In [ ]:
```
import tensorflow as tf
import IPython.display as display
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import os
import pathlib
from skimage import io, color
import tensorflow as tf
from tqdm import tqdm
```

In [0]:
```
#Taking the paths for train images
x_tr = tf.data.Dataset.list_files((str(pathlib.Path('idd20k_lite/leftImg8bit/train')/'*/*')),shuffle=False)
```

In [0]:
```
#Taking the paths for train masks
y_tr = tf.data.Dataset.list_files((str(pathlib.Path('idd20k_lite/gtFine/train')/'*/*_inst_label.png')),shuffle=False)
```

In [10]:
```
for f in x_tr.take(5):
    print(f.numpy())
```

```
b'idd20k_lite/leftImg8bit/train/0/024541_image.jpg'
b'idd20k_lite/leftImg8bit/train/0/024703_image.jpg'
b'idd20k_lite/leftImg8bit/train/1/092468_image.jpg'
b'idd20k_lite/leftImg8bit/train/1/340676_image.jpg'
b'idd20k_lite/leftImg8bit/train/1/502201_image.jpg'
```

In [11]:
```
for f in y_tr.take(5):
    print(f.numpy())
```

```
b'idd20k_lite/gtFine/train/0/024541_inst_label.png'
b'idd20k_lite/gtFine/train/0/024703_inst_label.png'
b'idd20k_lite/gtFine/train/1/092468_inst_label.png'
```

```
b'idd20k_lite/gtFine/train/1/340676_inst_label.png'
b'idd20k_lite/gtFine/train/1/502201_inst_label.png'
```

In [0]:

```
#Taking the paths for validation images
x_val = tf.data.Dataset.list_files((str(pathlib.Path('idd20k_lite/leftImg8bit/val')/'*/*')),shuffle=False)
```

In [0]:

```
#Taking the paths for validation masks
y_val = tf.data.Dataset.list_files((str(pathlib.Path('idd20k_lite/gtFine/val')/'*/*_inst_label.png')),shuffle=False)
```

In [14]:

```
for f in x_val.take(5):
    print(f.numpy())
```

```
b'idd20k_lite/leftImg8bit/val/119/903127_image.jpg'
b'idd20k_lite/leftImg8bit/val/132/475092_image.jpg'
b'idd20k_lite/leftImg8bit/val/132/489315_image.jpg'
b'idd20k_lite/leftImg8bit/val/132/874777_image.jpg'
b'idd20k_lite/leftImg8bit/val/147/425716_image.jpg'
```

In [15]:

```
for f in y_val.take(5):
    print(f.numpy())
```

```
b'idd20k_lite/gtFine/val/119/903127_inst_label.png'
b'idd20k_lite/gtFine/val/132/475092_inst_label.png'
b'idd20k_lite/gtFine/val/132/489315_inst_label.png'
b'idd20k_lite/gtFine/val/132/874777_inst_label.png'
b'idd20k_lite/gtFine/val/147/425716_inst_label.png'
```

In [0]:

```
#THis function return processed images
def process_path(file_path):
    img = tf.io.read_file(file_path)
    img = tf.image.decode_jpeg(img, channels=3)
    img = tf.image.convert_image_dtype(img, tf.float32)
    img = tf.image.resize(img, [224, 224])
    return img
```

In [0]:

```
#THis function return processed labels
def segment(filename):
    def a1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,0]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def b1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,1]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def c1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,2]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def d1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,3]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def e1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,4]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def f1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,5]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def g1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,6]]), tf.constant([1],dtype=tf.dtypes.float32))
        return c
    def h1():
        c= tf.tensor_scatter_nd_update(b, tf.constant([[i,j,6]]), tf.constant([0],dtype=tf.dtypes.float32))
        return c
    img = tf.io.read_file(filename)
    img = tf.image.decode_jpeg(img, channels=3)
    img = tf.image.resize(img, [224, 224])
    img = img[:, :, 0]
    img=tf.math.round(img)
    a=img
    b=tf.zeros([224,224,7], dtype=tf.dtypes.float32)
    for i in tqdm(range(0,224)):
        for j in range(0,224):
            b=tf.case([(tf.math.equal(a[i][j],0), a1), (tf.math.equal(a
[i][j],1), b1),
                       (tf.math.equal(a[i][j],2), c1),(tf.math.equal(a[
i][j],3), d1),
                      (tf.math.equal(a[i][j],4), e1), (tf.math.equal(a[
i][j],5), f1),
                      (tf.math.equal(a[i][j],6), g1)],default=h1,exclus
ive=True)
    return b
```

