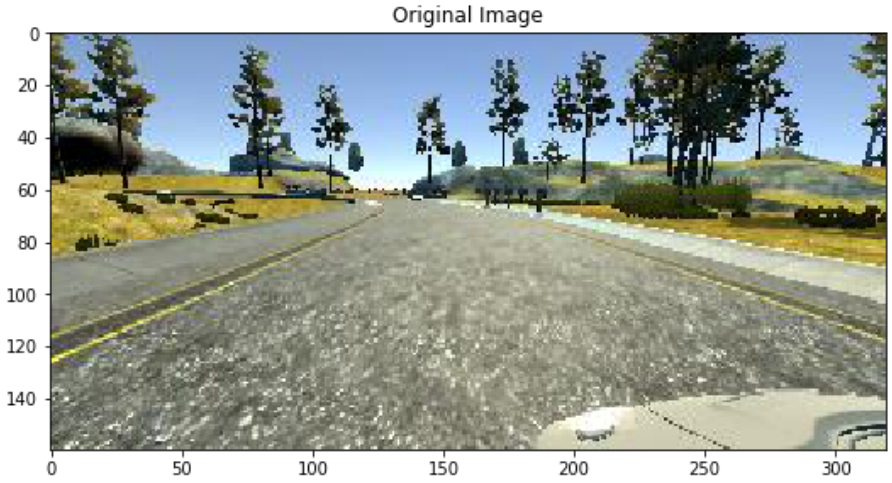

Augmented Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

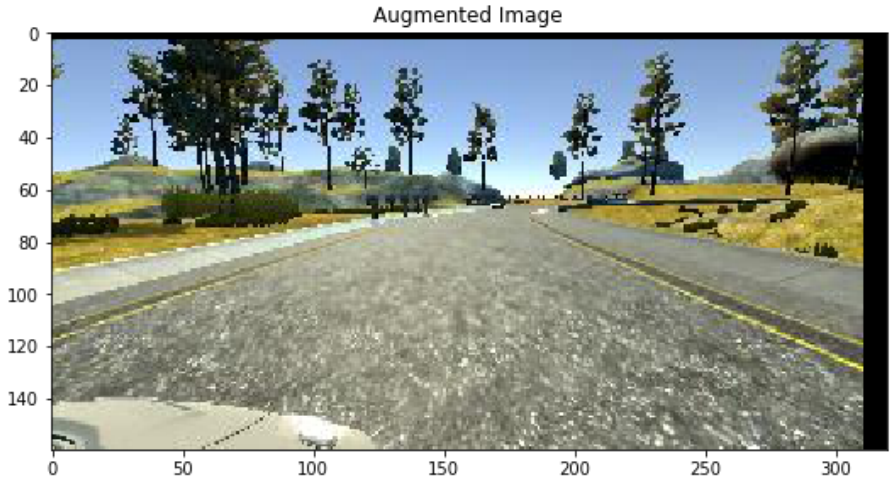

Original Image
0
20
40
60
80
100
120
140

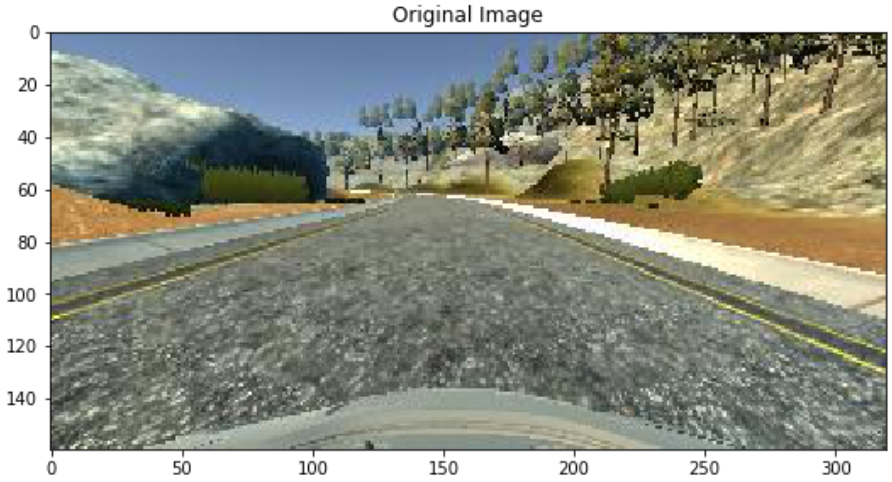

Augmented Image
0
20
40
60
80
100
120
140

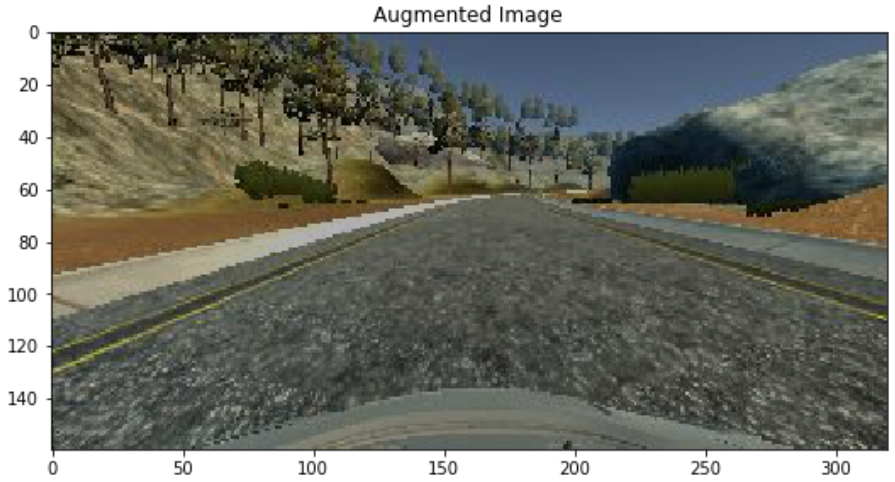

0
50
100
150
200
250
300

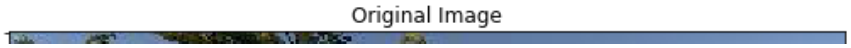

0
50
100
150
200
250
300

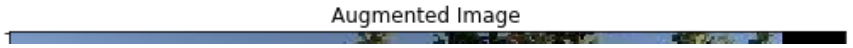

Original Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

