Image Segementation on Indian Driving Dataset

1. Business Problem

1.1 Description

In computer vision, Image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

Indian Driving Dataset is a novel dataset for road scene understanding in unstructured environments. Unstructured environments usually corresponds to well-delineated infrastructure such as lanes, a small number of well-defined categories for traffic participants, low variation in object or background appearance and strong adherence to traffic rules.

1.2 Sources/Useful Links

- Source : https://idd.insaan.iiit.ac.in/ (https://idd.insaan.iiit.ac.in/)

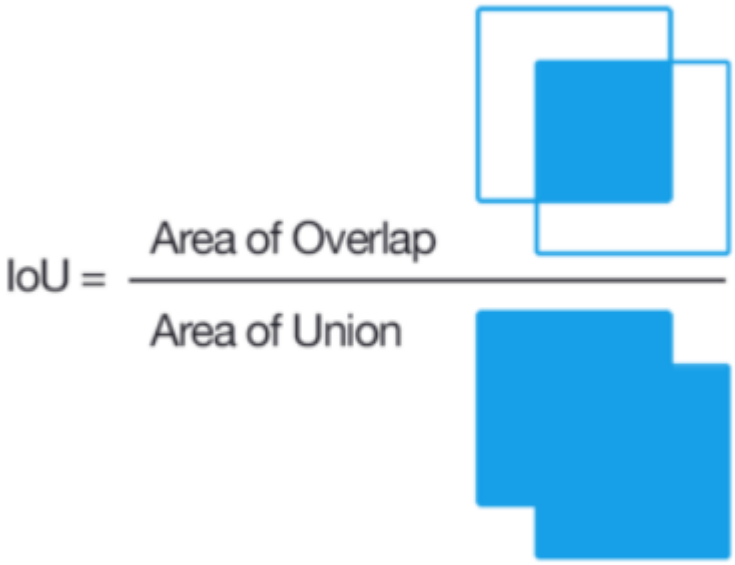
2. Data

2.1.1 Data Overview

- Indian Driving Dataset consists of 10,000 images, finely annotated with 34 classes collected from 182 drive sequences on Indian roads.
- The dataset consists of images obtained from a front facing camera attached to a car. The car was driven around Hyderabad, Bangalore cities and their outskirts.
- This case study would be implemented on IDD Lite dataset which contains 7 classes compared to 30 in IDD
- The 7 classes in IDD Lite dataset are :
  - drivable - 0
  - non-drivable - 1
  - living-thing - 2
  - 2-Wheeler, autorickshaw, large-vehicle - 3
  - barrier, structures - 4
  - construction - 5
  - vegetation , sky -6

2.1.2 Performance Metric

The performance metric would be mIoU - Mean Intersection Over Union also known as Jaccard Index. The mean IoU of the image is calculated by taking the IoU of each class and averaging them.



3. Data Analysis

In [0]:
```
import cv2, os, random
import numpy as np
import shutil
import pandas as pd
import matplotlib.pyplot as plt
```

In [0]:
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential,Model
from tensorflow.keras.layers import Dense, Dropout, Flatten,concatenate,Input
from tensorflow.keras.layers import Conv2D, MaxPooling2D,Conv1D
from tensorflow.keras import backend as K
from tensorflow.keras.layers import BatchNormalization


from numpy import asarray
from numpy import zeros
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Embedding
from tensorflow.keras.initializers import he_normal
from time import time
from tensorflow.keras.callbacks import TensorBoard

from tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler
from tensorflow.keras import backend as K
```

Data preparation

In [0]:
```
os. listdir('idd20k_lite')
```

Out[6]:
```
['gtFine', 'leftImg8bit']
```

In [0]:
```
print(os. listdir('idd20k_lite/gtFine'))
print(os. listdir('idd20k_lite/leftImg8bit'))
```

```
['train', 'val']
['train', 'val', 'test']
```

In [0]:
```
print(os. listdir('idd20k_lite/gtFine/train'))
```

```
['175', '528', '575', '203', '493', '473', '267', '448', '95', '547', '28
0', '173', '371', '263', '409', '143', '30', '1', '160', '277', '32', '49
8', '437', '237', '137', '146', '116', '539', '560', '299', '428', '151',
'177', '80', '29', '377', '477', '5', '490', '57', '168', '11', '76', '44
2', '65', '441', '93', '265', '406', '339', '446', '452', '438', '548', '33
1', '293', '34', '376', '522', '303', '578', '476', '258', '126', '460', '5
32', '322', '100', '91', '351', '115', '207', '321', '544', '108', '431',
'373', '491', '432', '306', '556', '433', '163', '174', '282', '235', '48
9', '216', '478', '561', '411', '162', '340', '520', '60', '87', '429',
'9', '482', '508', '98', '382', '252', '225', '513', '302', '266', '453',
'462', '63', '413', '512', '70', '301', '135', '457', '36', '220', '247',
'25', '81', '273', '502', '102', '529', '419', '157', '138', '573', '421',
'470', '464', '77', '164', '7', '204', '68', '475', '206', '145', '26', '33
6', '56', '503', '201', '59', '78', '84', '417', '400', '106', '154', '32
7', '449', '31', '94', '141', '46', '243', '387', '577', '218', '361', '26
2', '110', '178', '333', '245', '311', '48', '517', '410', '44', '236', '43
9', '563', '430', '372', '317', '269', '338', '155', '329', '101', '2', '46
7', '454', '550', '128', '353', '465', '211', '52', '133', '0', '213', '28
3', '479', '443', '540', '359', '347', '40', '261', '16', '104', '380', '23
1', '334', '230', '117', '85', '367', '152', '64', '423', '268', '171', '30
8', '288', '6', '320', '42', '285', '124', '176', '541', '275', '127', '35
5', '20', '543', '416', '357', '312', '455', '564', '469', '257', '139', '2
3', '136', '142', '505', '310', '45', '350', '403', '375', '37', '90', '22
3', '121', '248', '144', '140', '314', '422', '41', '125', '38', '414', '40
2', '295', '530', '554', '370', '209', '75', '551', '130', '250', '96', '22
4', '483', '471', '499', '315', '238', '298', '519', '72', '118', '131', '1
58', '79', '472', '49', '325', '10', '260', '122', '69', '170', '383', '35
6', '28', '501', '468', '43', '525', '156', '506', '39', '354', '316', '16
6', '53', '86']
```

In [0]:
```
print(os. listdir('idd20k_lite/gtFine/train/0'))
print(os. listdir('idd20k_lite/gtFine/train/1'))
print(os. listdir('idd20k_lite/gtFine/train/10'))
```

```
['024541_inst_label.png', '024703_inst_label.png', '024703_label.png', '024
541_label.png']
['502201_label.png', '662706_label.png', '725984_label.png', '820938_inst_l
abel.png', '502201_inst_label.png', '662706_inst_label.png', '340676_inst_l
abel.png', '725984_inst_label.png', '092468_inst_label.png', '601536_label.
png', '820938_label.png', '340676_label.png', '092468_label.png', '601536_i
nst_label.png']
['092196_label.png', '092196_inst_label.png']
```

In [0]:
```
print(os. listdir('idd20k_lite/leftImg8bit/train/0'))
print(os. listdir('idd20k_lite/leftImg8bit/train/1'))
print(os. listdir('idd20k_lite/leftImg8bit/train/10'))
```

```
['024541_image.jpg', '024703_image.jpg']
['601536_image.jpg', '502201_image.jpg', '820938_image.jpg', '662706_image.
jpg', '092468_image.jpg', '340676_image.jpg', '725984_image.jpg']
['092196_image.jpg']
```

- From the folder structure, we can see that the images are present in the path 'idd20k_lite/leftImg8bit/train/0' where the final folder varies
- From the folder structure, we can see that the labels are present in the path 'idd20k_lite/gtFine/train/0' where the final folder varies

Grouping all images into a single folder i.e., train,val

In [0]:
```
data = 'idd20k_lite/'

img_train = data + 'leftImg8bit/train/'
seg_train = data + 'gtFine/train/'

img_val = data + 'leftImg8bit/val/'
seg_val = data + 'gtFine/val/'

img_test = data + 'leftImg8bit/test/'
```

In [0]:
```
img_train_files = sorted(os.listdir(img_train))
seg_train_files = sorted(os.listdir(seg_train))
img_val_files = sorted(os.listdir(img_val))
seg_val_files = sorted(os.listdir(seg_val))
img_test_files = sorted(os.listdir(img_test))
```

In [0]:
```
# train images
path1='idd20k_lite1/leftImg8bit/train/'
for i in img_train_files:
    subpath = img_train + i+'/'
    for j in os.listdir(subpath):
        source=subpath+j
        destination=path1+j
        dest = shutil.copy(source, destination)

# val images
path1='idd20k_lite1/leftImg8bit/val/'
for i in img_val_files:
    subpath = img_val + i+'/'
    for j in os.listdir(subpath):
        source=subpath+j
        destination=path1+j
        dest = shutil.copy(source, destination)

# test images
path1='idd20k_lite1/leftImg8bit/test/'
for i in img_test_files:
    subpath = img_test + i+'/'
    for j in os.listdir(subpath):
        source=subpath+j
        destination=path1+j
        dest = shutil.copy(source, destination)

# train labels
path1='idd20k_lite1/gtFine/train/'
for i in seg_train_files:
    subpath = seg_train + i+'/'
    for j in os.listdir(subpath):
        source=subpath+j
        destination=path1+j
        dest = shutil.copy(source, destination)

# val labels
path1='idd20k_lite1/gtFine/val/'
for i in seg_val_files:
    subpath = seg_val + i+'/'
    for j in os.listdir(subpath):
        source=subpath+j
        destination=path1+j
        dest = shutil.copy(source, destination)
```

Data Prepearation completed

Loading the final data

In [0]:
```
data = 'idd20k_lite1/'

img_train = data + 'leftImg8bit/train/'
seg_train = data + 'gtFine/train/'

img_test = data + 'leftImg8bit/val/'
seg_test = data + 'gtFine/val/'
```

In [0]:
```
img_train_files = sorted(os.listdir(img_train))
seg_train_files = sorted(os.listdir(seg_train))
img_test_files = sorted(os.listdir(img_test))
seg_test_files = sorted(os.listdir(seg_test))
```

In [0]:
```
train_img = os.listdir(img_train)
train_img.sort()
train_seg = os.listdir(seg_train)
train_seg.sort()

test_img = os.listdir(img_test)
test_img.sort()
test_seg = os.listdir(seg_test)
test_seg.sort()
```

In [0]:
```
train_seg[:6]
```

Out[12]:
```
['0000002_inst_label.png',
 '0000002_label.png',
 '0000097_inst_label.png',
 '0000097_label.png',
 '0000192_inst_label.png',
 '0000192_label.png']
```

In [0]:
```
test_seg[:6]
```

Out[7]:
```
['0000000_inst_label.png',
 '0000000_label.png',
 '000065_inst_label.png',
 '000065_label.png',
 '0001080_inst_label.png',
 '0001080_label.png']
```

- The label images are of 2 types,semantic segementation and instance segmentation

In [0]:
```
# separating semantic and instance segmentation labels
train_seg_label=[]
train_inst_seg=[]
for i in range(1,len(train_seg)):
    if(i%2 !=0):
        train_seg_label.append(train_seg[i])
    else:
        train_inst_seg.append(train_seg[i])

test_seg_label=[]
test_inst_seg=[]
for i in range(len(test_seg)):
    if(i%2 !=0):
        test_seg_label.append(test_seg[i])
    else:
        test_inst_seg.append(test_seg[i])
```

In [0]:
```
print(train_img[:5])
print(train_seg_label[:5])
print(test_img[:5])
print(test_seg_label[:5])
```

```
['0000002_image.jpg', '0000097_image.jpg', '0000192_image.jpg', '0000215_im
age.jpg', '0000247_image.jpg']
['0000002_label.png', '0000097_label.png', '0000192_label.png', '0000215_la
bel.png', '0000247_label.png']
['0000000_image.jpg', '000065_image.jpg', '0001080_image.jpg', '000190_imag
e.jpg', '0001923_image.jpg']
['0000000_label.png', '000065_label.png', '0001080_label.png', '000190_labe
l.png', '0001923_label.png']
```

Number of images and size of image

In [0]:

```
for i in range(len(train_img)):
    first_img = cv2.imread('idd20k_lite1/leftImg8bit/train/'+train_img[0])
    img = cv2.imread('idd20k_lite1/leftImg8bit/train/'+train_img[i])
    if(first_img.shape == img.shape):
        if(i==len(train_img)-1):
            print('Number of train Images =',len(train_img))
            print('All Train Images have same shape')
            print('Shape of all Train images =',img.shape)
        continue
    else:
        print(train_img[i]+' has shape :',img.shape)
print('='*80)

for i in range(len(train_seg_label)):
    first_img = cv2.imread('idd20k_lite1/gtFine/train/'+train_seg_label[0])
    img = cv2.imread('idd20k_lite1/gtFine/train/'+train_seg_label[i])
    if(first_img.shape == img.shape):
        if(i==len(train_img)-1):
            print('Number of train labels =',len(train_seg_label))
            print('All Train Labels have same shape')
            print('Shape of all Train Labels =',img.shape)
        continue
    else:
        print(train_seg_label[i]+' has shape :',img.shape)
print('='*80)

for i in range(len(test_img)):
    first_img = cv2.imread('idd20k_lite1/leftImg8bit/val/'+test_img[0])
    img = cv2.imread('idd20k_lite1/leftImg8bit/val/'+test_img[i])
    if(first_img.shape == img.shape):
        if(i==len(test_img)-1):
            print('Number of Test Images =',len(test_img))
            print('All Test Images have same shape')
            print('Shape of all Test images =',img.shape)
        continue
    else:
        print(test_img[i]+' has shape :',img.shape)
print('='*80)
```

```
Number of train Images = 1380
All Train Images have same shape
Shape of all Train images = (227, 320, 3)
================================================================================
=====
Number of train labels = 1380
All Train Labels have same shape
Shape of all Train Labels = (227, 320, 3)
================================================================================
=====
Number of Test Images = 204
All Test Images have same shape
Shape of all Test images = (227, 320, 3)
================================================================================
=====
```

Observations :-

- All images and labels in train, validation and test have same size which is (227,320,3)
- All images and labels have 3 channels, i.e., all are of RGB format

Preparing final train and val datasets :-

In [0]:
```
from sklearn.model_selection import train_test_split

X_tr,X_cr, y_tr, y_cr = train_test_split(train_img, train_seg_label, test_siz
```

In [0]:
```
print(X_tr[:5])
print(y_tr[:5])
print(X_cr[:5])
print(y_cr[:5])
```

```
['0043118_image.jpg', '881557_image.jpg', '862328_image.jpg', '960225_imag
e.jpg', 'frame0731_image.jpg']
['0043118_label.png', '881557_label.png', '862328_label.png', '960225_labe
l.png', 'frame0731_label.png']
['355117_image.jpg', 'frame7284_image.jpg', '978273_image.jpg', 'frame7149_
image.jpg', '010441_image.jpg']
['355117_label.png', 'frame7284_label.png', '978273_label.png', 'frame7149_
label.png', '010441_label.png']
```

In [0]:
```
# Resizing image height to 256 beacuse to send as image as inout to uNET mode
# multiples of 32
height=256
width=320
n_classes=7
def prepare_image_data(path,data):
    src=path+data
    img = cv2.imread(src)
    img=cv2.resize(img,(width,height))
    img = np.float32(img)  / 255          #normalization
    return img

# https://github.com/advaitsave/Multiclass-Semantic-Segmentation-CamVid/blob/
def prepare_label_data(path,data):
    label = np.zeros((height, width, n_classes))
    src=path+data
    img = cv2.imread(src)
    img=cv2.resize(img,(width,height))
    img1=img[:,:,0]
    for i in range(n_classes):
        label[:,:,i] = (img1==i).astype(int)
    return label
```

In [0]:
```
X_train, y_train, X_val, y_val, X_test, y_test = [], [], [], [], [], []
```

In [0]:
```
for i in range(len(X_tr)):
    X_train.append(prepare_image_data(img_train,X_tr[i]))

for i in range(len(y_tr)):
    y_train.append(prepare_label_data(seg_train,y_tr[i]))

for i in range(len(X_cr)):
    X_val.append(prepare_image_data(img_train,X_cr[i]))

for i in range(len(y_cr)):
    y_val.append(prepare_label_data(seg_train,y_cr[i]))
```

In [0]:
```
X_train=np.array(X_train)
y_train=np.array(y_train)
X_val=np.array(X_val)
y_val=np.array(y_val)
```

In [0]:
```
for i in range(len(test_img)):
    X_test.append(prepare_image_data(img_test,X_cr[i]))

for i in range(len(test_seg)):
    y_test.append(prepare_label_data(seg_test,y_cr[i]))
X_test=np.array(X_val)
y_test=np.array(y_val)
```

In [0]:
```
print('Train Data : ')
print('Images-',X_train.shape)
print('Labels-',y_train.shape)
print('='*40)
print('Validation Data : ')
print('Images-',X_val.shape)
print('Labels-',y_val.shape)
print('='*40)
```