In [0]:
```
#Mapping the function to transform all the train and validation images
x_tr_1=x_tr.map(process_path)
x_val_1=x_val.map(process_path)
```

In [0]:
```
#Mapping the function to transform all the train and validation masks
y_tr_1=y_tr.map(segment)
y_val_1=y_val.map(segment)
```

In [0]:
```
#Preparing train and validation datasets using zipping
train = tf.data.Dataset.zip((x_tr_1, y_tr_1))
val = tf.data.Dataset.zip((x_val_1, y_val_1))
```

In [0]:
```
#Preparing dataset to feed to model by shuffling and batching
train_dataset = train.cache().shuffle(500).batch(32).repeat()
train_dataset = train_dataset.prefetch(buffer_size=tf.data.experimental
.AUTOTUNE)
test_dataset = val.batch(32)
```

In [30]:
```
train_dataset
```

Out[30]:
```
<PrefetchDataset shapes: ((None, 224, 224, 3), (None, 224, 224, 7)), ty
pes: (tf.float32, tf.float64)>
```

In [31]:
```
test_dataset
```

Out[31]:
```
<BatchDataset shapes: ((None, 224, 224, 3), (None, 224, 224, 7)), type
s: (tf.float32, tf.float64)>
```

In [0]:
```
IMAGE_ORDERING = 'channels_last'
input_height, input_width = 224, 224
output_height, output_width = 224, 224
n_classes=7
```

In [0]:
```
#Creating model with keras subclass api
class VGGNetModel(tf.keras.Model):
        def __init__(self, classes,chanDim=-1):
                super(VGGNetModel, self).__init__(name='my_model')
                self.conv1A = layers.Conv2D(64, (3, 3), activation = 'r
elu', padding = 'same', name = 'block1_conv1', data_format = IMAGE_ORDE
RING,input_shape=(224,224,3))
                self.act1A = layers.Conv2D(64, (3, 3), activation = 're
lu', padding = 'same', name = 'block1_conv2', data_format = IMAGE_ORDER
ING )
                self.pool1 = layers.MaxPooling2D((2, 2), strides = (2, 
2), name = 'block1_pool', data_format = IMAGE_ORDERING )
                self.conv2A = layers.Conv2D(128, (3, 3), activation = 
'relu', padding = 'same', name = 'block2_conv1', data_format = IMAGE_OR
DERING )
                self.act2A = layers.Conv2D(128, (3, 3), activation = 'r
elu', padding = 'same', name = 'block2_conv2', data_format = IMAGE_ORDE
RING )
                self.bn2A = layers.MaxPooling2D((2, 2), strides = (2, 2
), name = 'block2_pool', data_format = IMAGE_ORDERING )
                self.conv2B = layers.Conv2D(256, (3, 3), activation = 
'relu', padding = 'same', name = 'block3_conv1', data_format = IMAGE_OR
DERING )
                self.act2B =layers.Conv2D(256, (3, 3), activation = 're
lu', padding = 'same', name = 'block3_conv2', data_format = IMAGE_ORDER
ING )
                self.bn2B = layers.Conv2D(256, (3, 3), activation = 're
lu', padding = 'same', name = 'block3_conv3', data_format = IMAGE_ORDER
ING )
```

```
                self.pool2 = layers.MaxPooling2D((2, 2), strides = (2, 2), name = 'block3_pool', data_format = IMAGE_ORDERING )
                self.flatten = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block4_conv1', data_format = IMAGE_ORDERING )
                self.dense3 = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block4_conv2', data_format = IMAGE_ORDERING )
                self.act3 = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block4_conv3', data_format = IMAGE_ORDERING )
                self.bn3 = layers.MaxPooling2D((2, 2), strides = (2, 2), name = 'block4_pool', data_format = IMAGE_ORDERING )
                self.do3 = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block5_conv1', data_format = IMAGE_ORDERING )
                self.dense4 = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block5_conv2', data_format = IMAGE_ORDERING )
                self.softmax = layers.Conv2D(512, (3, 3), activation = 'relu', padding = 'same', name = 'block5_conv3', data_format = IMAGE_ORDERING )
                self.test1=layers.MaxPooling2D((2, 2), strides = (2, 2), name = 'block5_pool', data_format = IMAGE_ORDERING )

                self.test2=layers.Conv2D(4096, (7, 7), activation = 'relu', padding = 'same', name = 'conv6', data_format = IMAGE_ORDERING)
                self.test3=layers.Conv2D(4096, (1, 1), activation = 'relu', padding = 'same', name = 'conv7', data_format = IMAGE_ORDERING)
                self.test4=layers.Conv2DTranspose(nClasses, kernel_size = (4, 4) ,strides = (4, 4), use_bias = False, data_format = IMAGE_ORDERING )
                self.test5=layers.Conv2D(nClasses, (1, 1), activation = 'relu', padding = 'same', name = 'pool4_11', data_format = IMAGE_ORDERING)
                self.test6=layers.Conv2DTranspose(nClasses, kernel_size = (2, 2), strides = (2, 2), use_bias = False, data_format = IMAGE_ORDERING )
                self.test7=layers.Conv2D(nClasses, (1, 1), activation =
 'relu', padding = 'same', name = 'pool3_11', data_format = IMAGE_ORDER
ING)
                self.test8=layers.Add(name = 'add')
                self.test9=layers.Conv2DTranspose(nClasses, kernel_size
 = (8, 8), strides = (8, 8), use_bias = False, data_format = IMAGE_ORDE
RING )
                self.test10=layers.Activation('softmax')


        def call(self, inputs):
                inp=Input(shape = (input_height, input_width, 3))
                x = self.conv1A(inputs)
                x = self.act1A(x)
                x = self.pool1(x)
                f1 = x
                x = self.conv2A(x)
                x = self.act2A(x)
                x = self.bn2A(x)
                f2 = x
                x = self.conv2B(x)
                x = self.act2B(x)
                x = self.bn2B(x)
                x = self.pool2(x)
                pool3 = x
                x = self.flatten(x)
                x = self.dense3(x)
                x = self.act3(x)
                x = self.bn3(x)
                pool4 = x
                x = self.do3(x)
                x = self.dense4(x)
                x = self.softmax(x)
                x=self.test1(x)
                pool5 = x

                o = self.test2(pool5)
                conv7 = (self.test3)(o)
                conv7_4 = (self.test4)(conv7)
                pool411 = (self.test5)(pool4)
                pool411_2 = (self.test6)(pool411)
                pool311 = (self.test7)(pool3)
                o = self.test8([pool411_2, pool311, conv7_4])
                o = self.test9(o)
                o = (self.test10)(o)

                return o
```