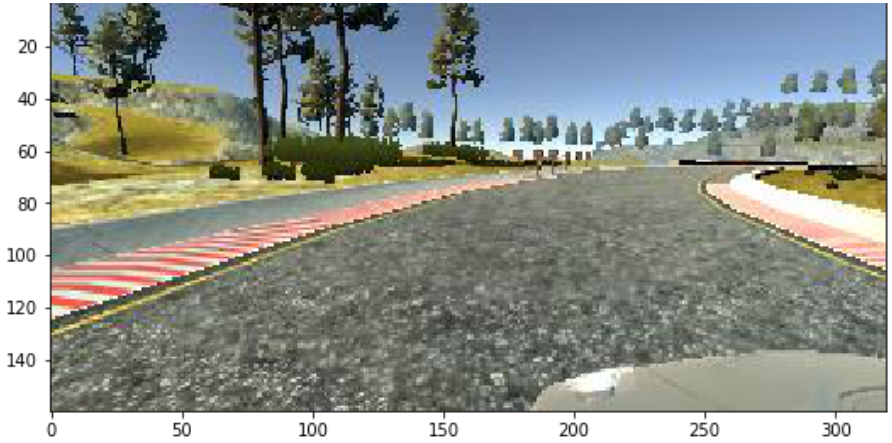

Augmented Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

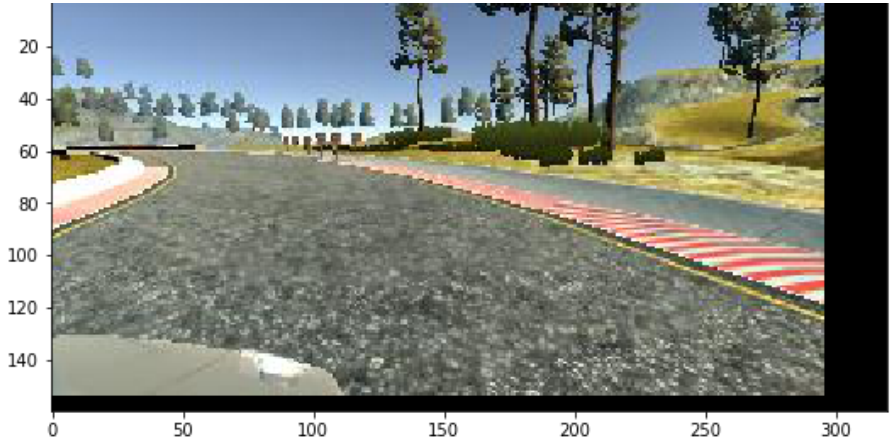

Original Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

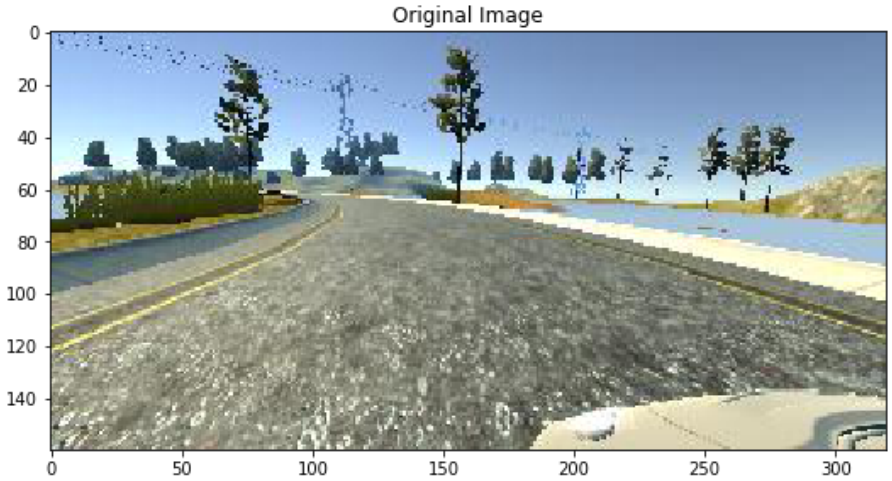

Augmented Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300

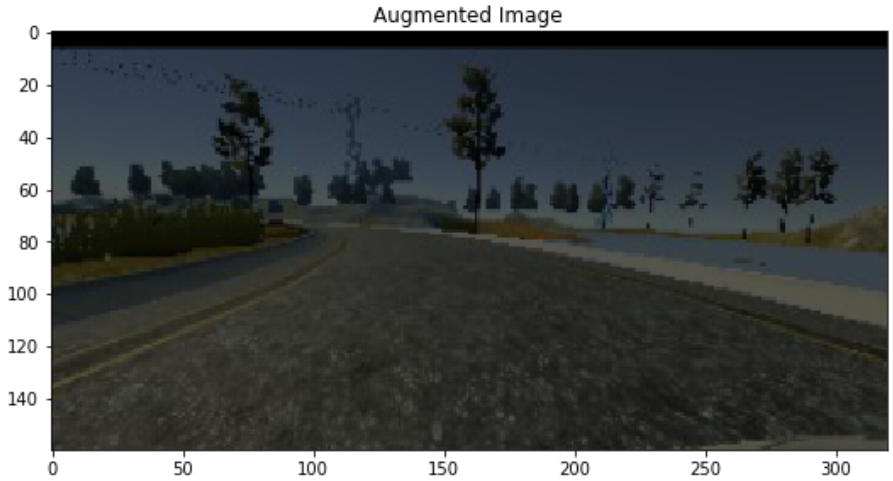

Original Image
0


Augmented Image
0


20
40
60
80
100
120
140
0
50
100
150
200
250
300

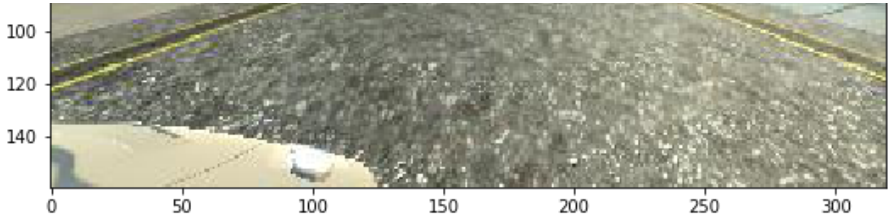