```
Train Data :
Images- (1035, 256, 320, 3)
Labels- (1035, 256, 320, 7)
========================================
Validation Data :
Images- (345, 256, 320, 3)
Labels- (345, 256, 320, 7)
========================================
```

Performance Metric :

- The performance metric would be mIoU - Mean Intersection Over Union also known as Jaccard Index. The mean IoU of the image is calculated by taking the IoU of each class and averaging

them.

In [0]:
```
tf.keras.backend.clear_session()
```

In [0]:
```
import random as rn
np.random.seed(24)
tf.random.set_seed(28)
rn.seed(12)
```

In [0]:
```
def IoU(y_val, y_pred):
    class_iou = []
    n_classes = 7

    y_predi = np.argmax(y_pred, axis=3)
    y_truei = np.argmax(y_val, axis=3)

    for c in range(n_classes):
        TP = np.sum((y_truei == c) & (y_predi == c))
        FP = np.sum((y_truei != c) & (y_predi == c))
        FN = np.sum((y_truei == c) & (y_predi != c))
        IoU = TP / float(TP + FP + FN)
        if(float(TP + FP + FN) == 0):
          IoU=TP/0.001
        class_iou.append(IoU)
    MIoU=sum(class_iou)/n_classes
    return MIoU
```

In [0]:
```
def miou( y_true, y_pred ) :
    score = tf.py_function( lambda y_true, y_pred : IoU( y_true, y_pred).asty
                        [y_true, y_pred],
                        'float32')
    return score
```

Model 1 : UNet (without Image augmentation)

In [0]:

```
def unet(pretrained_weights = None):

    inputs = Input(shape=(256, 320,3))
    print(inputs , inputs.shape)

    conv1 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    conv1 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    drop3 = Dropout(0.5)(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(drop3)

    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_init
    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_init
    drop4 = Dropout(0.5)(conv4)

    up5 = Conv2D(256, 2, activation = 'relu', padding = 'same', kernel_initia
    merge5 = concatenate([conv3,up5], axis = 3)
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init

    up6 = Conv2D(128, 2, activation = 'relu', padding = 'same', kernel_initia
    merge6 = concatenate([conv2,up6], axis = 3)
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init

    up7 = Conv2D(64, 2, activation = 'relu', padding = 'same', kernel_initial
    merge7 = concatenate([conv1,up7], axis = 3)
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi

    conv8 = Conv2D(7, 3, activation = 'relu', padding = 'same', kernel_initia
    out = (Activation('softmax'))(conv8)

    model = Model(inputs,out)

    return model
```

In [0]:

```
%load_ext tensorboard
```

In [0]:
```
unet_model1 = unet()
unet_model1.summary()
```

```
Tensor("input_1:0", shape=(None, 256, 320, 3), dtype=float32) (None, 256,
320, 3)
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
Connected to
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)      0
_________________________________________________________________
conv2d (Conv2D)              (None, 256, 320, 64)      1792
input_1[0][0]
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 256, 320, 64)      36928
conv2d[0][0]
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 160, 64)      0
conv2d_1[0][0]
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 160, 128      73856
max_pooling2d[0][0]
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 128, 160, 128      147584
conv2d_2[0][0]
_________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 64, 80, 128)     0
conv2d_3[0][0]
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 80, 256)       295168
max_pooling2d_1[0][0]
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 80, 256)       590080
conv2d_4[0][0]
_________________________________________________________________
dropout (Dropout)            (None, 64, 80, 256)       0
conv2d_5[0][0]
_________________________________________________________________
max_pooling2d_2 (MaxPooling2D) (None, 32, 40, 256)     0
dropout[0][0]
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 32, 40, 512)       1180160
max_pool
```

```
ing2d_2[0][0]
__________________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, 32, 40, 512)  2359808     conv2d_6
[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 32, 40, 512)  0           conv2d_7
[0][0]
__________________________________________________________________________________________________
up_sampling2d (UpSampling2D)    (None, 64, 80, 512)  0           dropout_
1[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 64, 80, 256)  524544      up_sampl
ing2d[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 64, 80, 512)  0           conv2d_5
[0][0]
                                                                 conv2d_8
[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, 64, 80, 256)  1179904     concaten
ate[0][0]
__________________________________________________________________________________________________
conv2d_10 (Conv2D)              (None, 64, 80, 256)  590080      conv2d_9
[0][0]
__________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 128, 160, 256 0           conv2d_1
0[0][0]
__________________________________________________________________________________________________
conv2d_11 (Conv2D)              (None, 128, 160, 128 131200      up_sampl
ing2d_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 128, 160, 256 0           conv2d_3
[0][0]
                                                                 conv2d_1
1[0][0]
__________________________________________________________________________________________________
conv2d_12 (Conv2D)              (None, 128, 160, 128 295040      concaten
ate_1[0][0]
__________________________________________________________________________________________________
conv2d_13 (Conv2D)              (None, 128, 160, 128 147584      conv2d_1
2[0][0]
__________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D)  (None, 256, 320, 128 0           conv2d_1
3[0][0]
```

```
__________________________________________________________________________________
_____________________________
conv2d_14 (Conv2D)              (None, 256, 320, 64) 32832       up_sampl
ing2d_2[0][0]
__________________________________________________________________________________
_____________________________
concatenate_2 (Concatenate)     (None, 256, 320, 128 0           conv2d_1
[0][0]
                                                                 conv2d_1
4[0][0]
__________________________________________________________________________________
_____________________________
conv2d_15 (Conv2D)              (None, 256, 320, 64) 73792       concaten
ate_2[0][0]
__________________________________________________________________________________
_____________________________
conv2d_16 (Conv2D)              (None, 256, 320, 64) 36928       conv2d_1
5[0][0]
__________________________________________________________________________________
_____________________________
conv2d_17 (Conv2D)              (None, 256, 320, 7)  4039        conv2d_1
6[0][0]
__________________________________________________________________________________
_____________________________
activation (Activation)         (None, 256, 320, 7)  0           conv2d_1
7[0][0]
==================================================================================
==========================
Total params: 7,701,319
Trainable params: 7,701,319
Non-trainable params: 0
__________________________________________________________________________________
_____________________________
```

In [0]:
```
log_dir_3 = os.path.join('unet_model1')
tensorboard_callback3 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_3)
```

In [0]:
```
unet_model1.compile(optimizer = Adam(0.0001), loss = 'categorical_crossentrop
```

In [0]:
```
x=X_train.shape[0]//3
x
```

Out[24]: 345

In [0]:
```
history3 = unet_model1.fit(X_train, y_train,steps_per_epoch=x,epochs=20,verbo
```

```
Epoch 1/20
345/345 [==============================] - 66s 192ms/step - loss: 1.0544
- miou: 0.2778 - val_loss: 0.8257 - val_miou: 0.3308
Epoch 2/20
345/345 [==============================] - 59s 172ms/step - loss: 0.8029
- miou: 0.3281 - val_loss: 0.7679 - val_miou: 0.3523
Epoch 3/20
345/345 [==============================] - 59s 171ms/step - loss: 0.7452
- miou: 0.3462 - val_loss: 0.7933 - val_miou: 0.3462
Epoch 4/20
345/345 [==============================] - 59s 172ms/step - loss: 0.7084
- miou: 0.3558 - val_loss: 0.7242 - val_miou: 0.3647
Epoch 5/20
345/345 [==============================] - 59s 172ms/step - loss: 0.6893
- miou: 0.3607 - val_loss: 0.7104 - val_miou: 0.3594
Epoch 6/20
345/345 [==============================] - 59s 171ms/step - loss: 0.6709
- miou: 0.3652 - val_loss: 0.6551 - val_miou: 0.3806
Epoch 7/20
345/345 [==============================] - 59s 171ms/step - loss: 0.6088
- miou: 0.4136 - val_loss: 0.5745 - val_miou: 0.4487
Epoch 8/20
345/345 [==============================] - 59s 171ms/step - loss: 0.5653
- miou: 0.4383 - val_loss: 0.5901 - val_miou: 0.4415
Epoch 9/20
345/345 [==============================] - 59s 172ms/step - loss: 0.5452
- miou: 0.4482 - val_loss: 0.5453 - val_miou: 0.4482
Epoch 10/20
345/345 [==============================] - 59s 172ms/step - loss: 0.5229
- miou: 0.4572 - val_loss: 0.5449 - val_miou: 0.4555
Epoch 11/20
345/345 [==============================] - 59s 172ms/step - loss: 0.5071
- miou: 0.4643 - val_loss: 0.5346 - val_miou: 0.4732
Epoch 12/20
345/345 [==============================] - 60s 174ms/step - loss: 0.5043
- miou: 0.4661 - val_loss: 0.5633 - val_miou: 0.4550
Epoch 13/20
345/345 [==============================] - 60s 173ms/step - loss: 0.4750
- miou: 0.4783 - val_loss: 0.4973 - val_miou: 0.4818
Epoch 14/20
345/345 [==============================] - 59s 172ms/step - loss: 0.4680
- miou: 0.4900 - val_loss: 0.5014 - val_miou: 0.4993
Epoch 15/20
345/345 [==============================] - 59s 171ms/step - loss: 0.4520
- miou: 0.4982 - val_loss: 0.5116 - val_miou: 0.4928
Epoch 16/20
345/345 [==============================] - 59s 171ms/step - loss: 0.4371
- miou: 0.5087 - val_loss: 0.5297 - val_miou: 0.4894
Epoch 17/20
345/345 [==============================] - 59s 171ms/step - loss: 0.4302
- miou: 0.5163 - val_loss: 0.4681 - val_miou: 0.5107
Epoch 18/20
345/345 [==============================] - 59s 171ms/step - loss: 0.4143
```

```
- miou: 0.5246 - val_loss: 0.4930 - val_miou: 0.4896
Epoch 19/20
345/345 [==============================] - 59s 172ms/step - loss: 0.4068
- miou: 0.5299 - val_loss: 0.4669 - val_miou: 0.5296
Epoch 20/20
345/345 [==============================] - 59s 172ms/step - loss: 0.4010
- miou: 0.5367 - val_loss: 0.4532 - val_miou: 0.5203
```

In [0]:
```
!kill 4601
```

In [0]:
```
%tensorboard --logdir unet_model1
```

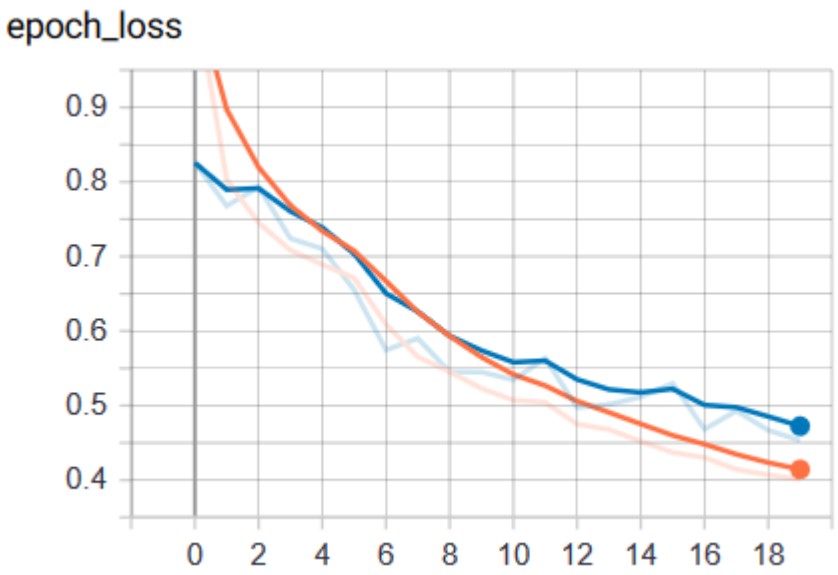
```
Reusing TensorBoard on port 6007 (pid 4601), started 0:07:19 ago. (Use '!ki
ll 4601' to kill it.)
```

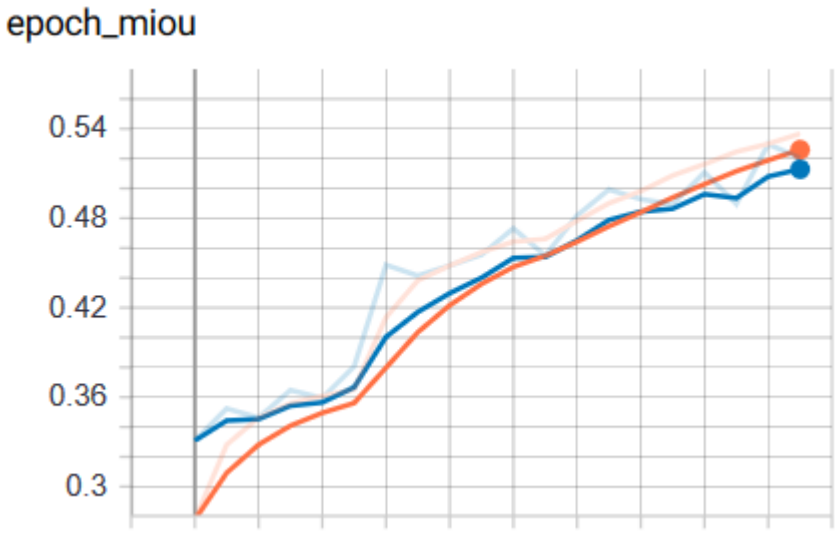
```
<IPython.core.display.Javascript object>
```

In [0]:
```
y_pred = unet_model1.predict(X_val)
print('MIoU for VGG16_UNet model is :',IoU(y_val, y_pred))
```

```
MIoU for VGG16_UNet model is : 0.5209974685655815
```

Tensorboard plots





Model 2 : UNet (with Image augmentation)

In [0]:
```
%load_ext tensorboard
```

In [0]:

```
def unet2(pretrained_weights = None):

    inputs = Input(shape=(256, 320,3))
    print(inputs , inputs.shape)

    conv1 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    conv1 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    conv2 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    drop3 = Dropout(0.5)(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(drop3)

    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_init
    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same', kernel_init
    drop4 = Dropout(0.5)(conv4)

    up5 = Conv2D(256, 2, activation = 'relu', padding = 'same', kernel_initia
    merge5 = concatenate([conv3,up5], axis = 3)
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same', kernel_init

    up6 = Conv2D(128, 2, activation = 'relu', padding = 'same', kernel_initia
    merge6 = concatenate([conv2,up6], axis = 3)
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same', kernel_init

    up7 = Conv2D(64, 2, activation = 'relu', padding = 'same', kernel_initial
    merge7 = concatenate([conv1,up7], axis = 3)
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same', kernel_initi

    conv8 = Conv2D(7, 3, activation = 'relu', padding = 'same', kernel_initia
    out = (Activation('softmax'))(conv8)

    model = Model(inputs,out)

    return model
```

In [0]:
```
unet_model2 = unet2()
unet_model2.summary()
```

```
Tensor("input_1:0", shape=(None, 256, 320, 3), dtype=float32) (None, 256, 3
20, 3)
Model: "model"
_________________________________________________________________________
_____________________
Layer (type)                 Output Shape              Param #     Connected
to
=========================================================================
=======================
input_1 (InputLayer)         [(None, 256, 320, 3) 0
_________________________________________________________________________
_____________________
conv2d (Conv2D)              (None, 256, 320, 64) 1792        input_1[0]
[0]
_________________________________________________________________________
_____________________
conv2d_1 (Conv2D)            (None, 256, 320, 64) 36928       conv2d[0]
[0]
_________________________________________________________________________
_____________________
max_pooling2d (MaxPooling2D) (None, 128, 160, 64) 0           conv2d_1
[0][0]
_________________________________________________________________________
_____________________
conv2d_2 (Conv2D)            (None, 128, 160, 128 73856       max_poolin
g2d[0][0]
_________________________________________________________________________
_____________________
conv2d_3 (Conv2D)            (None, 128, 160, 128 147584      conv2d_2
[0][0]
_________________________________________________________________________
_____________________
max_pooling2d_1 (MaxPooling2D) (None, 64, 80, 128)  0          conv2d_3
[0][0]
_________________________________________________________________________
_____________________
conv2d_4 (Conv2D)            (None, 64, 80, 256)  295168      max_poolin
g2d_1[0][0]
_________________________________________________________________________
_____________________
conv2d_5 (Conv2D)            (None, 64, 80, 256)  590080      conv2d_4
[0][0]
_________________________________________________________________________
_____________________
dropout (Dropout)            (None, 64, 80, 256)  0           conv2d_5
[0][0]
_________________________________________________________________________
_____________________
max_pooling2d_2 (MaxPooling2D) (None, 32, 40, 256)  0          dropout[0]
[0]
_________________________________________________________________________
_____________________
conv2d_6 (Conv2D)            (None, 32, 40, 512)  1180160     max_poolin
```