In [0]:
```
model=VGGNetModel(classes=7)
```

In [0]:
```
#building model with input shape
model.build((32,224,224,3))
```

In [36]:
```
model.summary()
```

```
Model: "my_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
block1_conv1 (Conv2D)        multiple                  1792
_________________________________________________________________
block1_conv2 (Conv2D)        multiple                  36928
_________________________________________________________________
block1_pool (MaxPooling2D)   multiple                  0
_________________________________________________________________
block2_conv1 (Conv2D)        multiple                  73856
_________________________________________________________________
block2_conv2 (Conv2D)        multiple                  147584
_________________________________________________________________
block2_pool (MaxPooling2D)   multiple                  0
_________________________________________________________________
block3_conv1 (Conv2D)        multiple                  295168
_________________________________________________________________
block3_conv2 (Conv2D)        multiple                  590080
_________________________________________________________________
block3_conv3 (Conv2D)        multiple                  590080
_________________________________________________________________
block3_pool (MaxPooling2D)   multiple                  0
```

```
_________________________________________________________________
block4_conv1 (Conv2D)        multiple                  1180160
_________________________________________________________________
block4_conv2 (Conv2D)        multiple                  2359808
_________________________________________________________________
block4_conv3 (Conv2D)        multiple                  2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   multiple                  0
_________________________________________________________________
block5_conv1 (Conv2D)        multiple                  2359808
_________________________________________________________________
block5_conv2 (Conv2D)        multiple                  2359808
_________________________________________________________________
block5_conv3 (Conv2D)        multiple                  2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   multiple                  0
_________________________________________________________________
conv6 (Conv2D)               multiple                  102764544
_________________________________________________________________
conv7 (Conv2D)               multiple                  16781312
_________________________________________________________________
conv2d_transpose (Conv2DTran multiple                  458752
_________________________________________________________________
pool4_11 (Conv2D)            multiple                  3591
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr multiple                  196
_________________________________________________________________
pool3_11 (Conv2D)            multiple                  1799
_________________________________________________________________
add (Add)                    multiple                  0
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr multiple                  3136
_________________________________________________________________
activation (Activation)      multiple                  0
=================================================================
Total params: 134,728,018
Trainable params: 134,728,018
Non-trainable params: 0
_________________________________________________________________
```

In [0]:

```
from keras.callbacks import ModelCheckpoint

CHECKPOINT_FILE_PATH = "model_main.hdf5"
checkpoint = ModelCheckpoint(CHECKPOINT_FILE_PATH, monitor='val_acc', v
erbose=1, save_best_only=True, mode='max', period=1)
```