20
40
60
80
100
120
140
0
50
100
150
200
250
300

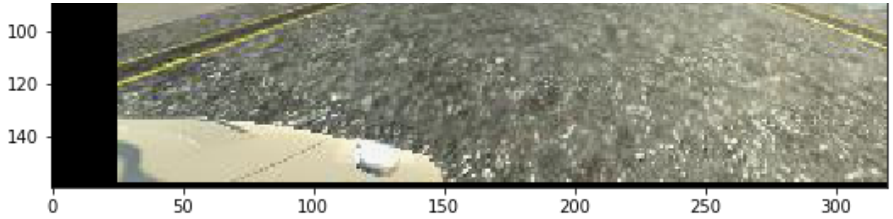

Original Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300


Augmented Image
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300


Original Image
0
20
40
60
80


Augmented Image
0
20
40
60
80



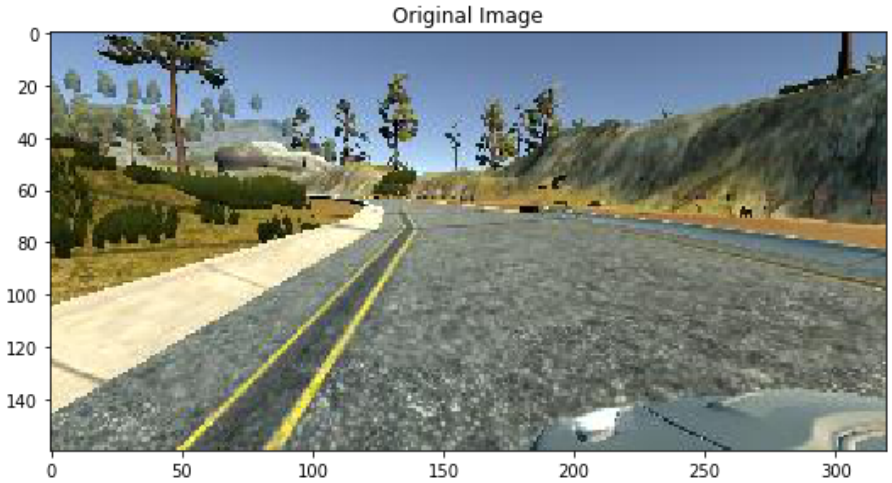


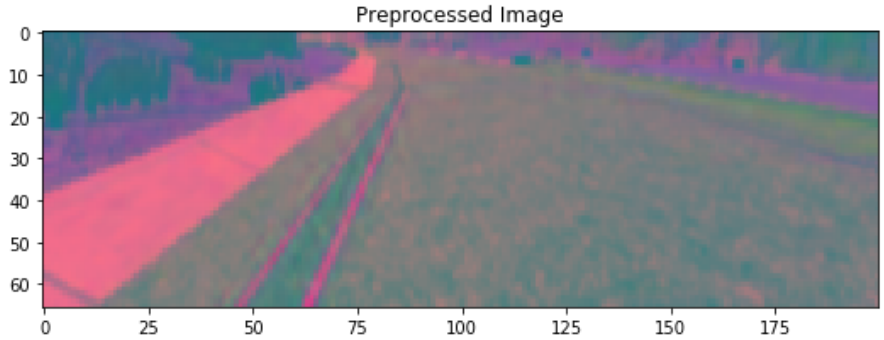
Usual regular Preprocessing of RGB to YUV,Gauss Blurr, and resizing

In [65]:
```
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
```

In [66]:
```
image = image_paths[100]
original_image = mpimg.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')
```

Out[66]: Text(0.5, 1.0, 'Preprocessed Image')





A Realtime batch processor with 'Yield' as return to handle realtime simulation

In [67]:
```
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)

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

      else:
        im = mpimg.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))
```