```
g2d_2[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_7 (Conv2D)               (None, 32, 40, 512)  2359808     conv2d_6
[0][0]
__________________________________________________________________________________________________
__________________________
dropout_1 (Dropout)             (None, 32, 40, 512)  0           conv2d_7
[0][0]
__________________________________________________________________________________________________
__________________________
up_sampling2d (UpSampling2D)    (None, 64, 80, 512)  0           dropout_1
[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_8 (Conv2D)               (None, 64, 80, 256)  524544      up_samplin
g2d[0][0]
__________________________________________________________________________________________________
__________________________
concatenate (Concatenate)       (None, 64, 80, 512)  0           conv2d_5
[0][0]
                                                                 conv2d_8
[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_9 (Conv2D)               (None, 64, 80, 256)  1179904     concatenat
e[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_10 (Conv2D)              (None, 64, 80, 256)  590080      conv2d_9
[0][0]
__________________________________________________________________________________________________
__________________________
up_sampling2d_1 (UpSampling2D)  (None, 128, 160, 256 0           conv2d_10
[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_11 (Conv2D)              (None, 128, 160, 128 131200      up_samplin
g2d_1[0][0]
__________________________________________________________________________________________________
__________________________
concatenate_1 (Concatenate)     (None, 128, 160, 256 0           conv2d_3
[0][0]
                                                                 conv2d_11
[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_12 (Conv2D)              (None, 128, 160, 128 295040      concatenat
e_1[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_13 (Conv2D)              (None, 128, 160, 128 147584      conv2d_12
[0][0]
__________________________________________________________________________________________________
__________________________
up_sampling2d_2 (UpSampling2D)  (None, 256, 320, 128 0           conv2d_13
[0][0]
```

```
__________________________________________________________________________________
__________________________
conv2d_14 (Conv2D)              (None, 256, 320, 64) 32832       up_samplin
g2d_2[0][0]
__________________________________________________________________________________
__________________________
concatenate_2 (Concatenate)     (None, 256, 320, 128 0           conv2d_1
[0][0]
                                                                 conv2d_14
[0][0]
__________________________________________________________________________________
__________________________
conv2d_15 (Conv2D)              (None, 256, 320, 64) 73792       concatenat
e_2[0][0]
__________________________________________________________________________________
__________________________
conv2d_16 (Conv2D)              (None, 256, 320, 64) 36928       conv2d_15
[0][0]
__________________________________________________________________________________
__________________________
conv2d_17 (Conv2D)              (None, 256, 320, 7)  4039        conv2d_16
[0][0]
__________________________________________________________________________________
__________________________
activation (Activation)         (None, 256, 320, 7)  0           conv2d_17
[0][0]
==================================================================================
======================
Total params: 7,701,319
Trainable params: 7,701,319
Non-trainable params: 0
__________________________________________________________________________________
__________________________
```

In [0]:
```
log_dir_4 = os.path.join('unet_model2')
tensorboard_callback4 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_4)
```

In [0]:
```
unet_model2.compile(optimizer = Adam(lr = 0.001), loss = 'categorical_crossen
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator

#Data Augmentation
datagen = ImageDataGenerator(rotation_range=30,width_shift_range=0.15, height
# prepare iterator
trainX_gen = datagen.flow(X_train,seed=1234)
trainY_gen = datagen.flow(y_train,seed=1234)
```

```
Using TensorFlow backend.
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/numpy_arra
y_iterator.py:127: UserWarning: NumpyArrayIterator is set to use the data f
ormat convention "channels_last" (channels on axis 3), i.e. expected either
1, 3, or 4 channels on axis 3. However, it was passed an array with shape
(1035, 256, 320, 7) (7 channels).
  str(self.x.shape[channels_axis]) + ' channels).')
```

In [0]:
```
train_generator = zip(trainX_gen, trainY_gen)
```

In [0]:
```
x=X_train.shape[0]//15
x
```

Out[30]: 69

In [0]:
```
history4=unet_model2.fit(train_generator,steps_per_epoch=x,epochs=25,verbose=
                         validation_data=(X_val, y_val),callbacks=[t
```

```
Epoch 1/25
69/69 [==============================] - 131s 2s/step - loss: 1.4133 - mi
ou: 0.1856 - val_loss: 1.1201 - val_miou: 0.2566
Epoch 2/25
69/69 [==============================] - 129s 2s/step - loss: 0.9855 - mi
ou: 0.2905 - val_loss: 0.9141 - val_miou: 0.3030
Epoch 3/25
69/69 [==============================] - 129s 2s/step - loss: 0.8039 - mi
ou: 0.3337 - val_loss: 0.7181 - val_miou: 0.3811
Epoch 4/25
69/69 [==============================] - 128s 2s/step - loss: 0.7054 - mi
ou: 0.3788 - val_loss: 0.6793 - val_miou: 0.3800
Epoch 5/25
69/69 [==============================] - 129s 2s/step - loss: 0.6826 - mi
ou: 0.3893 - val_loss: 0.6488 - val_miou: 0.4006
Epoch 6/25
69/69 [==============================] - 129s 2s/step - loss: 0.6525 - mi
ou: 0.4045 - val_loss: 0.6598 - val_miou: 0.4054
Epoch 7/25
69/69 [==============================] - 129s 2s/step - loss: 0.6377 - mi
ou: 0.4104 - val_loss: 0.6259 - val_miou: 0.4227
Epoch 8/25
69/69 [==============================] - 128s 2s/step - loss: 0.6291 - mi
ou: 0.4170 - val_loss: 0.6075 - val_miou: 0.4252
Epoch 9/25
69/69 [==============================] - 128s 2s/step - loss: 0.6037 - mi
ou: 0.4258 - val_loss: 0.5920 - val_miou: 0.4365
Epoch 10/25
69/69 [==============================] - 129s 2s/step - loss: 0.5933 - mi
ou: 0.4331 - val_loss: 0.5754 - val_miou: 0.4453
Epoch 11/25
69/69 [==============================] - 127s 2s/step - loss: 0.5846 - mi
ou: 0.4351 - val_loss: 0.5615 - val_miou: 0.4463
Epoch 12/25
69/69 [==============================] - 129s 2s/step - loss: 0.5752 - mi
ou: 0.4413 - val_loss: 0.5887 - val_miou: 0.4522
Epoch 13/25
69/69 [==============================] - 128s 2s/step - loss: 0.5680 - mi
ou: 0.4442 - val_loss: 0.5489 - val_miou: 0.4530
Epoch 14/25
69/69 [==============================] - 128s 2s/step - loss: 0.5530 - mi
ou: 0.4534 - val_loss: 0.5506 - val_miou: 0.4561
Epoch 15/25
69/69 [==============================] - 129s 2s/step - loss: 0.5433 - mi
ou: 0.4555 - val_loss: 0.5266 - val_miou: 0.4647
Epoch 16/25
69/69 [==============================] - 130s 2s/step - loss: 0.5386 - mi
ou: 0.4574 - val_loss: 0.5286 - val_miou: 0.4690
Epoch 17/25
69/69 [==============================] - 128s 2s/step - loss: 0.5336 - mi
ou: 0.4611 - val_loss: 0.5242 - val_miou: 0.4545
Epoch 18/25
69/69 [==============================] - 128s 2s/step - loss: 0.5314 - mi
```

```
ou: 0.4622 - val_loss: 0.5346 - val_miou: 0.4659
Epoch 19/25
69/69 [==============================] - 128s 2s/step - loss: 0.5297 - mi
ou: 0.4627 - val_loss: 0.5201 - val_miou: 0.4777
Epoch 20/25
69/69 [==============================] - 129s 2s/step - loss: 0.5033 - mi
ou: 0.4743 - val_loss: 0.4938 - val_miou: 0.4831
Epoch 21/25
69/69 [==============================] - 129s 2s/step - loss: 0.5014 - mi
ou: 0.4750 - val_loss: 0.4935 - val_miou: 0.4804
Epoch 22/25
69/69 [==============================] - 127s 2s/step - loss: 0.4976 - mi
ou: 0.4782 - val_loss: 0.4971 - val_miou: 0.4841
Epoch 23/25
69/69 [==============================] - 128s 2s/step - loss: 0.4879 - mi
ou: 0.4825 - val_loss: 0.4913 - val_miou: 0.4863
Epoch 24/25
69/69 [==============================] - 129s 2s/step - loss: 0.4792 - mi
ou: 0.4837 - val_loss: 0.4830 - val_miou: 0.4914
Epoch 25/25
69/69 [==============================] - 128s 2s/step - loss: 0.4893 - mi
ou: 0.4813 - val_loss: 0.4840 - val_miou: 0.4872
```

In [0]:

```
#running the model for anither 5 epochs
history4=unet_model2.fit(train_generator,steps_per_epoch=x,epochs=5,verbose=1
                                    validation_data=(X_val, y_val),callbacks=[t
```

```
Epoch 1/5
69/69 [==============================] - 129s 2s/step - loss: 0.4591 - mio
u: 0.4931 - val_loss: 0.4711 - val_miou: 0.4904
Epoch 2/5
69/69 [==============================] - 128s 2s/step - loss: 0.4699 - mio
u: 0.4890 - val_loss: 0.4728 - val_miou: 0.4923
Epoch 3/5
69/69 [==============================] - 127s 2s/step - loss: 0.4571 - mio
u: 0.4955 - val_loss: 0.4715 - val_miou: 0.4913
Epoch 4/5
69/69 [==============================] - 128s 2s/step - loss: 0.4545 - mio
u: 0.4968 - val_loss: 0.4776 - val_miou: 0.4934
Epoch 5/5
69/69 [==============================] - 128s 2s/step - loss: 0.4480 - mio
u: 0.5004 - val_loss: 0.4602 - val_miou: 0.4987
```

In [0]:

```
%tensorboard --logdir unet_model2
```

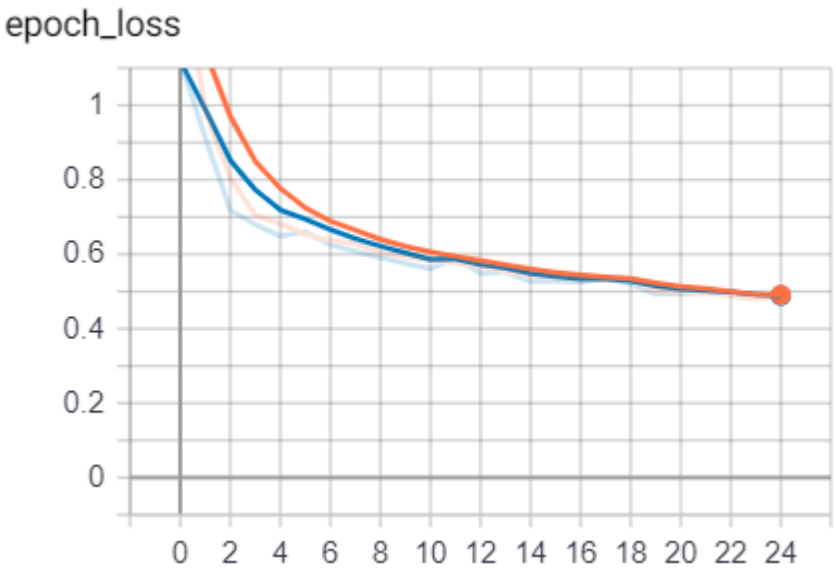
```
<IPython.core.display.Javascript object>
```

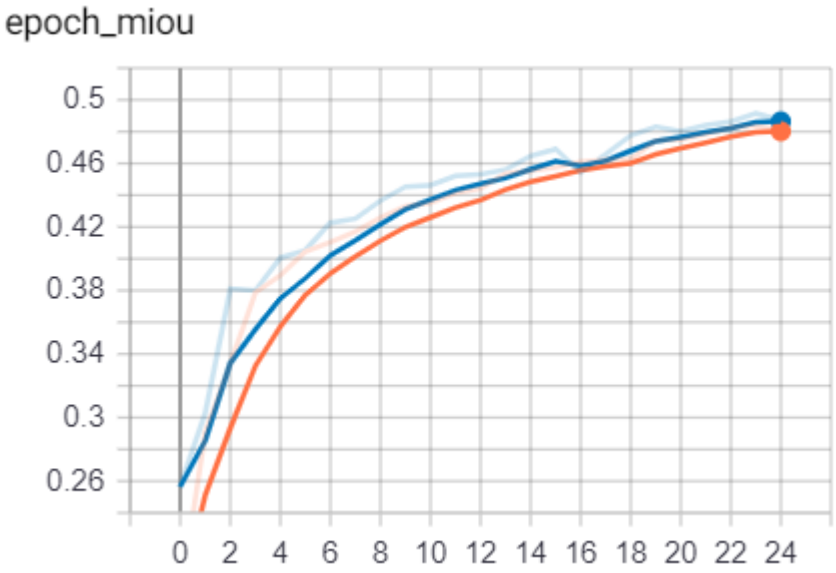
In [0]:

```
y_pred = unet_model2.predict(X_val)
print('MIoU for UNet model with Image augmentation is :',IoU(y_val, y_pred))
```

```
MIoU for UNet model with Image augmentation is : 0.49899015321575374
```

Tensorboard plots





VGG16(encoder)+Unet(decoder)

In [0]:
```
from tensorflow.keras.applications.vgg16 import VGG16
encoder_vgg16 = VGG16(input_shape =  (256, 320,3), include_top = False, weigh
encoder_vgg16.trainable = False
```

In [0]:
```
encoder_vgg16.summary()
```

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 320, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 320, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 160, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 160, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 160, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 80, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 80, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 80, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 80, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 40, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 40, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 40, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 40, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 20, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 8, 10, 512)        0
=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [0]:
```
for l in encoder_vgg16.layers:
    l.trainable = False
```

In [0]:
```
conv1 = encoder_vgg16.get_layer("block1_conv2").output
conv2 = encoder_vgg16.get_layer("block2_conv2").output
conv3 = encoder_vgg16.get_layer("block3_conv3").output
conv4 = encoder_vgg16.get_layer("block4_conv3").output

def vgg_unet1():

    up5 = Conv2D(256, 2, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed
    merge5 = concatenate([conv3,up5], axis = 3)
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed

    up6 = Conv2D(128, 2, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed=5
    merge6 = concatenate([conv2,up6], axis = 3)
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed

    up7 = Conv2D(64, 2, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed=5
    merge7 = concatenate([conv1,up7], axis = 3)
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed

    conv8 = Conv2D(7, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.glorot_normal(seed
    out = (Activation('softmax'))(conv8)

    model = Model(encoder_vgg16.input,out)

    return model
```

Model 3 : VGG16(encoder)+Unet(decoder) - [without image augmentation]

In [0]:
```
%load_ext tensorboard
```

In [0]:
```
tf.keras.backend.clear_session()
```

In [0]:
```
vgg_unet_model1 = vgg_unet1()
vgg_unet_model1.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
Connected to                     
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)      0         
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 320, 64)      1792      
input_1[0][0]                    
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 320, 64)      36928     
block1_conv1[0][0]               
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 160, 64)      0         
block1_conv2[0][0]               
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 160, 128      73856     
block1_pool[0][0]                
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 160, 128      147584    
block2_conv1[0][0]               
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 80, 128)       0         
block2_conv2[0][0]               
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 80, 256)       295168    
block2_pool[0][0]                
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 80, 256)       590080    
block3_conv1[0][0]               
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 80, 256)       590080    
block3_conv2[0][0]               
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 40, 256)       0         
block3_conv3[0][0]               
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 40, 512)       1180160   
block3_pool[0][0]                
_________________________________________________________________
```

```
__________________________
block4_conv2 (Conv2D)           (None, 32, 40, 512)  2359808     block4_con
v1[0][0]
__________________________________________________________________________
__________________________
block4_conv3 (Conv2D)           (None, 32, 40, 512)  2359808     block4_con
v2[0][0]
__________________________________________________________________________
__________________________
up_sampling2d (UpSampling2D)    (None, 64, 80, 512)  0           block4_con
v3[0][0]
__________________________________________________________________________
__________________________
conv2d (Conv2D)                 (None, 64, 80, 256)  524544      up_samplin
g2d[0][0]
__________________________________________________________________________
__________________________
concatenate (Concatenate)       (None, 64, 80, 512)  0           block3_con
v3[0][0]
                                                                 conv2d[0]
[0]
__________________________________________________________________________
__________________________
conv2d_1 (Conv2D)               (None, 64, 80, 256)  1179904     concatenat
e[0][0]
__________________________________________________________________________
__________________________
conv2d_2 (Conv2D)               (None, 64, 80, 256)  590080      conv2d_1
[0][0]
__________________________________________________________________________
__________________________
up_sampling2d_1 (UpSampling2D)  (None, 128, 160, 256 0           conv2d_2
[0][0]
__________________________________________________________________________
__________________________
conv2d_3 (Conv2D)               (None, 128, 160, 128 131200      up_samplin
g2d_1[0][0]
__________________________________________________________________________
__________________________
concatenate_1 (Concatenate)     (None, 128, 160, 256 0           block2_con
v2[0][0]
                                                                 conv2d_3
[0][0]
__________________________________________________________________________
__________________________
conv2d_4 (Conv2D)               (None, 128, 160, 128 295040      concatenat
e_1[0][0]
__________________________________________________________________________
__________________________
conv2d_5 (Conv2D)               (None, 128, 160, 128 147584      conv2d_4
[0][0]
__________________________________________________________________________
__________________________
up_sampling2d_2 (UpSampling2D)  (None, 256, 320, 128 0           conv2d_5
[0][0]
__________________________________________________________________________
__________________________
```