In [0]:

```
sgd = optimizers.SGD(lr = 0.01, decay = 5 ** (-4), momentum = 0.9, nest
erov = True)
model.compile(loss = 'categorical_crossentropy',
              optimizer = 'adam',
              metrics = [tf.keras.metrics.MeanIoU(num_classes=7)])
```

In [38]:

```
hist2 = model.fit(train_dataset,
                  validation_data = test_dataset,epochs = 300, verbose
= 1,steps_per_epoch=43,validation_steps=6,
                 callbacks=[checkpoint])
```

```
Epoch 1/300
43/43 [==============================] - 26s 603ms/step - loss: 1.9521
- mean_io_u: 0.4286 - val_loss: 1.9172 - val_mean_io_u: 0.4286
Epoch 2/300
43/43 [==============================] - 24s 555ms/step - loss: 1.6359
- mean_io_u: 0.4286 - val_loss: 1.2350 - val_mean_io_u: 0.4286
Epoch 3/300
43/43 [==============================] - 24s 550ms/step - loss: 1.1239
- mean_io_u: 0.4286 - val_loss: 1.0791 - val_mean_io_u: 0.4286
Epoch 4/300
43/43 [==============================] - 24s 557ms/step - loss: 1.0223
- mean_io_u: 0.4286 - val_loss: 1.0215 - val_mean_io_u: 0.4286
Epoch 5/300
43/43 [==============================] - 24s 553ms/step - loss: 0.8964
- mean_io_u: 0.4286 - val_loss: 0.8803 - val_mean_io_u: 0.4286
Epoch 6/300
43/43 [==============================] - 24s 552ms/step - loss: 0.8103
- mean_io_u: 0.4286 - val_loss: 0.8139 - val_mean_io_u: 0.4286
Epoch 7/300
43/43 [==============================] - 24s 560ms/step - loss: 0.7525
- mean_io_u: 0.4286 - val_loss: 0.7486 - val_mean_io_u: 0.4286
```

```
Epoch 8/300
43/43 [==============================] - 24s 554ms/step - loss: 0.6914
- mean_io_u: 0.4286 - val_loss: 0.7106 - val_mean_io_u: 0.4286
Epoch 9/300
43/43 [==============================] - 24s 549ms/step - loss: 0.6680
- mean_io_u: 0.4286 - val_loss: 0.7065 - val_mean_io_u: 0.4286
Epoch 10/300
43/43 [==============================] - 24s 563ms/step - loss: 0.6395
- mean_io_u: 0.4286 - val_loss: 0.6839 - val_mean_io_u: 0.4286
Epoch 11/300
43/43 [==============================] - 24s 548ms/step - loss: 0.6120
- mean_io_u: 0.4287 - val_loss: 0.6788 - val_mean_io_u: 0.4286
Epoch 12/300
43/43 [==============================] - 24s 556ms/step - loss: 0.5763
- mean_io_u: 0.4289 - val_loss: 0.6986 - val_mean_io_u: 0.4292
Epoch 13/300
43/43 [==============================] - 24s 561ms/step - loss: 0.5357
- mean_io_u: 0.4295 - val_loss: 0.6950 - val_mean_io_u: 0.4301
Epoch 14/300
43/43 [==============================] - 24s 559ms/step - loss: 0.4793
- mean_io_u: 0.4337 - val_loss: 0.7035 - val_mean_io_u: 0.4347
Epoch 15/300
43/43 [==============================] - 24s 562ms/step - loss: 0.4228
- mean_io_u: 0.4403 - val_loss: 0.7110 - val_mean_io_u: 0.4396
Epoch 16/300
43/43 [==============================] - 24s 559ms/step - loss: 0.3747
- mean_io_u: 0.4468 - val_loss: 0.7404 - val_mean_io_u: 0.4533
Epoch 17/300
43/43 [==============================] - 24s 556ms/step - loss: 0.3384
- mean_io_u: 0.4553 - val_loss: 0.7564 - val_mean_io_u: 0.4523
Epoch 18/300
43/43 [==============================] - 24s 557ms/step - loss: 0.3076
- mean_io_u: 0.4619 - val_loss: 0.7979 - val_mean_io_u: 0.4621
Epoch 19/300
43/43 [==============================] - 24s 555ms/step - loss: 0.2831
- mean_io_u: 0.4684 - val_loss: 0.8017 - val_mean_io_u: 0.4659
Epoch 20/300
43/43 [==============================] - 24s 557ms/step - loss: 0.2549
- mean_io_u: 0.4761 - val_loss: 0.8263 - val_mean_io_u: 0.4735
```

```
Epoch 21/300
43/43 [==============================] - 24s 561ms/step - loss: 0.2341
- mean_io_u: 0.4829 - val_loss: 0.8822 - val_mean_io_u: 0.4792
Epoch 22/300
43/43 [==============================] - 24s 555ms/step - loss: 0.2205
- mean_io_u: 0.4896 - val_loss: 0.8858 - val_mean_io_u: 0.4839
Epoch 23/300
43/43 [==============================] - 24s 560ms/step - loss: 0.2074
- mean_io_u: 0.4942 - val_loss: 0.9242 - val_mean_io_u: 0.4938