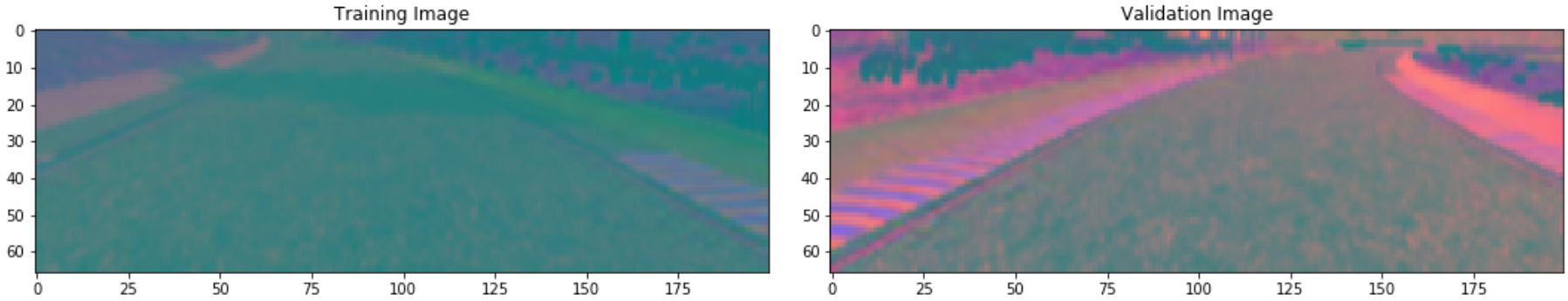
Training in batches using batch generator

In [68]:
```
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))
```

Attemting visual verification of training and validation image from batch generator output

In [69]:
```
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')
```

Out[69]: Text(0.5, 1.0, 'Validation Image')



Building NVIDIA Model

Thanks to NVIDIA for paper,figs & model are from two nvidia papers and references mentioned. Fully acknowledged, appreciated and thanked for sharing with public.

Title :End to End Learning for Self-Driving Cars

REF :https://arxiv.org/pdf/1604.07316v1.pdf¶ (https://arxiv.org/pdf/1604.07316v1.pdf¶)

I also found another interesting but advanced paper from NVIDIA

TItle : Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car

https://arxiv.org/pdf/1704.07911.pdf (https://arxiv.org/pdf/1704.07911.pdf)

In [75]:
```
def 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-3)
  model.compile(loss='mse', optimizer=optimizer)
  return model
model = nvidia_model()
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_16 (Conv2D)           (None, 31, 98, 24)        1824      
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 14, 47, 36)        21636     
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 5, 22, 48)         43248     
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 3, 20, 64)         27712     
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 1, 18, 64)         36928     
_________________________________________________________________
flatten_4 (Flatten)          (None, 1152)              0         
_________________________________________________________________
dense_13 (Dense)             (None, 100)               115300    
_________________________________________________________________
dropout_7 (Dropout)          (None, 100)               0         
_________________________________________________________________
dense_14 (Dense)             (None, 50)                5050      
_________________________________________________________________
dense_15 (Dense)             (None, 10)                510       
_________________________________________________________________
dropout_8 (Dropout)          (None, 10)                0         
_________________________________________________________________
dense_16 (Dense)             (None, 1)                 11        
=================================================================
Total params: 252,219
Trainable params: 252,219
Non-trainable params: 0
_________________________________________________________________
None
```

```
/miniconda3/envs/udrivesimul/lib/python3.6/site-packages/ipykernel_launcher.py:3: UserWarning: Update your `
Conv2D` call to the Keras 2 API: `Conv2D(24, (5, 5), input_shape=(66, 200, ..., activation="elu", strides=
(2, 2))`
  This is separate from the ipykernel package so we can avoid doing imports until
/miniconda3/envs/udrivesimul/lib/python3.6/site-packages/ipykernel_launcher.py:4: UserWarning: Update your `
Conv2D` call to the Keras 2 API: `Conv2D(36, (5, 5), activation="elu", strides=(2, 2))`
  after removing the cwd from sys.path.
/miniconda3/envs/udrivesimul/lib/python3.6/site-packages/ipykernel_launcher.py:5: UserWarning: Update your `
Conv2D` call to the Keras 2 API: `Conv2D(48, (5, 5), activation="elu", strides=(2, 2))`
  """
/miniconda3/envs/udrivesimul/lib/python3.6/site-packages/ipykernel_launcher.py:6: UserWarning: Update your `
Conv2D` call to the Keras 2 API: `Conv2D(64, (3, 3), activation="elu")`

/miniconda3/envs/udrivesimul/lib/python3.6/site-packages/ipykernel_launcher.py:8: UserWarning: Update your `
Conv2D` call to the Keras 2 API: `Conv2D(64, (3, 3), activation="elu")`
```