```
conv2d_6 (Conv2D)               (None, 256, 320, 64) 32832       up_samplin
g2d_2[0][0]
__________________________________________________________________________
_____________________
concatenate_2 (Concatenate)     (None, 256, 320, 128 0           block1_con
v2[0][0]
                                                                 conv2d_6
[0][0]
__________________________________________________________________________
_____________________
conv2d_7 (Conv2D)               (None, 256, 320, 64) 73792       concatenat
e_2[0][0]
__________________________________________________________________________
_____________________
conv2d_8 (Conv2D)               (None, 256, 320, 64) 36928       conv2d_7
[0][0]
__________________________________________________________________________
_____________________
conv2d_9 (Conv2D)               (None, 256, 320, 7)  4039        conv2d_8
[0][0]
__________________________________________________________________________
_____________________
activation (Activation)         (None, 256, 320, 7)  0           conv2d_9
[0][0]
==========================================================================
======================
Total params: 10,651,207
Trainable params: 3,015,943
Non-trainable params: 7,635,264
__________________________________________________________________________
_____________________
```

In [0]:
```
log_dir_1 = os.path.join('vgg_unet_model1')
tensorboard_callback1 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_1)
```

In [0]:
```
vgg_unet_model1.compile(optimizer = Adam(0.0001), loss = 'categorical_crossen
```

In [0]:
```
x=X_train.shape[0]//15
x
```

Out[24]: 69

In [0]:
```
history1 = vgg_unet_model1.fit(X_train, y_train,steps_per_epoch=x,epochs=25,v
```

```
Epoch 1/25
69/69 [==============================] - 50s 722ms/step - loss: 1.3459 -
miou: 0.1693 - val_loss: 1.1180 - val_miou: 0.2275
Epoch 2/25
69/69 [==============================] - 46s 668ms/step - loss: 1.1010 -
miou: 0.2326 - val_loss: 1.0689 - val_miou: 0.2267
Epoch 3/25
69/69 [==============================] - 46s 667ms/step - loss: 1.0661 -
miou: 0.2431 - val_loss: 1.0412 - val_miou: 0.2442
Epoch 4/25
69/69 [==============================] - 46s 662ms/step - loss: 1.0448 -
miou: 0.2486 - val_loss: 1.0366 - val_miou: 0.2488
Epoch 5/25
69/69 [==============================] - 46s 664ms/step - loss: 1.0333 -
miou: 0.2506 - val_loss: 1.0197 - val_miou: 0.2560
Epoch 6/25
69/69 [==============================] - 46s 664ms/step - loss: 1.0258 -
miou: 0.2528 - val_loss: 1.0120 - val_miou: 0.2518
Epoch 7/25
69/69 [==============================] - 46s 664ms/step - loss: 0.9940 -
miou: 0.2937 - val_loss: 0.9623 - val_miou: 0.3131
Epoch 8/25
69/69 [==============================] - 46s 665ms/step - loss: 0.9518 -
miou: 0.3203 - val_loss: 0.9398 - val_miou: 0.3277
Epoch 9/25
69/69 [==============================] - 46s 663ms/step - loss: 0.9369 -
miou: 0.3263 - val_loss: 0.9338 - val_miou: 0.3333
Epoch 10/25
69/69 [==============================] - 46s 662ms/step - loss: 0.9222 -
miou: 0.3324 - val_loss: 0.9251 - val_miou: 0.3307
Epoch 11/25
69/69 [==============================] - 46s 661ms/step - loss: 0.9170 -
miou: 0.3358 - val_loss: 0.9198 - val_miou: 0.3334
Epoch 12/25
69/69 [==============================] - 46s 662ms/step - loss: 0.9134 -
miou: 0.3404 - val_loss: 0.9137 - val_miou: 0.3456
Epoch 13/25
69/69 [==============================] - 46s 663ms/step - loss: 0.8970 -
miou: 0.3536 - val_loss: 0.9048 - val_miou: 0.3501
Epoch 14/25
69/69 [==============================] - 46s 663ms/step - loss: 0.8948 -
miou: 0.3590 - val_loss: 0.9196 - val_miou: 0.3558
Epoch 15/25
69/69 [==============================] - 46s 662ms/step - loss: 0.8965 -
miou: 0.3627 - val_loss: 0.9031 - val_miou: 0.3646
Epoch 16/25
69/69 [==============================] - 46s 664ms/step - loss: 0.8795 -
miou: 0.3719 - val_loss: 0.9069 - val_miou: 0.3671
Epoch 17/25
69/69 [==============================] - 46s 665ms/step - loss: 0.8754 -
miou: 0.3759 - val_loss: 0.8957 - val_miou: 0.3759
Epoch 18/25
69/69 [==============================] - 46s 666ms/step - loss: 0.8714 -
```

```
miou: 0.3780 - val_loss: 0.8880 - val_miou: 0.3794
Epoch 19/25
69/69 [==============================] - 46s 665ms/step - loss: 0.8097 -
miou: 0.4142 - val_loss: 0.5154 - val_miou: 0.5115
Epoch 20/25
69/69 [==============================] - 46s 666ms/step - loss: 0.4500 -
miou: 0.5418 - val_loss: 0.4649 - val_miou: 0.5374
Epoch 21/25
69/69 [==============================] - 46s 665ms/step - loss: 0.4159 -
miou: 0.5654 - val_loss: 0.4774 - val_miou: 0.5420
Epoch 22/25
69/69 [==============================] - 46s 666ms/step - loss: 0.4127 -
miou: 0.5681 - val_loss: 0.4524 - val_miou: 0.5540
Epoch 23/25
69/69 [==============================] - 46s 669ms/step - loss: 0.4012 -
miou: 0.5775 - val_loss: 0.4397 - val_miou: 0.5576
Epoch 24/25
69/69 [==============================] - 46s 669ms/step - loss: 0.3944 -
miou: 0.5806 - val_loss: 0.4392 - val_miou: 0.5489
Epoch 25/25
69/69 [==============================] - 46s 668ms/step - loss: 0.3881 -
miou: 0.5835 - val_loss: 0.4459 - val_miou: 0.5515
```

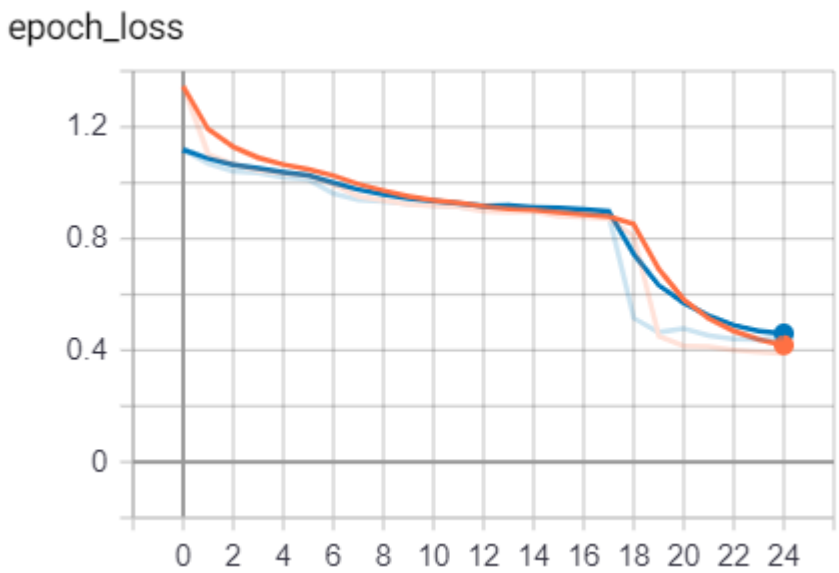
In [0]:
```
%tensorboard --logdir vgg_unet_model1
```

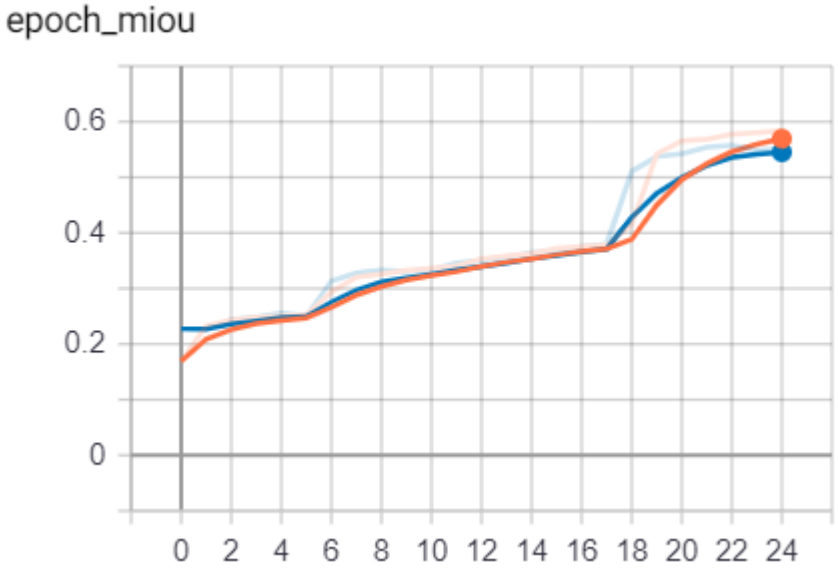
```
<IPython.core.display.Javascript object>
```

In [0]:
```
y_pred = vgg_unet_model1.predict(X_val)
print('MIoU for VGG16_UNet model is :',IoU(y_val, y_pred))
```

```
MIoU for VGG16_UNet model is : 0.5533828781195624
```

Tensorboard plots





Model 4 : VGG16(encoder)+Unet(decoder) - [with image augmentation]

In [0]:
```
%load_ext tensorboard
```

In [0]:

```
conv1 = encoder_vgg16.get_layer("block1_conv2").output
conv2 = encoder_vgg16.get_layer("block2_conv2").output
conv3 = encoder_vgg16.get_layer("block3_conv3").output
conv4 = encoder_vgg16.get_layer("block4_conv3").output

def vgg_unet2():

    up5 = Conv2D(256, 2, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)
    merge5 = concatenate([conv3,up5], axis = 3)
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)
    conv5 = Conv2D(256, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)

    up6 = Conv2D(128, 2, activation = 'relu', padding = 'same',
                kernel_initializer = keras.initializers.he_normal(seed=58))(
    merge6 = concatenate([conv2,up6], axis = 3)
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)
    conv6 = Conv2D(128, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)

    up7 = Conv2D(64, 2, activation = 'relu', padding = 'same',
                kernel_initializer = keras.initializers.he_normal(seed=58))(
    merge7 = concatenate([conv1,up7], axis = 3)
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)
    conv7 = Conv2D(64, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)

    conv8 = Conv2D(7, 3, activation = 'relu', padding = 'same',
                 kernel_initializer = keras.initializers.he_normal(seed=58)
    out = (Activation('softmax'))(conv8)

    model = Model(encoder_vgg16.input,out)

    return model
```

In [0]:
```
vgg_unet_model2 = vgg_unet2()
vgg_unet_model2.summary()
```

```
Model: "model"
_________________________________________________________________________________
_________________________
Layer (type)                 Output Shape              Param #     Connected
to
=================================================================================
=========================
input_1 (InputLayer)         [(None, 256, 320, 3) 0
_________________________________________________________________________________
_________________________
block1_conv1 (Conv2D)        (None, 256, 320, 64) 1792        input_1[0]
[0]
_________________________________________________________________________________
_________________________
block1_conv2 (Conv2D)        (None, 256, 320, 64) 36928       block1_con
v1[0][0]
_________________________________________________________________________________
_________________________
block1_pool (MaxPooling2D)   (None, 128, 160, 64) 0           block1_con
v2[0][0]
_________________________________________________________________________________
_________________________
block2_conv1 (Conv2D)        (None, 128, 160, 128 73856       block1_poo
l[0][0]
_________________________________________________________________________________
_________________________
block2_conv2 (Conv2D)        (None, 128, 160, 128 147584      block2_con
v1[0][0]
_________________________________________________________________________________
_________________________
block2_pool (MaxPooling2D)   (None, 64, 80, 128)  0           block2_con
v2[0][0]
_________________________________________________________________________________
_________________________
block3_conv1 (Conv2D)        (None, 64, 80, 256)  295168      block2_poo
l[0][0]
_________________________________________________________________________________
_________________________
block3_conv2 (Conv2D)        (None, 64, 80, 256)  590080      block3_con
v1[0][0]
_________________________________________________________________________________
_________________________
block3_conv3 (Conv2D)        (None, 64, 80, 256)  590080      block3_con
v2[0][0]
_________________________________________________________________________________
_________________________
block3_pool (MaxPooling2D)   (None, 32, 40, 256)  0           block3_con
v3[0][0]
_________________________________________________________________________________
_________________________
block4_conv1 (Conv2D)        (None, 32, 40, 512)  1180160     block3_poo
l[0][0]
_________________________________________________________________________________
```

```
__________________________
block4_conv2 (Conv2D)           (None, 32, 40, 512)  2359808     block4_con
v1[0][0]
__________________________________________________________________________
__________________________
block4_conv3 (Conv2D)           (None, 32, 40, 512)  2359808     block4_con
v2[0][0]
__________________________________________________________________________
__________________________
up_sampling2d (UpSampling2D)    (None, 64, 80, 512)  0           block4_con
v3[0][0]
__________________________________________________________________________
__________________________
conv2d (Conv2D)                 (None, 64, 80, 256)  524544      up_samplin
g2d[0][0]
__________________________________________________________________________
__________________________
concatenate (Concatenate)       (None, 64, 80, 512)  0           block3_con
v3[0][0]
                                                                 conv2d[0]
[0]
__________________________________________________________________________
__________________________
conv2d_1 (Conv2D)               (None, 64, 80, 256)  1179904     concatenat
e[0][0]
__________________________________________________________________________
__________________________
conv2d_2 (Conv2D)               (None, 64, 80, 256)  590080      conv2d_1
[0][0]
__________________________________________________________________________
__________________________
up_sampling2d_1 (UpSampling2D)  (None, 128, 160, 256 0           conv2d_2
[0][0]
__________________________________________________________________________
__________________________
conv2d_3 (Conv2D)               (None, 128, 160, 128 131200      up_samplin
g2d_1[0][0]
__________________________________________________________________________
__________________________
concatenate_1 (Concatenate)     (None, 128, 160, 256 0           block2_con
v2[0][0]
                                                                 conv2d_3
[0][0]
__________________________________________________________________________
__________________________
conv2d_4 (Conv2D)               (None, 128, 160, 128 295040      concatenat
e_1[0][0]
__________________________________________________________________________
__________________________
conv2d_5 (Conv2D)               (None, 128, 160, 128 147584      conv2d_4
[0][0]
__________________________________________________________________________
__________________________
up_sampling2d_2 (UpSampling2D)  (None, 256, 320, 128 0           conv2d_5
[0][0]
__________________________________________________________________________
__________________________
```

```
conv2d_6 (Conv2D)               (None, 256, 320, 64) 32832       up_samplin
g2d_2[0][0]
__________________________________________________________________________
_____________________
concatenate_2 (Concatenate)     (None, 256, 320, 128 0           block1_con
v2[0][0]
                                                                 conv2d_6
[0][0]
__________________________________________________________________________
_____________________
conv2d_7 (Conv2D)               (None, 256, 320, 64) 73792       concatenat
e_2[0][0]
__________________________________________________________________________
_____________________
conv2d_8 (Conv2D)               (None, 256, 320, 64) 36928       conv2d_7
[0][0]
__________________________________________________________________________
_____________________
conv2d_9 (Conv2D)               (None, 256, 320, 7)  4039        conv2d_8
[0][0]
__________________________________________________________________________
_____________________
activation (Activation)         (None, 256, 320, 7)  0           conv2d_9
[0][0]
==========================================================================
======================
Total params: 10,651,207
Trainable params: 3,015,943
Non-trainable params: 7,635,264
__________________________________________________________________________
_____________________
```

In [0]:
```
log_dir_2 = os.path.join('vgg_unet_model2')
tensorboard_callback2 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_2)
```

In [0]:
```
vgg_unet_model2.compile(optimizer = Adam(0.0001), loss = 'categorical_crossen
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator

#Data Augmentation
datagen = ImageDataGenerator(rotation_range=30,width_shift_range=0.15, height
# prepare iterator
trainX_gen = datagen.flow(X_train,seed=123)
trainY_gen = datagen.flow(y_train,seed=123)
```