Epoch 24/300
43/43 [==============================] - 24s 552ms/step - loss: 0.1943
- mean_io_u: 0.5027 - val_loss: 0.9292 - val_mean_io_u: 0.4952
Epoch 25/300
43/43 [==============================] - 24s 554ms/step - loss: 0.1857
- mean_io_u: 0.5058 - val_loss: 0.9691 - val_mean_io_u: 0.5030
Epoch 26/300
43/43 [==============================] - 24s 564ms/step - loss: 0.1770
- mean_io_u: 0.5124 - val_loss: 0.9761 - val_mean_io_u: 0.5034
Epoch 27/300
43/43 [==============================] - 24s 555ms/step - loss: 0.1659
- mean_io_u: 0.5217 - val_loss: 1.0279 - val_mean_io_u: 0.5071
Epoch 28/300
43/43 [==============================] - 24s 549ms/step - loss: 0.1623
- mean_io_u: 0.5233 - val_loss: 1.0324 - val_mean_io_u: 0.5139
Epoch 29/300
43/43 [==============================] - 24s 564ms/step - loss: 0.1542
- mean_io_u: 0.5300 - val_loss: 1.0453 - val_mean_io_u: 0.5215
Epoch 30/300
43/43 [==============================] - 24s 559ms/step - loss: 0.1469
- mean_io_u: 0.5362 - val_loss: 1.1009 - val_mean_io_u: 0.5308
Epoch 31/300
43/43 [==============================] - 24s 558ms/step - loss: 0.1412
- mean_io_u: 0.5418 - val_loss: 1.1367 - val_mean_io_u: 0.5270
Epoch 32/300
43/43 [==============================] - 24s 556ms/step - loss: 0.1373
- mean_io_u: 0.5456 - val_loss: 1.1451 - val_mean_io_u: 0.5300
Epoch 33/300
43/43 [==============================] - 24s 560ms/step - loss: 0.1324
- mean_io_u: 0.5483 - val_loss: 1.1384 - val_mean_io_u: 0.5257
```

```
Epoch 34/300
43/43 [==============================] - 24s 554ms/step - loss: 0.1278
- mean_io_u: 0.5523 - val_loss: 1.1998 - val_mean_io_u: 0.5349
Epoch 35/300
43/43 [==============================] - 24s 555ms/step - loss: 0.1278
- mean_io_u: 0.5513 - val_loss: 1.1595 - val_mean_io_u: 0.5321
Epoch 36/300
43/43 [==============================] - 24s 556ms/step - loss: 0.1244
- mean_io_u: 0.5494 - val_loss: 1.2266 - val_mean_io_u: 0.5368
Epoch 37/300
43/43 [==============================] - 24s 553ms/step - loss: 0.1191
- mean_io_u: 0.5580 - val_loss: 1.2597 - val_mean_io_u: 0.5407
Epoch 38/300
43/43 [==============================] - 24s 554ms/step - loss: 0.1152
- mean_io_u: 0.5666 - val_loss: 1.2556 - val_mean_io_u: 0.5455
Epoch 39/300
43/43 [==============================] - 24s 559ms/step - loss: 0.1176
- mean_io_u: 0.5604 - val_loss: 1.2634 - val_mean_io_u: 0.5367
Epoch 40/300
43/43 [==============================] - 24s 555ms/step - loss: 0.1142
- mean_io_u: 0.5620 - val_loss: 1.2969 - val_mean_io_u: 0.5415
Epoch 41/300
43/43 [==============================] - 24s 550ms/step - loss: 0.1102
- mean_io_u: 0.5713 - val_loss: 1.3003 - val_mean_io_u: 0.5374
Epoch 42/300
43/43 [==============================] - 24s 561ms/step - loss: 0.1086
- mean_io_u: 0.5734 - val_loss: 1.3217 - val_mean_io_u: 0.5519
Epoch 43/300
43/43 [==============================] - 24s 551ms/step - loss: 0.1061
- mean_io_u: 0.5819 - val_loss: 1.3368 - val_mean_io_u: 0.5523
Epoch 44/300
43/43 [==============================] - 24s 550ms/step - loss: 0.1052
- mean_io_u: 0.5801 - val_loss: 1.3814 - val_mean_io_u: 0.5582
Epoch 45/300
43/43 [==============================] - 24s 562ms/step - loss: 0.1060
- mean_io_u: 0.5789 - val_loss: 1.3514 - val_mean_io_u: 0.5487
Epoch 46/300
43/43 [==============================] - 24s 563ms/step - loss: 0.1035
- mean_io_u: 0.5839 - val_loss: 1.3587 - val_mean_io_u: 0.5559
```

```
Epoch 47/300
43/43 [==============================] - 24s 558ms/step - loss: 0.1020
- mean_io_u: 0.5875 - val_loss: 1.4301 - val_mean_io_u: 0.5574
Epoch 48/300
43/43 [==============================] - 24s 555ms/step - loss: 0.1009
- mean_io_u: 0.5905 - val_loss: 1.3808 - val_mean_io_u: 0.5558
Epoch 49/300
43/43 [==============================] - 24s 558ms/step - loss: 0.1006
- mean_io_u: 0.5905 - val_loss: 1.3938 - val_mean_io_u: 0.5581
Epoch 50/300