Training Time

In [76]:
```
history = model.fit_generator(batch_generator(X_train, y_train, 100, 1),
                                    steps_per_epoch=300,
                                    epochs=45,
                                    validation_data=batch_generator(X_valid, y_valid, 100, 0),
                                    validation_steps=200,
                                    verbose=1,
                                    shuffle = 1)
```

```
Epoch 1/45
300/300 [==============================] - 349s 1s/step - loss: 0.0658 - val_loss: 0.0347
Epoch 2/45
300/300 [==============================] - 349s 1s/step - loss: 0.0490 - val_loss: 0.0321
Epoch 3/45
300/300 [==============================] - 347s 1s/step - loss: 0.0481 - val_loss: 0.0316
Epoch 4/45
300/300 [==============================] - 347s 1s/step - loss: 0.0467 - val_loss: 0.0327
Epoch 5/45
300/300 [==============================] - 341s 1s/step - loss: 0.0454 - val_loss: 0.0326
Epoch 6/45
300/300 [==============================] - 338s 1s/step - loss: 0.0426 - val_loss: 0.0265
Epoch 7/45
300/300 [==============================] - 339s 1s/step - loss: 0.0410 - val_loss: 0.0239
Epoch 8/45
300/300 [==============================] - 339s 1s/step - loss: 0.0388 - val_loss: 0.0282
Epoch 9/45
300/300 [==============================] - 339s 1s/step - loss: 0.0374 - val_loss: 0.0262
Epoch 10/45
300/300 [==============================] - 339s 1s/step - loss: 0.0362 - val_loss: 0.0281
Epoch 11/45
300/300 [==============================] - 347s 1s/step - loss: 0.0360 - val_loss: 0.0229
Epoch 12/45
300/300 [==============================] - 347s 1s/step - loss: 0.0353 - val_loss: 0.0264
Epoch 13/45
300/300 [==============================] - 348s 1s/step - loss: 0.0353 - val_loss: 0.0259
Epoch 14/45
300/300 [==============================] - 350s 1s/step - loss: 0.0347 - val_loss: 0.0251
Epoch 15/45
300/300 [==============================] - 340s 1s/step - loss: 0.0331 - val_loss: 0.0230
Epoch 16/45
300/300 [==============================] - 342s 1s/step - loss: 0.0323 - val_loss: 0.0252
Epoch 17/45
300/300 [==============================] - 346s 1s/step - loss: 0.0320 - val_loss: 0.0230
Epoch 18/45
300/300 [==============================] - 350s 1s/step - loss: 0.0307 - val_loss: 0.0227
Epoch 19/45
300/300 [==============================] - 347s 1s/step - loss: 0.0315 - val_loss: 0.0228
Epoch 20/45
300/300 [==============================] - 345s 1s/step - loss: 0.0308 - val_loss: 0.0251
Epoch 21/45
300/300 [==============================] - 338s 1s/step - loss: 0.0302 - val_loss: 0.0224
```