```
Using TensorFlow backend.
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/numpy_arra
y_iterator.py:127: UserWarning: NumpyArrayIterator is set to use the data f
ormat convention "channels_last" (channels on axis 3), i.e. expected either
1, 3, or 4 channels on axis 3. However, it was passed an array with shape
(1035, 256, 320, 7) (7 channels).
  str(self.x.shape[channels_axis]) + ' channels).')
```

In [0]:
```
train_generator = zip(trainX_gen, trainY_gen)
```

In [0]:
```
x=X_train.shape[0]//15
x
```

Out[27]: 69

In [0]:
```
history2=vgg_unet_model2.fit(train_generator,steps_per_epoch=x,epochs=65,verb
                             validation_data=(X_val, y_val),callbacks=[t
```

```
Epoch 1/65
69/69 [==============================] - 137s 2s/step - loss: 1.5308 - mi
ou: 0.1673 - val_loss: 0.9904 - val_miou: 0.2897
Epoch 2/65
69/69 [==============================] - 135s 2s/step - loss: 0.9339 - mi
ou: 0.2950 - val_loss: 0.8465 - val_miou: 0.3123
Epoch 3/65
69/69 [==============================] - 134s 2s/step - loss: 0.8407 - mi
ou: 0.3232 - val_loss: 0.7646 - val_miou: 0.3432
Epoch 4/65
69/69 [==============================] - 133s 2s/step - loss: 0.7715 - mi
ou: 0.3456 - val_loss: 0.7306 - val_miou: 0.3463
Epoch 5/65
69/69 [==============================] - 133s 2s/step - loss: 0.7480 - mi
ou: 0.3510 - val_loss: 0.6909 - val_miou: 0.3582
Epoch 6/65
69/69 [==============================] - 134s 2s/step - loss: 0.7208 - mi
ou: 0.3594 - val_loss: 0.6728 - val_miou: 0.3634
Epoch 7/65
69/69 [==============================] - 133s 2s/step - loss: 0.6963 - mi
ou: 0.3633 - val_loss: 0.6606 - val_miou: 0.3749
Epoch 8/65
69/69 [==============================] - 134s 2s/step - loss: 0.6994 - mi
ou: 0.3647 - val_loss: 0.6768 - val_miou: 0.3728
Epoch 9/65
69/69 [==============================] - 132s 2s/step - loss: 0.6812 - mi
ou: 0.3679 - val_loss: 0.6408 - val_miou: 0.3765
Epoch 10/65
69/69 [==============================] - 133s 2s/step - loss: 0.6725 - mi
ou: 0.3710 - val_loss: 0.6308 - val_miou: 0.3814
Epoch 11/65
69/69 [==============================] - 132s 2s/step - loss: 0.6398 - mi
ou: 0.4076 - val_loss: 0.5735 - val_miou: 0.4490
Epoch 12/65
69/69 [==============================] - 132s 2s/step - loss: 0.5856 - mi
ou: 0.4448 - val_loss: 0.5681 - val_miou: 0.4528
Epoch 13/65
69/69 [==============================] - 134s 2s/step - loss: 0.5759 - mi
ou: 0.4471 - val_loss: 0.5472 - val_miou: 0.4663
Epoch 14/65
69/69 [==============================] - 134s 2s/step - loss: 0.5677 - mi
ou: 0.4519 - val_loss: 0.5423 - val_miou: 0.4595
Epoch 15/65
69/69 [==============================] - 133s 2s/step - loss: 0.5563 - mi
ou: 0.4530 - val_loss: 0.5343 - val_miou: 0.4623
Epoch 16/65
69/69 [==============================] - 134s 2s/step - loss: 0.5614 - mi
ou: 0.4537 - val_loss: 0.5391 - val_miou: 0.4638
Epoch 17/65
69/69 [==============================] - 133s 2s/step - loss: 0.5483 - mi
ou: 0.4570 - val_loss: 0.5275 - val_miou: 0.4634
Epoch 18/65
69/69 [==============================] - 133s 2s/step - loss: 0.5480 - mi
```

```
ou: 0.4588 - val_loss: 0.5241 - val_miou: 0.4644
Epoch 19/65
69/69 [==============================] - 134s 2s/step - loss: 0.5333 - mi
ou: 0.4625 - val_loss: 0.5071 - val_miou: 0.4763
Epoch 20/65
69/69 [==============================] - 134s 2s/step - loss: 0.5234 - mi
ou: 0.4665 - val_loss: 0.5103 - val_miou: 0.4775
Epoch 21/65
69/69 [==============================] - 133s 2s/step - loss: 0.5286 - mi
ou: 0.4641 - val_loss: 0.5166 - val_miou: 0.4762
Epoch 22/65
69/69 [==============================] - 131s 2s/step - loss: 0.5185 - mi
ou: 0.4694 - val_loss: 0.5019 - val_miou: 0.4806
Epoch 23/65
69/69 [==============================] - 132s 2s/step - loss: 0.5064 - mi
ou: 0.4844 - val_loss: 0.4815 - val_miou: 0.5026
Epoch 24/65
69/69 [==============================] - 133s 2s/step - loss: 0.4956 - mi
ou: 0.4921 - val_loss: 0.4767 - val_miou: 0.5087
Epoch 25/65
69/69 [==============================] - 133s 2s/step - loss: 0.4966 - mi
ou: 0.4908 - val_loss: 0.4831 - val_miou: 0.4944
Epoch 26/65
69/69 [==============================] - 130s 2s/step - loss: 0.4871 - mi
ou: 0.4961 - val_loss: 0.4853 - val_miou: 0.5142
Epoch 27/65
69/69 [==============================] - 131s 2s/step - loss: 0.4780 - mi
ou: 0.5045 - val_loss: 0.4532 - val_miou: 0.5369
Epoch 28/65
69/69 [==============================] - 131s 2s/step - loss: 0.4690 - mi
ou: 0.5272 - val_loss: 0.4507 - val_miou: 0.5454
Epoch 29/65
69/69 [==============================] - 133s 2s/step - loss: 0.4606 - mi
ou: 0.5322 - val_loss: 0.4528 - val_miou: 0.5523
Epoch 30/65
69/69 [==============================] - 133s 2s/step - loss: 0.4632 - mi
ou: 0.5319 - val_loss: 0.4490 - val_miou: 0.5444
Epoch 31/65
69/69 [==============================] - 132s 2s/step - loss: 0.4602 - mi
ou: 0.5322 - val_loss: 0.4452 - val_miou: 0.5511
Epoch 32/65
69/69 [==============================] - 135s 2s/step - loss: 0.4563 - mi
ou: 0.5363 - val_loss: 0.4489 - val_miou: 0.5542
Epoch 33/65
69/69 [==============================] - 133s 2s/step - loss: 0.4508 - mi
ou: 0.5390 - val_loss: 0.4606 - val_miou: 0.5296
Epoch 34/65
69/69 [==============================] - 134s 2s/step - loss: 0.4515 - mi
ou: 0.5383 - val_loss: 0.4402 - val_miou: 0.5461
Epoch 35/65
69/69 [==============================] - 135s 2s/step - loss: 0.4541 - mi
ou: 0.5394 - val_loss: 0.4359 - val_miou: 0.5523
Epoch 36/65
69/69 [==============================] - 135s 2s/step - loss: 0.4413 - mi
ou: 0.5453 - val_loss: 0.4550 - val_miou: 0.5550
Epoch 37/65
69/69 [==============================] - 135s 2s/step - loss: 0.4347 - mi
```

```
ou: 0.5525 - val_loss: 0.4339 - val_miou: 0.5586
Epoch 38/65
69/69 [==============================] - 134s 2s/step - loss: 0.4396 - mi
ou: 0.5505 - val_loss: 0.4361 - val_miou: 0.5456
Epoch 39/65
69/69 [==============================] - 135s 2s/step - loss: 0.4277 - mi
ou: 0.5538 - val_loss: 0.4233 - val_miou: 0.5681
Epoch 40/65
69/69 [==============================] - 135s 2s/step - loss: 0.4273 - mi
ou: 0.5541 - val_loss: 0.4290 - val_miou: 0.5642
Epoch 41/65
69/69 [==============================] - 134s 2s/step - loss: 0.4229 - mi
ou: 0.5607 - val_loss: 0.4364 - val_miou: 0.5629
Epoch 42/65
69/69 [==============================] - 135s 2s/step - loss: 0.4230 - mi
ou: 0.5619 - val_loss: 0.4271 - val_miou: 0.5563
Epoch 43/65
69/69 [==============================] - 134s 2s/step - loss: 0.4273 - mi
ou: 0.5593 - val_loss: 0.4198 - val_miou: 0.5673
Epoch 44/65
69/69 [==============================] - 133s 2s/step - loss: 0.4266 - mi
ou: 0.5598 - val_loss: 0.4357 - val_miou: 0.5577
Epoch 45/65
69/69 [==============================] - 132s 2s/step - loss: 0.4171 - mi
ou: 0.5645 - val_loss: 0.4335 - val_miou: 0.5564
Epoch 46/65
69/69 [==============================] - 132s 2s/step - loss: 0.4107 - mi
ou: 0.5694 - val_loss: 0.4210 - val_miou: 0.5715
Epoch 47/65
69/69 [==============================] - 133s 2s/step - loss: 0.4166 - mi
ou: 0.5684 - val_loss: 0.4196 - val_miou: 0.5772
Epoch 48/65
69/69 [==============================] - 132s 2s/step - loss: 0.4100 - mi
ou: 0.5711 - val_loss: 0.4343 - val_miou: 0.5693
Epoch 49/65
69/69 [==============================] - 132s 2s/step - loss: 0.4113 - mi
ou: 0.5717 - val_loss: 0.4517 - val_miou: 0.5563
Epoch 50/65
69/69 [==============================] - 132s 2s/step - loss: 0.4089 - mi
ou: 0.5713 - val_loss: 0.4136 - val_miou: 0.5751
Epoch 51/65
69/69 [==============================] - 133s 2s/step - loss: 0.4005 - mi
ou: 0.5774 - val_loss: 0.4203 - val_miou: 0.5733
Epoch 52/65
69/69 [==============================] - 133s 2s/step - loss: 0.4051 - mi
ou: 0.5763 - val_loss: 0.4136 - val_miou: 0.5726
Epoch 53/65
69/69 [==============================] - 132s 2s/step - loss: 0.4023 - mi
ou: 0.5760 - val_loss: 0.4172 - val_miou: 0.5744
Epoch 54/65
69/69 [==============================] - 133s 2s/step - loss: 0.3914 - mi
ou: 0.5853 - val_loss: 0.4162 - val_miou: 0.5662
Epoch 55/65
69/69 [==============================] - 130s 2s/step - loss: 0.3937 - mi
ou: 0.5831 - val_loss: 0.4500 - val_miou: 0.5612
Epoch 56/65
69/69 [==============================] - 131s 2s/step - loss: 0.3975 - mi
```

```
ou: 0.5817 - val_loss: 0.4249 - val_miou: 0.5812
Epoch 57/65
69/69 [==============================] - 132s 2s/step - loss: 0.3962 - mi
ou: 0.5818 - val_loss: 0.4264 - val_miou: 0.5748
Epoch 58/65
69/69 [==============================] - 133s 2s/step - loss: 0.3921 - mi
ou: 0.5864 - val_loss: 0.4189 - val_miou: 0.5712
Epoch 59/65
69/69 [==============================] - 131s 2s/step - loss: 0.3902 - mi
ou: 0.5888 - val_loss: 0.4234 - val_miou: 0.5688
Epoch 60/65
69/69 [==============================] - 131s 2s/step - loss: 0.3892 - mi
ou: 0.5887 - val_loss: 0.4115 - val_miou: 0.5816
Epoch 61/65
69/69 [==============================] - 132s 2s/step - loss: 0.3938 - mi
ou: 0.5845 - val_loss: 0.4097 - val_miou: 0.5749
Epoch 62/65
69/69 [==============================] - 133s 2s/step - loss: 0.3851 - mi
ou: 0.5905 - val_loss: 0.4141 - val_miou: 0.5765
Epoch 63/65
69/69 [==============================] - 132s 2s/step - loss: 0.3827 - mi
ou: 0.5928 - val_loss: 0.4214 - val_miou: 0.5779
Epoch 64/65
69/69 [==============================] - 134s 2s/step - loss: 0.3847 - mi
ou: 0.5906 - val_loss: 0.4076 - val_miou: 0.5854
Epoch 65/65
69/69 [==============================] - 134s 2s/step - loss: 0.3820 - mi
ou: 0.5948 - val_loss: 0.4248 - val_miou: 0.5721
```

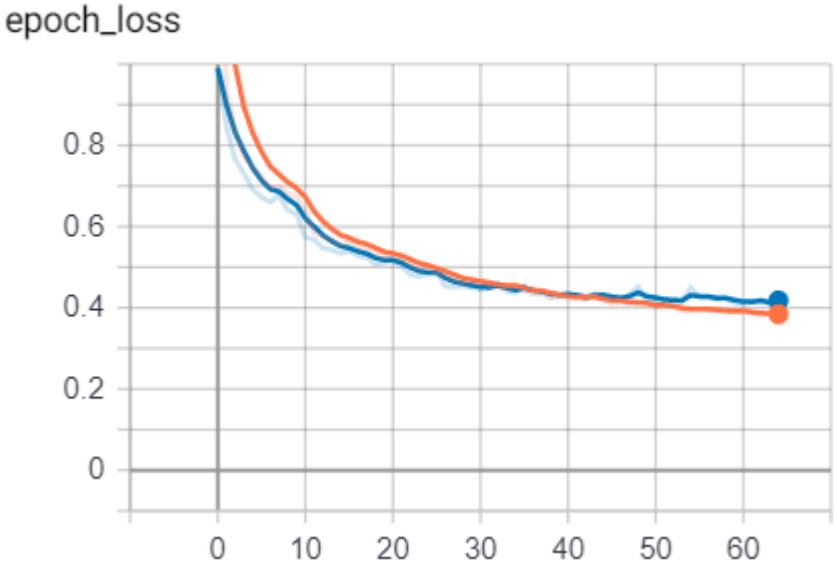
In [0]:
```
%tensorboard --logdir vgg_unet_model2
```

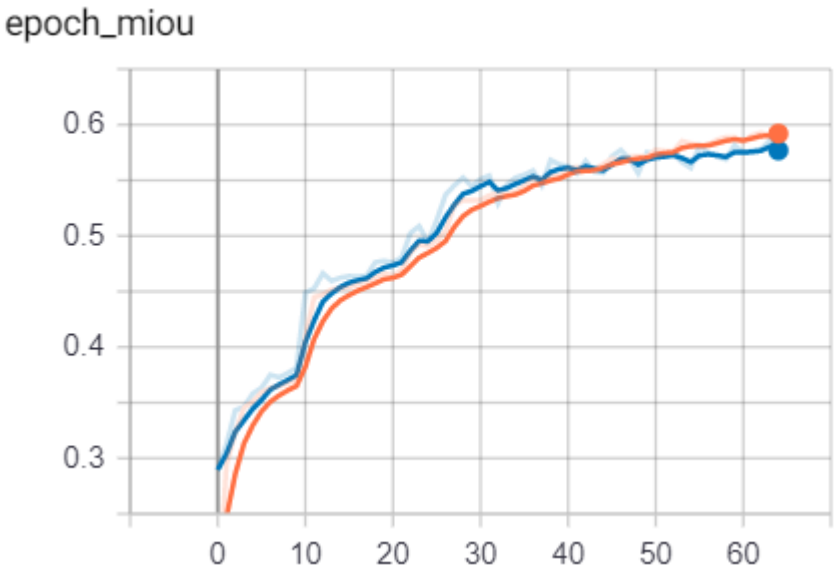
```
Reusing TensorBoard on port 6006 (pid 11950), started 0:00:52 ago. (Use '!k
ill 11950' to kill it.)