43/43 [==============================] - 24s 557ms/step - loss: 0.1000
- mean_io_u: 0.5897 - val_loss: 1.4091 - val_mean_io_u: 0.5544
Epoch 51/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0997
- mean_io_u: 0.5907 - val_loss: 1.3920 - val_mean_io_u: 0.5551
Epoch 52/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0986
- mean_io_u: 0.5920 - val_loss: 1.4241 - val_mean_io_u: 0.5654
Epoch 53/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0973
- mean_io_u: 0.5969 - val_loss: 1.4216 - val_mean_io_u: 0.5629
Epoch 54/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0972
- mean_io_u: 0.5972 - val_loss: 1.4417 - val_mean_io_u: 0.5643
Epoch 55/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0965
- mean_io_u: 0.5984 - val_loss: 1.4249 - val_mean_io_u: 0.5601
Epoch 56/300
43/43 [==============================] - 24s 549ms/step - loss: 0.0954
- mean_io_u: 0.6022 - val_loss: 1.4482 - val_mean_io_u: 0.5667
Epoch 57/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0961
- mean_io_u: 0.5986 - val_loss: 1.4306 - val_mean_io_u: 0.5567
Epoch 58/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0980
- mean_io_u: 0.5922 - val_loss: 1.4203 - val_mean_io_u: 0.5556
Epoch 59/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0965
- mean_io_u: 0.5902 - val_loss: 1.4081 - val_mean_io_u: 0.5558
```

```
Epoch 60/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0939
- mean_io_u: 0.5986 - val_loss: 1.4601 - val_mean_io_u: 0.5610
Epoch 61/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0934
- mean_io_u: 0.6034 - val_loss: 1.4650 - val_mean_io_u: 0.5616
Epoch 62/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0922
- mean_io_u: 0.6081 - val_loss: 1.4609 - val_mean_io_u: 0.5610
Epoch 63/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0964
- mean_io_u: 0.5917 - val_loss: 1.4252 - val_mean_io_u: 0.5585
Epoch 64/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0933
- mean_io_u: 0.5991 - val_loss: 1.4765 - val_mean_io_u: 0.5685
Epoch 65/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0908
- mean_io_u: 0.6105 - val_loss: 1.4928 - val_mean_io_u: 0.5693
Epoch 66/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0899
- mean_io_u: 0.6157 - val_loss: 1.4846 - val_mean_io_u: 0.5697
Epoch 67/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0916
- mean_io_u: 0.6104 - val_loss: 1.4891 - val_mean_io_u: 0.5659
Epoch 68/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0926
- mean_io_u: 0.6044 - val_loss: 1.4219 - val_mean_io_u: 0.5521
Epoch 69/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0911
- mean_io_u: 0.6060 - val_loss: 1.4752 - val_mean_io_u: 0.5686
Epoch 70/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0899
- mean_io_u: 0.6093 - val_loss: 1.4941 - val_mean_io_u: 0.5660
Epoch 71/300
43/43 [==============================] - 24s 569ms/step - loss: 0.0903
- mean_io_u: 0.6096 - val_loss: 1.4884 - val_mean_io_u: 0.5649
Epoch 72/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0902
- mean_io_u: 0.6120 - val_loss: 1.4583 - val_mean_io_u: 0.5642
```

```
Epoch 73/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0896
- mean_io_u: 0.6102 - val_loss: 1.4747 - val_mean_io_u: 0.5629
Epoch 74/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0898
- mean_io_u: 0.6146 - val_loss: 1.4616 - val_mean_io_u: 0.5624
Epoch 75/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0873
- mean_io_u: 0.6191 - val_loss: 1.5262 - val_mean_io_u: 0.5759
Epoch 76/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0882
- mean_io_u: 0.6180 - val_loss: 1.5554 - val_mean_io_u: 0.5767
Epoch 77/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0879
- mean_io_u: 0.6218 - val_loss: 1.4982 - val_mean_io_u: 0.5764
Epoch 78/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0899
- mean_io_u: 0.6083 - val_loss: 1.4678 - val_mean_io_u: 0.5672
Epoch 79/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0896