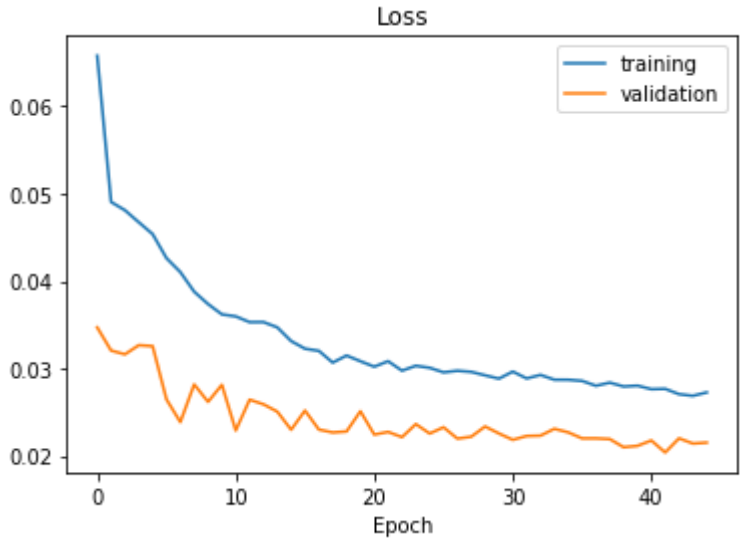
```
Epoch 22/45
300/300 [==============================] - 337s 1s/step - loss: 0.0308 - val_loss: 0.0227
Epoch 23/45
300/300 [==============================] - 337s 1s/step - loss: 0.0297 - val_loss: 0.0221
Epoch 24/45
300/300 [==============================] - 337s 1s/step - loss: 0.0303 - val_loss: 0.0236
Epoch 25/45
300/300 [==============================] - 337s 1s/step - loss: 0.0301 - val_loss: 0.0226
Epoch 26/45
300/300 [==============================] - 337s 1s/step - loss: 0.0296 - val_loss: 0.0233
Epoch 27/45
300/300 [==============================] - 337s 1s/step - loss: 0.0297 - val_loss: 0.0220
Epoch 28/45
300/300 [==============================] - 338s 1s/step - loss: 0.0296 - val_loss: 0.0222
Epoch 29/45
300/300 [==============================] - 339s 1s/step - loss: 0.0292 - val_loss: 0.0234
Epoch 30/45
300/300 [==============================] - 339s 1s/step - loss: 0.0288 - val_loss: 0.0226
Epoch 31/45
300/300 [==============================] - 339s 1s/step - loss: 0.0296 - val_loss: 0.0218
Epoch 32/45
300/300 [==============================] - 338s 1s/step - loss: 0.0289 - val_loss: 0.0223
Epoch 33/45
300/300 [==============================] - 343s 1s/step - loss: 0.0293 - val_loss: 0.0223
Epoch 34/45
300/300 [==============================] - 342s 1s/step - loss: 0.0287 - val_loss: 0.0231
Epoch 35/45
300/300 [==============================] - 340s 1s/step - loss: 0.0287 - val_loss: 0.0227
Epoch 36/45
300/300 [==============================] - 339s 1s/step - loss: 0.0286 - val_loss: 0.0220
Epoch 37/45
300/300 [==============================] - 340s 1s/step - loss: 0.0280 - val_loss: 0.0220
Epoch 38/45
300/300 [==============================] - 364s 1s/step - loss: 0.0284 - val_loss: 0.0219
Epoch 39/45
300/300 [==============================] - 352s 1s/step - loss: 0.0279 - val_loss: 0.0210
Epoch 40/45
300/300 [==============================] - 345s 1s/step - loss: 0.0280 - val_loss: 0.0211
Epoch 41/45
300/300 [==============================] - 337s 1s/step - loss: 0.0276 - val_loss: 0.0218
Epoch 42/45
300/300 [==============================] - 337s 1s/step - loss: 0.0277 - val_loss: 0.0204
```

```
Epoch 43/45
300/300 [==============================] - 338s 1s/step - loss: 0.0271 - val_loss: 0.0220
Epoch 44/45
300/300 [==============================] - 339s 1s/step - loss: 0.0269 - val_loss: 0.0214
Epoch 45/45
300/300 [==============================] - 338s 1s/step - loss: 0.0273 - val_loss: 0.0215
```

Plotting Model loss

In [77]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training', 'validation'])
plt.title('Loss')
plt.xlabel('Epoch')
```

Out[77]: Text(0.5, 0, 'Epoch')



Saving Model to File system.

In [78]:
```
model.save('Trail1_combined_Dropout_model2.h5')
```

End