<IPython.core.display.Javascript object>
```

In [0]:
```
y_pred = vgg_unet_model2.predict(X_val)
print('MIoU for VGG16_UNet model with Image augmentation is :',IoU(y_val, y_p
```

```
MIoU for VGG16_UNet model with Image augmentation is : 0.5732448016046646
```

Tensorboard plots





Model 5 : SegNet (without Image augmentation)

In [0]:
```
tf.keras.backend.clear_session()
```

In [0]:

```
%load_ext tensorboard
def SegNet1():
    inputs = Input(shape=(256, 320,3))

    # Encoder
    conv1 = Convolution2D(64, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(inputs)
    conv1 = BatchNormalization()(conv1)
    conv1 = Activation("relu")(conv1)
    conv2 = Convolution2D(64, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(conv1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Activation("relu")(conv2)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv2)

    conv3 = Convolution2D(128, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(pool1)
    conv3 = BatchNormalization()(conv3)
    conv3 = Activation("relu")(conv3)
    conv4 = Convolution2D(128, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(conv3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Activation("relu")(conv4)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv4)

    conv5 = Convolution2D(256, 3, padding="same", kernel_initializer = keras.
    conv5 = BatchNormalization()(conv5)
    conv5 = Activation("relu")(conv5)
    conv6 = Convolution2D(256, 3, padding="same", kernel_initializer = keras.
    conv6 = BatchNormalization()(conv6)
    conv6 = Activation("relu")(conv6)
    conv7 = Convolution2D(256, 3, padding="same", kernel_initializer = keras.
    conv7 = BatchNormalization()(conv7)
    conv7 = Activation("relu")(conv7)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv7)

    conv8 = Convolution2D(512, 3, padding="same", kernel_initializer = keras.
    conv8 = BatchNormalization()(conv8)
    conv8 = Activation("relu")(conv8)
    conv9 = Convolution2D(512, 3, padding="same", kernel_initializer = keras.
    conv9 = BatchNormalization()(conv9)
    conv9 = Activation("relu")(conv9)
    conv10 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv10 = BatchNormalization()(conv10)
    conv10 = Activation("relu")(conv10)
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv10)

    # Decoder
    up1 = UpSampling2D(size=(2, 2))(pool4)
    conv11 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv11 = BatchNormalization()(conv11)
    conv11 = Activation("relu")(conv11)
    conv12 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv12 = BatchNormalization()(conv12)
    conv12 = Activation("relu")(conv12)
    conv13 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
```

```
    conv13 = BatchNormalization()(conv13)
    conv13 = Activation("relu")(conv13)

    up2 = UpSampling2D(size=(2, 2))(conv13)
    conv14 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv14 = BatchNormalization()(conv14)
    conv14 = Activation("relu")(conv14)
    conv15 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv15 = BatchNormalization()(conv15)
    conv15 = Activation("relu")(conv15)
    conv16 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv16 = BatchNormalization()(conv16)
    conv16 = Activation("relu")(conv16)

    up3 = UpSampling2D(size=(2, 2))(conv16)
    conv17 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv17 = BatchNormalization()(conv17)
    conv17 = Activation("relu")(conv17)
    conv18 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv18 = BatchNormalization()(conv18)
    conv18 = Activation("relu")(conv18)

    up4 = UpSampling2D(size=(2, 2))(conv18)
    conv19 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv19 = BatchNormalization()(conv19)
    conv19 = Activation("relu")(conv19)
    conv20 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv20 = BatchNormalization()(conv20)
    conv20 = Activation("relu")(conv20)

    conv21 = Convolution2D(7, 3, padding="same", kernel_initializer = keras.i
    out = Activation("softmax")(conv21)
    model = Model(inputs,out)
    return model
```

In [0]:
```
segnet_model1 = SegNet1()
segnet_model1.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 256, 320, 64)      1792
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 320, 64)      256
_________________________________________________________________
activation (Activation)      (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 256, 320, 64)      36928
_________________________________________________________________
batch_normalization_1 (Batch (None, 256, 320, 64)      256
_________________________________________________________________
activation_1 (Activation)    (None, 256, 320, 64)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 160, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 160, 128)     73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 128, 160, 128)     512
_________________________________________________________________
activation_2 (Activation)    (None, 128, 160, 128)     0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 128, 160, 128)     147584
_________________________________________________________________
batch_normalization_3 (Batch (None, 128, 160, 128)     512
_________________________________________________________________
activation_3 (Activation)    (None, 128, 160, 128)     0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 80, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 80, 256)       295168
_________________________________________________________________
batch_normalization_4 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_4 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 80, 256)       590080
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_5 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 80, 256)       590080
_________________________________________________________________
batch_normalization_6 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_6 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 40, 256)       0
```

conv2d_7 (Conv2D)	(None, 32, 40, 512)	1180160
batch_normalization_7 (Batch	(None, 32, 40, 512)	2048
activation_7 (Activation)	(None, 32, 40, 512)	0
conv2d_8 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_8 (Batch	(None, 32, 40, 512)	2048
activation_8 (Activation)	(None, 32, 40, 512)	0
conv2d_9 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_9 (Batch	(None, 32, 40, 512)	2048
activation_9 (Activation)	(None, 32, 40, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 16, 20, 512)	0
up_sampling2d (UpSampling2D)	(None, 32, 40, 512)	0
conv2d_10 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_10 (Batc	(None, 32, 40, 512)	2048
activation_10 (Activation)	(None, 32, 40, 512)	0
conv2d_11 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_11 (Batc	(None, 32, 40, 512)	2048
activation_11 (Activation)	(None, 32, 40, 512)	0
conv2d_12 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_12 (Batc	(None, 32, 40, 512)	2048
activation_12 (Activation)	(None, 32, 40, 512)	0
up_sampling2d_1 (UpSampling2	(None, 64, 80, 512)	0
conv2d_13 (Conv2D)	(None, 64, 80, 256)	1179904
batch_normalization_13 (Batc	(None, 64, 80, 256)	1024
activation_13 (Activation)	(None, 64, 80, 256)	0
conv2d_14 (Conv2D)	(None, 64, 80, 256)	590080
batch_normalization_14 (Batc	(None, 64, 80, 256)	1024
activation_14 (Activation)	(None, 64, 80, 256)	0
conv2d_15 (Conv2D)	(None, 64, 80, 256)	590080

```
batch_normalization_15 (Batc (None, 64, 80, 256)       1024
_________________________________________________________________
activation_15 (Activation)   (None, 64, 80, 256)       0
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 128, 160, 256)     0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 128, 160, 128)     295040
_________________________________________________________________
batch_normalization_16 (Batc (None, 128, 160, 128)     512
_________________________________________________________________
activation_16 (Activation)   (None, 128, 160, 128)     0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 128, 160, 128)     147584
_________________________________________________________________
batch_normalization_17 (Batc (None, 128, 160, 128)     512
_________________________________________________________________
activation_17 (Activation)   (None, 128, 160, 128)     0
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 256, 320, 128)     0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 256, 320, 64)      73792
_________________________________________________________________
batch_normalization_18 (Batc (None, 256, 320, 64)      256
_________________________________________________________________
activation_18 (Activation)   (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 256, 320, 64)      36928
_________________________________________________________________
batch_normalization_19 (Batc (None, 256, 320, 64)      256
_________________________________________________________________
activation_19 (Activation)   (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 256, 320, 7)       4039
_________________________________________________________________
activation_20 (Activation)   (None, 256, 320, 7)       0
=================================================================
Total params: 17,653,639
Trainable params: 17,642,887
Non-trainable params: 10,752
_________________________________________________________________
```

In [0]:
```
log_dir_5 = os.path.join('segnet_model1')
tensorboard_callback5 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_5)
```

In [0]:
```
segnet_model1.compile(optimizer = Adam(0.001), loss = 'categorical_crossentro
```

In [0]:
```
x=X_train.shape[0]//3
x
```

Out[27]: 345

In [0]:
```
history3 = segnet_model1.fit(X_train, y_train,steps_per_epoch=x,epochs=25,ver
```

```
Epoch 1/25
345/345 [==============================] - 68s 196ms/step - loss: 0.9505 -
miou: 0.2910 - val_loss: 6.7652 - val_miou: 0.0673
Epoch 2/25
345/345 [==============================] - 61s 178ms/step - loss: 0.7868 -
miou: 0.3371 - val_loss: 0.7881 - val_miou: 0.3540
Epoch 3/25
345/345 [==============================] - 62s 179ms/step - loss: 0.7349 -
miou: 0.3563 - val_loss: 6.5625 - val_miou: 0.1540
Epoch 4/25
345/345 [==============================] - 61s 178ms/step - loss: 0.7040 -
miou: 0.3666 - val_loss: 0.8301 - val_miou: 0.3652
Epoch 5/25
345/345 [==============================] - 62s 179ms/step - loss: 0.6505 -
miou: 0.3885 - val_loss: 0.7425 - val_miou: 0.3666
Epoch 6/25
345/345 [==============================] - 61s 178ms/step - loss: 0.6371 -
miou: 0.3981 - val_loss: 0.6560 - val_miou: 0.3946
Epoch 7/25
345/345 [==============================] - 61s 177ms/step - loss: 0.6039 -
miou: 0.4102 - val_loss: 0.6283 - val_miou: 0.4168
Epoch 8/25
345/345 [==============================] - 61s 177ms/step - loss: 0.6037 -
miou: 0.4137 - val_loss: 0.5720 - val_miou: 0.4259
Epoch 9/25
345/345 [==============================] - 61s 177ms/step - loss: 0.5972 -
miou: 0.4181 - val_loss: 0.6104 - val_miou: 0.4189
Epoch 10/25
345/345 [==============================] - 61s 176ms/step - loss: 0.5655 -
miou: 0.4315 - val_loss: 0.6193 - val_miou: 0.4347
Epoch 11/25
345/345 [==============================] - 61s 177ms/step - loss: 0.5513 -
miou: 0.4400 - val_loss: 0.6511 - val_miou: 0.4150
Epoch 12/25
345/345 [==============================] - 61s 176ms/step - loss: 0.5364 -
miou: 0.4472 - val_loss: 0.5533 - val_miou: 0.4504
Epoch 13/25
345/345 [==============================] - 61s 177ms/step - loss: 0.5238 -
miou: 0.4514 - val_loss: 0.5519 - val_miou: 0.4628
Epoch 14/25
345/345 [==============================] - 61s 177ms/step - loss: 0.5270 -
miou: 0.4524 - val_loss: 0.8180 - val_miou: 0.3927
Epoch 15/25
345/345 [==============================] - 61s 177ms/step - loss: 0.5102 -
miou: 0.4608 - val_loss: 0.5410 - val_miou: 0.4829
Epoch 16/25
345/345 [==============================] - 61s 176ms/step - loss: 0.5074 -
miou: 0.4630 - val_loss: 1.0845 - val_miou: 0.3394
Epoch 17/25
345/345 [==============================] - 61s 177ms/step - loss: 0.4821 -
miou: 0.4741 - val_loss: 0.5617 - val_miou: 0.4645
Epoch 18/25
345/345 [==============================] - 61s 177ms/step - loss: 0.4744 -
```

```
miou: 0.4773 - val_loss: 0.5014 - val_miou: 0.4760
Epoch 19/25
345/345 [==============================] - 61s 177ms/step - loss: 0.4609 -
miou: 0.4882 - val_loss: 0.4695 - val_miou: 0.5076
Epoch 20/25
345/345 [==============================] - 61s 178ms/step - loss: 0.4564 -
miou: 0.4895 - val_loss: 0.4844 - val_miou: 0.5030
Epoch 21/25
345/345 [==============================] - 61s 177ms/step - loss: 0.4604 -
miou: 0.4926 - val_loss: 0.4814 - val_miou: 0.5138
Epoch 22/25
345/345 [==============================] - 61s 176ms/step - loss: 0.4287 -
miou: 0.5157 - val_loss: 0.4477 - val_miou: 0.5378
Epoch 23/25
345/345 [==============================] - 61s 177ms/step - loss: 0.4305 -
miou: 0.5215 - val_loss: 0.4560 - val_miou: 0.5214
Epoch 24/25
345/345 [==============================] - 61s 176ms/step - loss: 0.4299 -
miou: 0.5237 - val_loss: 0.4562 - val_miou: 0.5337
Epoch 25/25
345/345 [==============================] - 61s 176ms/step - loss: 0.4129 -
miou: 0.5366 - val_loss: 0.4635 - val_miou: 0.5419
```

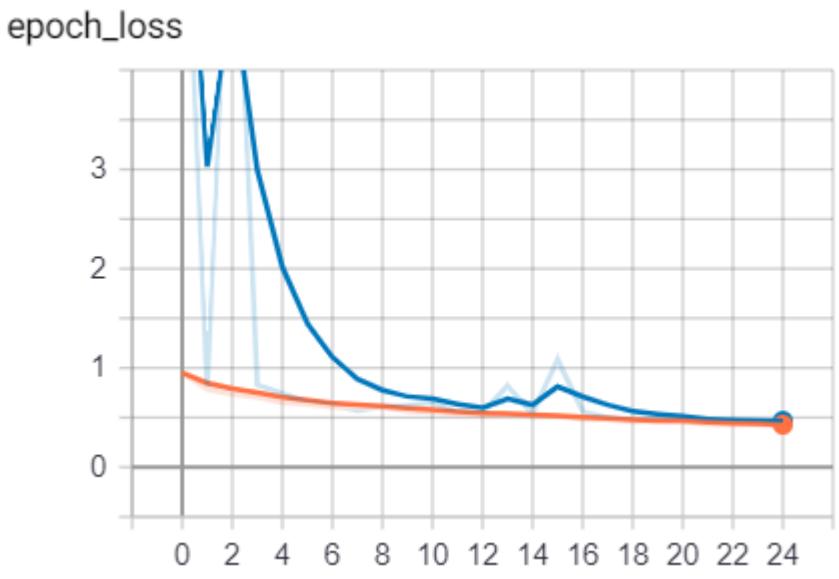
In [0]:
```
%tensorboard --logdir segnet_model1
```

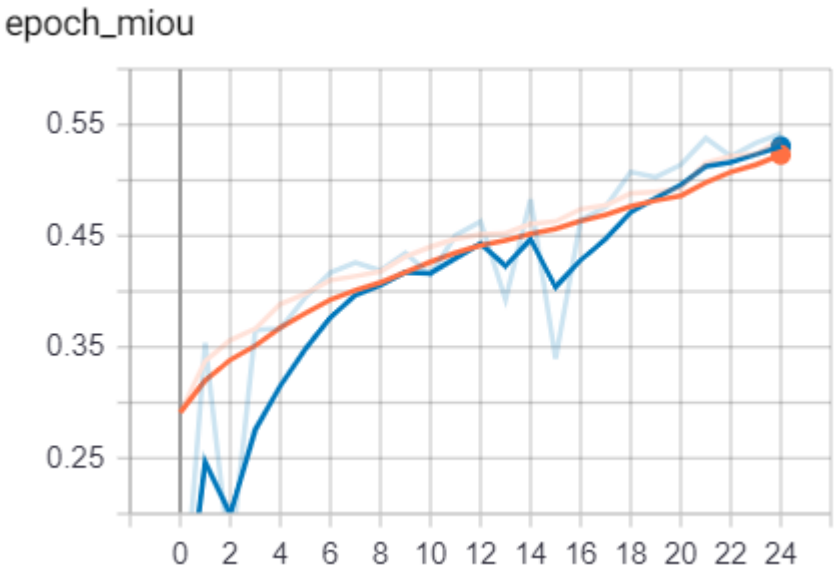
```
<IPython.core.display.Javascript object>
```

In [0]:
```
y_pred = segnet_model1.predict(X_val)
print('MIoU for SegNet model is :',IoU(y_val, y_pred))
```

```
MIoU for SegNet model is : 0.5439742734265953
```

Tensorboard plots





Model 6 : SegNet (with Image augmentation)

In [0]:

```
%load_ext tensorboard
def SegNet2():
    inputs = Input(shape=(256, 320,3))

    # Encoder
    conv1 = Convolution2D(64, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(inputs)
    conv1 = BatchNormalization()(conv1)
    conv1 = Activation("relu")(conv1)
    conv2 = Convolution2D(64, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(conv1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Activation("relu")(conv2)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv2)


    conv3 = Convolution2D(128, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(pool1)
    conv3 = BatchNormalization()(conv3)
    conv3 = Activation("relu")(conv3)
    conv4 = Convolution2D(128, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(conv3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Activation("relu")(conv4)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv4)


    conv5 = Convolution2D(256, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(pool2)
    conv5 = BatchNormalization()(conv5)
    conv5 = Activation("relu")(conv5)
    conv6 = Convolution2D(256, 3, padding="same", kernel_initializer =
                          keras.initializers.glorot_normal(seed=84))(conv5)
    conv6 = BatchNormalization()(conv6)
    conv6 = Activation("relu")(conv6)
    conv7 = Convolution2D(256, 3, padding="same", kernel_initializer = keras.
    conv7 = BatchNormalization()(conv7)
    conv7 = Activation("relu")(conv7)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv7)

    conv8 = Convolution2D(512, 3, padding="same", kernel_initializer = keras.
    conv8 = BatchNormalization()(conv8)
    conv8 = Activation("relu")(conv8)
    conv9 = Convolution2D(512, 3, padding="same", kernel_initializer = keras.
    conv9 = BatchNormalization()(conv9)
    conv9 = Activation("relu")(conv9)
    conv10 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv10 = BatchNormalization()(conv10)
    conv10 = Activation("relu")(conv10)
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv10)

    # Decoder
    up1 = UpSampling2D(size=(2, 2))(pool4)
    conv11 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv11 = BatchNormalization()(conv11)
    conv11 = Activation("relu")(conv11)
```