- mean_io_u: 0.6060 - val_loss: 1.4769 - val_mean_io_u: 0.5640
Epoch 80/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0883
- mean_io_u: 0.6107 - val_loss: 1.4753 - val_mean_io_u: 0.5632
Epoch 81/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0863
- mean_io_u: 0.6203 - val_loss: 1.5208 - val_mean_io_u: 0.5728
Epoch 82/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0863
- mean_io_u: 0.6233 - val_loss: 1.5087 - val_mean_io_u: 0.5726
Epoch 83/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0868
- mean_io_u: 0.6215 - val_loss: 1.5242 - val_mean_io_u: 0.5755
Epoch 84/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0869
- mean_io_u: 0.6212 - val_loss: 1.5322 - val_mean_io_u: 0.5709
Epoch 85/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0859
- mean_io_u: 0.6241 - val_loss: 1.5224 - val_mean_io_u: 0.5773
```

```
Epoch 86/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0853
- mean_io_u: 0.6251 - val_loss: 1.4912 - val_mean_io_u: 0.5703
Epoch 87/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0885
- mean_io_u: 0.6114 - val_loss: 1.4815 - val_mean_io_u: 0.5620
Epoch 88/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0881
- mean_io_u: 0.6083 - val_loss: 1.4294 - val_mean_io_u: 0.5608
Epoch 89/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0882
- mean_io_u: 0.6072 - val_loss: 1.4690 - val_mean_io_u: 0.5580
Epoch 90/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0847
- mean_io_u: 0.6220 - val_loss: 1.5135 - val_mean_io_u: 0.5716
Epoch 91/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0838
- mean_io_u: 0.6334 - val_loss: 1.5418 - val_mean_io_u: 0.5780
Epoch 92/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0838
- mean_io_u: 0.6359 - val_loss: 1.5724 - val_mean_io_u: 0.5780
Epoch 93/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0844
- mean_io_u: 0.6315 - val_loss: 1.5554 - val_mean_io_u: 0.5791
Epoch 94/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0839
- mean_io_u: 0.6348 - val_loss: 1.5567 - val_mean_io_u: 0.5786
Epoch 95/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0839
- mean_io_u: 0.6326 - val_loss: 1.5498 - val_mean_io_u: 0.5752
Epoch 96/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0846
- mean_io_u: 0.6308 - val_loss: 1.5081 - val_mean_io_u: 0.5709
Epoch 97/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0846
- mean_io_u: 0.6282 - val_loss: 1.5159 - val_mean_io_u: 0.5731
Epoch 98/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0830
- mean_io_u: 0.6331 - val_loss: 1.5677 - val_mean_io_u: 0.5822
```

```
Epoch 99/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0833
- mean_io_u: 0.6390 - val_loss: 1.5107 - val_mean_io_u: 0.5766
Epoch 100/300
43/43 [==============================] - 24s 566ms/step - loss: 0.0839
- mean_io_u: 0.6344 - val_loss: 1.5549 - val_mean_io_u: 0.5813
Epoch 101/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0840
- mean_io_u: 0.6296 - val_loss: 1.5448 - val_mean_io_u: 0.5814
Epoch 102/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0856
- mean_io_u: 0.6218 - val_loss: 1.5331 - val_mean_io_u: 0.5679
Epoch 103/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0845
- mean_io_u: 0.6259 - val_loss: 1.4987 - val_mean_io_u: 0.5655
Epoch 104/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0839
- mean_io_u: 0.6258 - val_loss: 1.5328 - val_mean_io_u: 0.5801
Epoch 105/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0826
- mean_io_u: 0.6348 - val_loss: 1.5913 - val_mean_io_u: 0.5831
Epoch 106/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0819
- mean_io_u: 0.6434 - val_loss: 1.5372 - val_mean_io_u: 0.5827
Epoch 107/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0830
- mean_io_u: 0.6364 - val_loss: 1.5349 - val_mean_io_u: 0.5710
Epoch 108/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0816