```
    conv12 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv12 = BatchNormalization()(conv12)
    conv12 = Activation("relu")(conv12)
    conv13 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv13 = BatchNormalization()(conv13)
    conv13 = Activation("relu")(conv13)

    up2 = UpSampling2D(size=(2, 2))(conv13)
    conv14 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv14 = BatchNormalization()(conv14)
    conv14 = Activation("relu")(conv14)
    conv15 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv15 = BatchNormalization()(conv15)
    conv15 = Activation("relu")(conv15)
    conv16 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv16 = BatchNormalization()(conv16)
    conv16 = Activation("relu")(conv16)

    up3 = UpSampling2D(size=(2, 2))(conv16)
    conv17 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv17 = BatchNormalization()(conv17)
    conv17 = Activation("relu")(conv17)
    conv18 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv18 = BatchNormalization()(conv18)
    conv18 = Activation("relu")(conv18)

    up4 = UpSampling2D(size=(2, 2))(conv18)
    conv19 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv19 = BatchNormalization()(conv19)
    conv19 = Activation("relu")(conv19)
    conv20 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv20 = BatchNormalization()(conv20)
    conv20 = Activation("relu")(conv20)

    conv21 = Convolution2D(7, 3, padding="same", kernel_initializer = keras.i
    out = Activation("softmax")(conv21)
    model = Model(inputs,out)
    return model
```

In [0]:
```
segnet_model2 = SegNet2()
segnet_model2.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 256, 320, 64)      1792
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 320, 64)      256
_________________________________________________________________
activation (Activation)      (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 256, 320, 64)      36928
_________________________________________________________________
batch_normalization_1 (Batch (None, 256, 320, 64)      256
_________________________________________________________________
activation_1 (Activation)    (None, 256, 320, 64)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 160, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 160, 128)     73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 128, 160, 128)     512
_________________________________________________________________
activation_2 (Activation)    (None, 128, 160, 128)     0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 128, 160, 128)     147584
_________________________________________________________________
batch_normalization_3 (Batch (None, 128, 160, 128)     512
_________________________________________________________________
activation_3 (Activation)    (None, 128, 160, 128)     0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 80, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 80, 256)       295168
_________________________________________________________________
batch_normalization_4 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_4 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 80, 256)       590080
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_5 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 80, 256)       590080
_________________________________________________________________
batch_normalization_6 (Batch (None, 64, 80, 256)       1024
_________________________________________________________________
activation_6 (Activation)    (None, 64, 80, 256)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 40, 256)       0
```

conv2d_7 (Conv2D)	(None, 32, 40, 512)	1180160
batch_normalization_7 (Batch	(None, 32, 40, 512)	2048
activation_7 (Activation)	(None, 32, 40, 512)	0
conv2d_8 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_8 (Batch	(None, 32, 40, 512)	2048
activation_8 (Activation)	(None, 32, 40, 512)	0
conv2d_9 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_9 (Batch	(None, 32, 40, 512)	2048
activation_9 (Activation)	(None, 32, 40, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 16, 20, 512)	0
up_sampling2d (UpSampling2D)	(None, 32, 40, 512)	0
conv2d_10 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_10 (Batc	(None, 32, 40, 512)	2048
activation_10 (Activation)	(None, 32, 40, 512)	0
conv2d_11 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_11 (Batc	(None, 32, 40, 512)	2048
activation_11 (Activation)	(None, 32, 40, 512)	0
conv2d_12 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_12 (Batc	(None, 32, 40, 512)	2048
activation_12 (Activation)	(None, 32, 40, 512)	0
up_sampling2d_1 (UpSampling2	(None, 64, 80, 512)	0
conv2d_13 (Conv2D)	(None, 64, 80, 256)	1179904
batch_normalization_13 (Batc	(None, 64, 80, 256)	1024
activation_13 (Activation)	(None, 64, 80, 256)	0
conv2d_14 (Conv2D)	(None, 64, 80, 256)	590080
batch_normalization_14 (Batc	(None, 64, 80, 256)	1024
activation_14 (Activation)	(None, 64, 80, 256)	0
conv2d_15 (Conv2D)	(None, 64, 80, 256)	590080

```
batch_normalization_15 (Batc (None, 64, 80, 256)       1024
_________________________________________________________________
activation_15 (Activation)   (None, 64, 80, 256)       0
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 128, 160, 256)     0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 128, 160, 128)     295040
_________________________________________________________________
batch_normalization_16 (Batc (None, 128, 160, 128)     512
_________________________________________________________________
activation_16 (Activation)   (None, 128, 160, 128)     0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 128, 160, 128)     147584
_________________________________________________________________
batch_normalization_17 (Batc (None, 128, 160, 128)     512
_________________________________________________________________
activation_17 (Activation)   (None, 128, 160, 128)     0
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 256, 320, 128)     0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 256, 320, 64)      73792
_________________________________________________________________
batch_normalization_18 (Batc (None, 256, 320, 64)      256
_________________________________________________________________
activation_18 (Activation)   (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 256, 320, 64)      36928
_________________________________________________________________
batch_normalization_19 (Batc (None, 256, 320, 64)      256
_________________________________________________________________
activation_19 (Activation)   (None, 256, 320, 64)      0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 256, 320, 7)       4039
_________________________________________________________________
activation_20 (Activation)   (None, 256, 320, 7)       0
=================================================================
Total params: 17,653,639
Trainable params: 17,642,887
Non-trainable params: 10,752
_________________________________________________________________
```

In [0]:
```
log_dir_6 = os.path.join('segnet_model2')
tensorboard_callback6 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_6)
```

In [0]:
```
segnet_model2.compile(optimizer = Adam(0.0001), loss = 'categorical_crossentr
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator

#Data Augmentation
datagen = ImageDataGenerator(rotation_range=30,width_shift_range=0.15, height
# prepare iterator
trainX_gen = datagen.flow(X_train,batch_size=1035,seed=123)
trainY_gen = datagen.flow(y_train,batch_size=1035,seed=123)
```

```
Using TensorFlow backend.
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/numpy_arra
y_iterator.py:127: UserWarning: NumpyArrayIterator is set to use the data f
ormat convention "channels_last" (channels on axis 3), i.e. expected either
1, 3, or 4 channels on axis 3. However, it was passed an array with shape
(1035, 256, 320, 7) (7 channels).
  str(self.x.shape[channels_axis]) + ' channels).')
```

In [0]:
```
train_generator = zip(trainX_gen, trainY_gen)
```

In [0]:
```
x=X_train.shape[0]//15
x
```

Out[1]:
```
345
```

In [0]:
```
#he_normal   seed=58
history6=segnet_model2.fit(train_generator,steps_per_epoch=x,epochs=20,verbos
                                    validation_data=(X_val, y_val),callbacks=[t
```

```
Epoch 1/20
345/345 [==============================] - 66s 190ms/step - loss: 0.8702
- miou: 0.3289 - val_loss: 3.7222 - val_miou: 0.1276
Epoch 2/20
345/345 [==============================] - 61s 176ms/step - loss: 0.7020
- miou: 0.3797 - val_loss: 0.6817 - val_miou: 0.3891
Epoch 3/20
345/345 [==============================] - 61s 177ms/step - loss: 0.6555
- miou: 0.3972 - val_loss: 0.6186 - val_miou: 0.4019
Epoch 4/20
345/345 [==============================] - 61s 177ms/step - loss: 0.6179
- miou: 0.4123 - val_loss: 0.6371 - val_miou: 0.4513
Epoch 5/20
345/345 [==============================] - 61s 178ms/step - loss: 0.5901
- miou: 0.4250 - val_loss: 0.6792 - val_miou: 0.4534
Epoch 6/20
345/345 [==============================] - 61s 178ms/step - loss: 0.5709
- miou: 0.4399 - val_loss: 0.5932 - val_miou: 0.4528
Epoch 7/20
345/345 [==============================] - 61s 178ms/step - loss: 0.5507
- miou: 0.4521 - val_loss: 0.5909 - val_miou: 0.4526
Epoch 8/20
345/345 [==============================] - 62s 179ms/step - loss: 0.5312
- miou: 0.4646 - val_loss: 0.5573 - val_miou: 0.4648
Epoch 9/20
345/345 [==============================] - 62s 179ms/step - loss: 0.5253
- miou: 0.4672 - val_loss: 0.5235 - val_miou: 0.4799
Epoch 10/20
345/345 [==============================] - 62s 180ms/step - loss: 0.5144
- miou: 0.4776 - val_loss: 0.5735 - val_miou: 0.4626
Epoch 11/20
345/345 [==============================] - 61s 177ms/step - loss: 0.4918
- miou: 0.4890 - val_loss: 0.4922 - val_miou: 0.5070
Epoch 12/20
345/345 [==============================] - 61s 177ms/step - loss: 0.4733
- miou: 0.4985 - val_loss: 0.5324 - val_miou: 0.4615
Epoch 13/20
345/345 [==============================] - 61s 176ms/step - loss: 0.4685
- miou: 0.5078 - val_loss: 0.6411 - val_miou: 0.4585
Epoch 14/20
345/345 [==============================] - 61s 177ms/step - loss: 0.4484
- miou: 0.5190 - val_loss: 0.5229 - val_miou: 0.4892
Epoch 15/20
345/345 [==============================] - 61s 177ms/step - loss: 0.4392
- miou: 0.5263 - val_loss: 0.5294 - val_miou: 0.4822
Epoch 16/20
345/345 [==============================] - 61s 176ms/step - loss: 0.4224
- miou: 0.5371 - val_loss: 0.5077 - val_miou: 0.5076
Epoch 17/20
345/345 [==============================] - 63s 182ms/step - loss: 0.4015
- miou: 0.5486 - val_loss: 0.5412 - val_miou: 0.4934
Epoch 18/20
```

```
345/345 [==============================] - 63s 181ms/step - loss: 0.3940
- miou: 0.5566 - val_loss: 0.4729 - val_miou: 0.5358
Epoch 19/20
345/345 [==============================] - 63s 182ms/step - loss: 0.3853
- miou: 0.5628 - val_loss: 0.5951 - val_miou: 0.4913
Epoch 20/20
345/345 [==============================] - 62s 178ms/step - loss: 0.3694
- miou: 0.5729 - val_loss: 0.4839 - val_miou: 0.5141
```

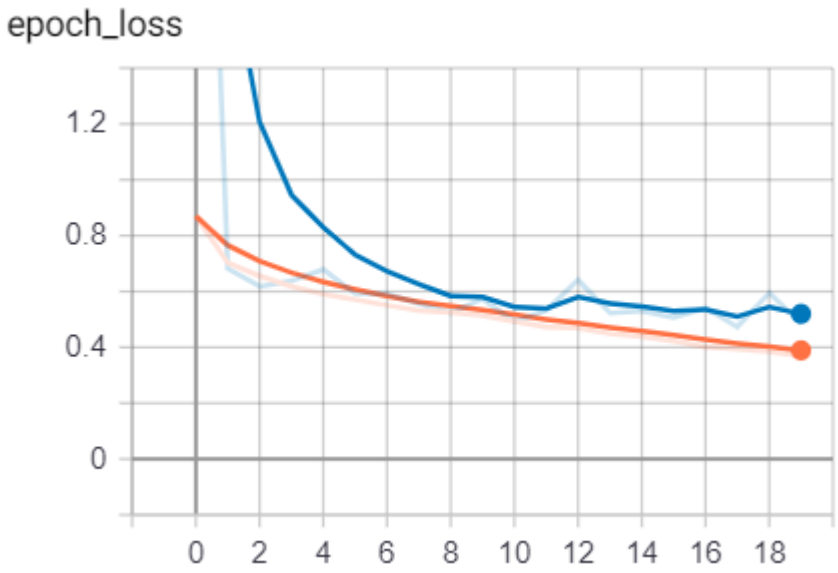
In [0]:
```
%tensorboard --logdir segnet_model2
```

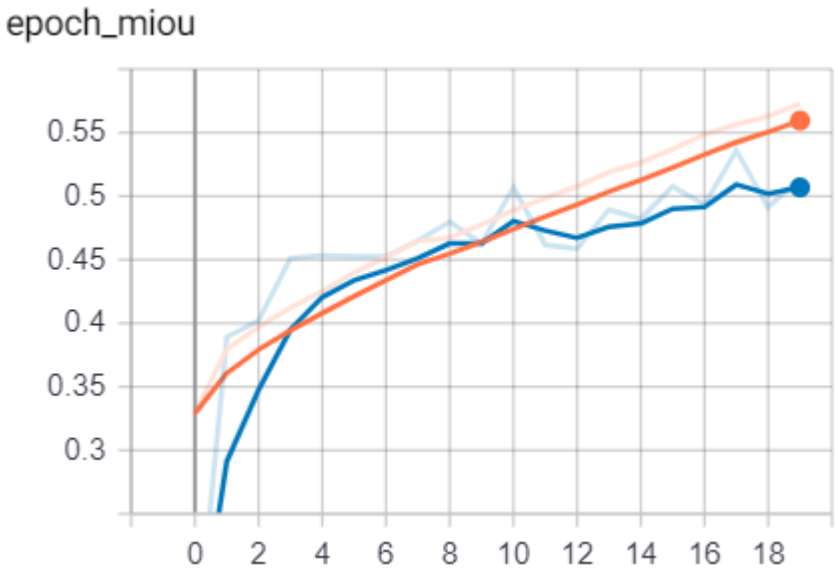
```
<IPython.core.display.Javascript object>
```

In [0]:
```
y_pred = segnet_model2.predict(X_val)
print('MIoU for SegNet model with Image Augmentation is :',IoU(y_val, y_pred))
```

```
MIoU for SegNet model with Image Augmentation is : 0.513960310130104
```

Tensorboard plots





Model 7 : VGG16(encoder)+SegNet(decoder)

In [0]:
```
%load_ext tensorboard
```

In [0]:

```
vgg_encoder = encoder_vgg16.get_layer("block5_conv3").output
def vgg_segnet1():
    # Decoder
    up1 = UpSampling2D(size=(2, 2))(vgg_encoder)
    conv11 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv11 = BatchNormalization()(conv11)
    conv11 = Activation("relu")(conv11)
    conv12 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv12 = BatchNormalization()(conv12)
    conv12 = Activation("relu")(conv12)
    conv13 = Convolution2D(512, 3, padding="same", kernel_initializer = keras
    conv13 = BatchNormalization()(conv13)
    conv13 = Activation("relu")(conv13)

    up2 = UpSampling2D(size=(2, 2))(conv13)
    conv14 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv14 = BatchNormalization()(conv14)
    conv14 = Activation("relu")(conv14)
    conv15 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv15 = BatchNormalization()(conv15)
    conv15 = Activation("relu")(conv15)
    conv16 = Convolution2D(256, 3, padding="same", kernel_initializer = keras
    conv16 = BatchNormalization()(conv16)
    conv16 = Activation("relu")(conv16)

    up3 = UpSampling2D(size=(2, 2))(conv16)
    conv17 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv17 = BatchNormalization()(conv17)
    conv17 = Activation("relu")(conv17)
    conv18 = Convolution2D(128, 3, padding="same", kernel_initializer = keras
    conv18 = BatchNormalization()(conv18)
    conv18 = Activation("relu")(conv18)

    up4 = UpSampling2D(size=(2, 2))(conv18)
    conv19 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv19 = BatchNormalization()(conv19)
    conv19 = Activation("relu")(conv19)
    conv20 = Convolution2D(64, 3, padding="same", kernel_initializer = keras.
    conv20 = BatchNormalization()(conv20)
    conv20 = Activation("relu")(conv20)

    conv21 = Convolution2D(7, 3, padding="same", kernel_initializer = keras.i
    out = Activation("softmax")(conv21)
    model = Model(encoder_vgg16.input,out)
    return model
```

In [0]:
```
vgg_segnet_model1 = vgg_segnet1()
vgg_segnet_model1.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 320, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 320, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 160, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 160, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 160, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 80, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 80, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 80, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 80, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 40, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 40, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 40, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 40, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 20, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 20, 512)       2359808
_________________________________________________________________
up_sampling2d (UpSampling2D) (None, 32, 40, 512)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 32, 40, 512)       2359808
_________________________________________________________________
batch_normalization (BatchNo (None, 32, 40, 512)       2048
_________________________________________________________________
activation (Activation)      (None, 32, 40, 512)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 40, 512)       2359808
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 40, 512)       2048
_________________________________________________________________
activation_1 (Activation)    (None, 32, 40, 512)       0
```

conv2d_2 (Conv2D)	(None, 32, 40, 512)	2359808
batch_normalization_2 (Batch	(None, 32, 40, 512)	2048
activation_2 (Activation)	(None, 32, 40, 512)	0
up_sampling2d_1 (UpSampling2	(None, 64, 80, 512)	0
conv2d_3 (Conv2D)	(None, 64, 80, 256)	1179904
batch_normalization_3 (Batch	(None, 64, 80, 256)	1024
activation_3 (Activation)	(None, 64, 80, 256)	0
conv2d_4 (Conv2D)	(None, 64, 80, 256)	590080
batch_normalization_4 (Batch	(None, 64, 80, 256)	1024
activation_4 (Activation)	(None, 64, 80, 256)	0
conv2d_5 (Conv2D)	(None, 64, 80, 256)	590080
batch_normalization_5 (Batch	(None, 64, 80, 256)	1024
activation_5 (Activation)	(None, 64, 80, 256)	0
up_sampling2d_2 (UpSampling2	(None, 128, 160, 256)	0
conv2d_6 (Conv2D)	(None, 128, 160, 128)	295040
batch_normalization_6 (Batch	(None, 128, 160, 128)	512
activation_6 (Activation)	(None, 128, 160, 128)	0
conv2d_7 (Conv2D)	(None, 128, 160, 128)	147584
batch_normalization_7 (Batch	(None, 128, 160, 128)	512
activation_7 (Activation)	(None, 128, 160, 128)	0
up_sampling2d_3 (UpSampling2	(None, 256, 320, 128)	0
conv2d_8 (Conv2D)	(None, 256, 320, 64)	73792
batch_normalization_8 (Batch	(None, 256, 320, 64)	256
activation_8 (Activation)	(None, 256, 320, 64)	0
conv2d_9 (Conv2D)	(None, 256, 320, 64)	36928
batch_normalization_9 (Batch	(None, 256, 320, 64)	256
activation_9 (Activation)	(None, 256, 320, 64)	0
conv2d_10 (Conv2D)	(None, 256, 320, 7)	4039

```
activation_10 (Activation)   (None, 256, 320, 7)       0
=================================================================
Total params: 24,722,311
Trainable params: 10,002,247
Non-trainable params: 14,720,064
_________________________________________________________________
```

In [0]:
```
log_dir = os.path.join('vgg_segnet_model1')
tensorboard_callback7 = tf.keras.callbacks.TensorBoard(log_dir=log_dir)
```

In [0]:
```
vgg_segnet_model1.compile(optimizer = Adam(0.00001), loss = 'categorical_cros
```

In [0]:
```
x=X_train.shape[0]//3
x
```

Out[24]: 345

In [0]:
```
history = vgg_segnet_model1.fit(X_train, y_train,steps_per_epoch=x,epochs=10,
```

```
Epoch 1/10
345/345 [==============================] - 55s 159ms/step - loss: 1.0824 -
miou: 0.3032 - val_loss: 0.7827 - val_miou: 0.3768
Epoch 2/10
345/345 [==============================] - 48s 139ms/step - loss: 0.7484 -
miou: 0.3790 - val_loss: 0.6701 - val_miou: 0.4078
Epoch 3/10
345/345 [==============================] - 48s 140ms/step - loss: 0.6695 -
miou: 0.3994 - val_loss: 0.6304 - val_miou: 0.4204
Epoch 4/10
345/345 [==============================] - 48s 140ms/step - loss: 0.6194 -
miou: 0.4192 - val_loss: 0.6004 - val_miou: 0.4383
Epoch 5/10
345/345 [==============================] - 49s 141ms/step - loss: 0.5858 -
miou: 0.4370 - val_loss: 0.5867 - val_miou: 0.4536
Epoch 6/10
345/345 [==============================] - 48s 140ms/step - loss: 0.5541 -
miou: 0.4580 - val_loss: 0.5787 - val_miou: 0.4631
Epoch 7/10
345/345 [==============================] - 48s 140ms/step - loss: 0.5285 -
miou: 0.4732 - val_loss: 0.5758 - val_miou: 0.4675
Epoch 8/10
345/345 [==============================] - 49s 143ms/step - loss: 0.5054 -
miou: 0.4862 - val_loss: 0.5724 - val_miou: 0.4734
Epoch 9/10
345/345 [==============================] - 49s 143ms/step - loss: 0.4857 -
miou: 0.5005 - val_loss: 0.5673 - val_miou: 0.4793
Epoch 10/10
345/345 [==============================] - 49s 143ms/step - loss: 0.4650 -
miou: 0.5150 - val_loss: 0.5673 - val_miou: 0.4820
```

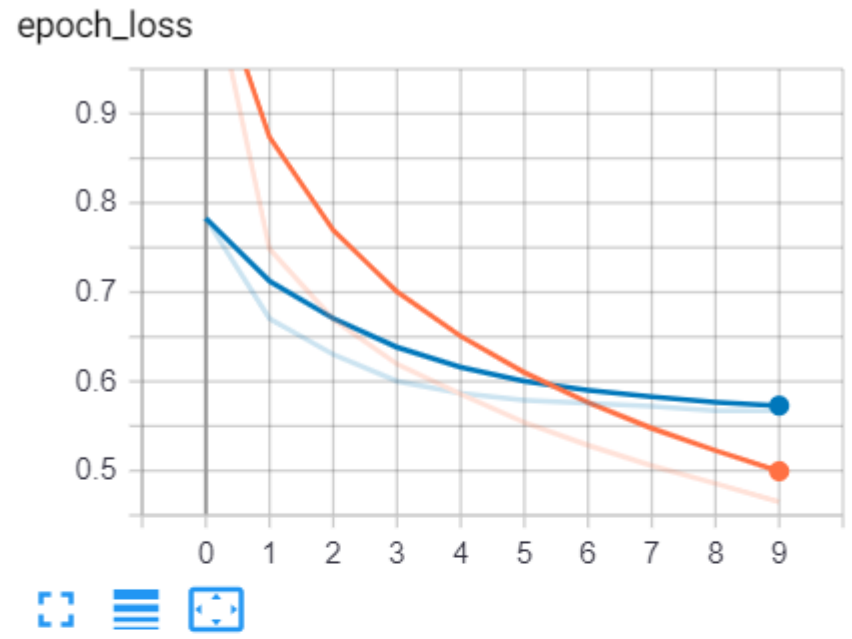
In [0]:
```
%tensorboard --logdir vgg_segnet_model1
```

In [0]:
```
y_pred = vgg_segnet_model1.predict(X_val)
print('MIoU for VGG_SegNet model without Image Augmentation is :',IoU(y_val,
```

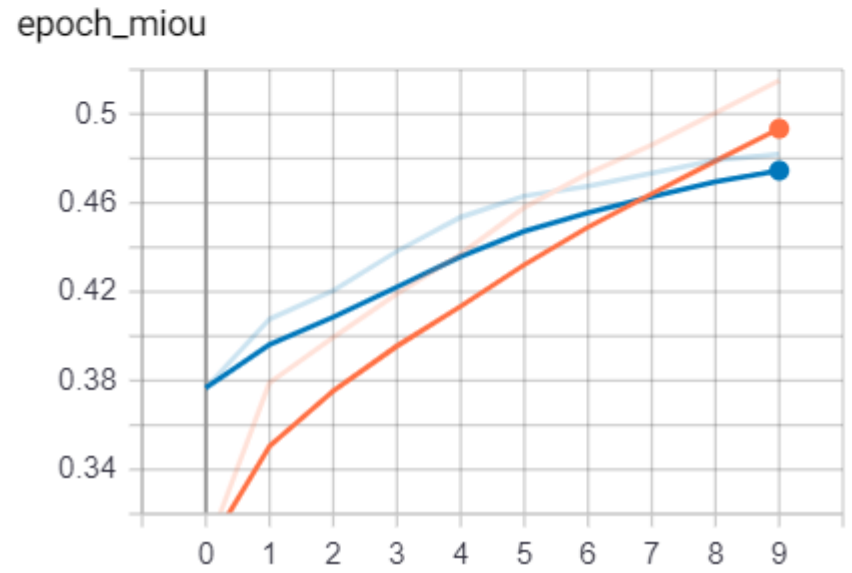
```
MIoU for VGG_SegNet model without Image Augmentation is : 0.482564036499241
8
```

In [0]:
```
vgg_segnet_model1.save_weights('vgg_segnet_model1_weightsfile.h5')
```

Tensorboard plots



epoch_miou



Model 8 : VGG16(encoder)+SegNet(decoder) [with Image Augmentation]

In [0]:
```
vgg_segnet_model2 = vgg_segnet1()
vgg_segnet_model2.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 320, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 320, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 320, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 160, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 160, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 160, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 80, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 80, 256)       295168
```

In [0]:
```
log_dir_2 = os.path.join('vgg_segnet_model2')
tensorboard_callback8 = tf.keras.callbacks.TensorBoard(log_dir=log_dir_2)
```

In [0]:
```
vgg_segnet_model2.compile(optimizer = Adam(0.0001), loss = 'categorical_cross
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator

#Data Augmentation
datagen = ImageDataGenerator(rotation_range=30,width_shift_range=0.15, height
# prepare iterator
trainX_gen = datagen.flow(X_train,seed=123)
trainY_gen = datagen.flow(y_train,seed=123)
```

```
Using TensorFlow backend.
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/numpy_arra
y_iterator.py:127: UserWarning: NumpyArrayIterator is set to use the data f
ormat convention "channels_last" (channels on axis 3), i.e. expected either
1, 3, or 4 channels on axis 3. However, it was passed an array with shape
(1035, 256, 320, 7) (7 channels).
  str(self.x.shape[channels_axis]) + ' channels).')
```

In [0]:
```
train_generator = zip(trainX_gen, trainY_gen)
```

In [0]:
```
x=X_train.shape[0]//15
x
```

Out[27]: 69

In [0]:
```
history2=vgg_segnet_model2.fit(train_generator,steps_per_epoch=x,epochs=15,ve
                               validation_data=(X_val, y_val),callbacks=[t
```

```
Epoch 1/15
69/69 [==============================] - 119s 2s/step - loss: 0.8869 - mio
u: 0.3497 - val_loss: 1.1220 - val_miou: 0.2802
Epoch 2/15
69/69 [==============================] - 117s 2s/step - loss: 0.6475 - mio
u: 0.4142 - val_loss: 0.7913 - val_miou: 0.3653
Epoch 3/15
69/69 [==============================] - 116s 2s/step - loss: 0.5985 - mio
u: 0.4402 - val_loss: 0.6161 - val_miou: 0.4301
Epoch 4/15
69/69 [==============================] - 117s 2s/step - loss: 0.5717 - mio
u: 0.4593 - val_loss: 0.5779 - val_miou: 0.4608
Epoch 5/15
69/69 [==============================] - 116s 2s/step - loss: 0.5554 - mio
u: 0.4732 - val_loss: 0.5621 - val_miou: 0.4562
Epoch 6/15
69/69 [==============================] - 116s 2s/step - loss: 0.5347 - mio
u: 0.4864 - val_loss: 0.5587 - val_miou: 0.4866
Epoch 7/15
69/69 [==============================] - 117s 2s/step - loss: 0.5190 - mio
u: 0.4932 - val_loss: 0.5388 - val_miou: 0.4858
Epoch 8/15
69/69 [==============================] - 116s 2s/step - loss: 0.5098 - mio
u: 0.5051 - val_loss: 0.5570 - val_miou: 0.4880
Epoch 9/15
69/69 [==============================] - 116s 2s/step - loss: 0.4969 - mio
u: 0.5121 - val_loss: 0.5311 - val_miou: 0.4985
Epoch 10/15
69/69 [==============================] - 117s 2s/step - loss: 0.4868 - mio
u: 0.5211 - val_loss: 0.5399 - val_miou: 0.4879
Epoch 11/15
69/69 [==============================] - 114s 2s/step - loss: 0.4733 - mio
u: 0.5291 - val_loss: 0.5582 - val_miou: 0.4861
Epoch 12/15
69/69 [==============================] - 116s 2s/step - loss: 0.4638 - mio
u: 0.5375 - val_loss: 0.5422 - val_miou: 0.5066
Epoch 13/15
69/69 [==============================] - 116s 2s/step - loss: 0.4629 - mio
u: 0.5387 - val_loss: 0.5336 - val_miou: 0.4913
Epoch 14/15
69/69 [==============================] - 116s 2s/step - loss: 0.4517 - mio
u: 0.5499 - val_loss: 0.5419 - val_miou: 0.4994
Epoch 15/15
69/69 [==============================] - 116s 2s/step - loss: 0.4439 - mio
u: 0.5522 - val_loss: 0.5396 - val_miou: 0.5071
```

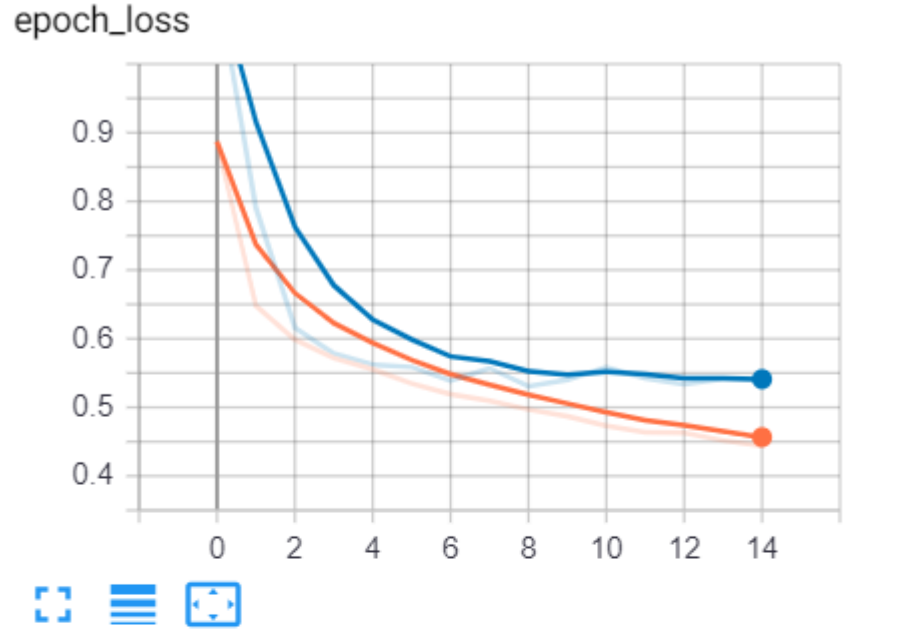
In [0]:
```
%tensorboard --logdir vgg_segnet_model2
```

In [0]:
```
y_pred = vgg_segnet_model2.predict(X_val)
print('MIoU for VGG_SegNet model without Image Augmentation is :',IoU(y_val,
```

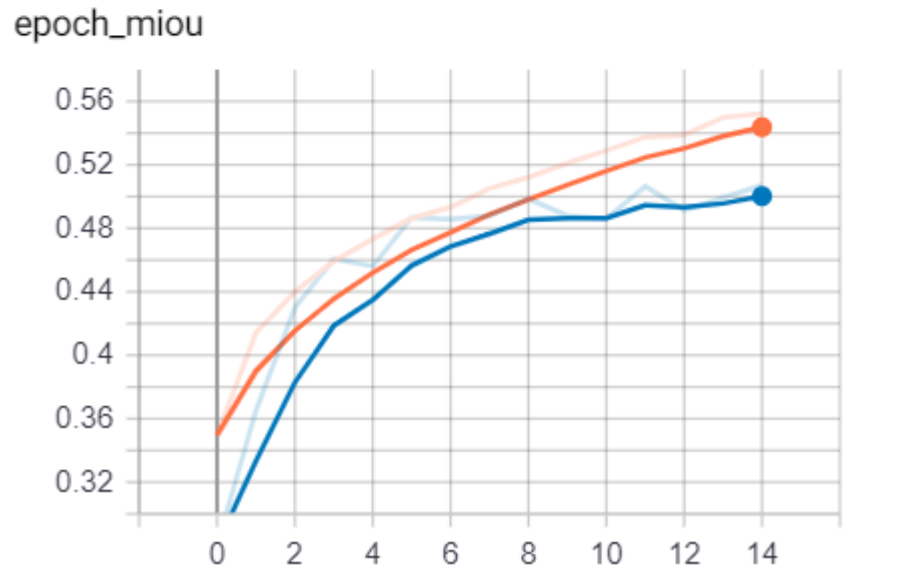
```
MIoU for VGG_SegNet model without Image Augmentation is : 0.509289727147481
6
```

In [0]:
```
vgg_segnet_model2.save_weights('vgg_segnet_model2_weightsfile.h5')
```

Tensorboard plots



epoch_miou



Results :

In [2]:

```
from prettytable import PrettyTable
x = PrettyTable()

x.field_names = ["Model",'Augmentation', "MIoU"]

x.add_row(["UNet", "No",  0.5209])
x.add_row(["UNet", "Yes", 0.4989])
x.add_row(["VGG16(Encoder)_UNet(Decoder)", "No",  0.5533])
x.add_row(["VGG16(Encoder)_UNet(Decoder)", "Yes", 0.5732])
x.add_row(["SegNet", "No", 0.5439])
x.add_row(["SegNet", "Yes", 0.5139])
x.add_row(["VGG16(Encoder)_SegNet(Decoder)", "No",  0.4825])
x.add_row(["VGG16(Encoder)_SegNet(Decoder)", "Yes", 0.5092])

print(x)
```

```
+--------------------------------+--------------+--------+
|             Model              | Augmentation |  MIoU  |
+--------------------------------+--------------+--------+
|              UNet              |      No      | 0.5209 |
|              UNet              |     Yes      | 0.4989 |
|  VGG16(Encoder)_UNet(Decoder)  |      No      | 0.5533 |
|  VGG16(Encoder)_UNet(Decoder)  |     Yes      | 0.5732 |
|             SegNet             |      No      | 0.5439 |
|             SegNet             |     Yes      | 0.5139 |
| VGG16(Encoder)_SegNet(Decoder) |      No      | 0.4825 |
| VGG16(Encoder)_SegNet(Decoder) |     Yes      | 0.5092 |
+--------------------------------+--------------+--------+
```