- mean_io_u: 0.6402 - val_loss: 1.6100 - val_mean_io_u: 0.5872
Epoch 109/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0816
- mean_io_u: 0.6456 - val_loss: 1.5593 - val_mean_io_u: 0.5843
Epoch 110/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0826
- mean_io_u: 0.6403 - val_loss: 1.5229 - val_mean_io_u: 0.5656
Epoch 111/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0831
- mean_io_u: 0.6316 - val_loss: 1.5470 - val_mean_io_u: 0.5819
```

```
Epoch 112/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0829
- mean_io_u: 0.6352 - val_loss: 1.5250 - val_mean_io_u: 0.5797
Epoch 113/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0829
- mean_io_u: 0.6344 - val_loss: 1.5139 - val_mean_io_u: 0.5738
Epoch 114/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0824
- mean_io_u: 0.6349 - val_loss: 1.5760 - val_mean_io_u: 0.5775
Epoch 115/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0812
- mean_io_u: 0.6421 - val_loss: 1.5565 - val_mean_io_u: 0.5735
Epoch 116/300
43/43 [==============================] - 24s 564ms/step - loss: 0.0814
- mean_io_u: 0.6448 - val_loss: 1.5490 - val_mean_io_u: 0.5812
Epoch 117/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0823
- mean_io_u: 0.6373 - val_loss: 1.5623 - val_mean_io_u: 0.5799
Epoch 118/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0824
- mean_io_u: 0.6363 - val_loss: 1.5363 - val_mean_io_u: 0.5769
Epoch 119/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0826
- mean_io_u: 0.6346 - val_loss: 1.5243 - val_mean_io_u: 0.5775
Epoch 120/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0833
- mean_io_u: 0.6314 - val_loss: 1.5108 - val_mean_io_u: 0.5720
Epoch 121/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0801
- mean_io_u: 0.6399 - val_loss: 1.5646 - val_mean_io_u: 0.5865
Epoch 122/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0815
- mean_io_u: 0.6431 - val_loss: 1.5420 - val_mean_io_u: 0.5830
Epoch 123/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0804
- mean_io_u: 0.6443 - val_loss: 1.5511 - val_mean_io_u: 0.5842
Epoch 124/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0808
- mean_io_u: 0.6432 - val_loss: 1.6020 - val_mean_io_u: 0.5889
```

```
Epoch 125/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0810
- mean_io_u: 0.6442 - val_loss: 1.5344 - val_mean_io_u: 0.5757
Epoch 126/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0821
- mean_io_u: 0.6368 - val_loss: 1.5237 - val_mean_io_u: 0.5784
Epoch 127/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0803
- mean_io_u: 0.6441 - val_loss: 1.5707 - val_mean_io_u: 0.5845
Epoch 128/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0802
- mean_io_u: 0.6483 - val_loss: 1.5459 - val_mean_io_u: 0.5772
Epoch 129/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0801
- mean_io_u: 0.6496 - val_loss: 1.5388 - val_mean_io_u: 0.5814
Epoch 130/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0806
- mean_io_u: 0.6476 - val_loss: 1.5640 - val_mean_io_u: 0.5832
Epoch 131/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0793
- mean_io_u: 0.6536 - val_loss: 1.5593 - val_mean_io_u: 0.5823
Epoch 132/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0816
- mean_io_u: 0.6439 - val_loss: 1.5279 - val_mean_io_u: 0.5751
Epoch 133/300
43/43 [==============================] - 24s 550ms/step - loss: 0.0807
- mean_io_u: 0.6403 - val_loss: 1.5454 - val_mean_io_u: 0.5807
Epoch 134/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0797
- mean_io_u: 0.6483 - val_loss: 1.5533 - val_mean_io_u: 0.5726
Epoch 135/300
43/43 [==============================] - 24s 564ms/step - loss: 0.0792
- mean_io_u: 0.6529 - val_loss: 1.5685 - val_mean_io_u: 0.5846
Epoch 136/300
43/43 [==============================] - 24s 548ms/step - loss: 0.0796
- mean_io_u: 0.6544 - val_loss: 1.5349 - val_mean_io_u: 0.5727
Epoch 137/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0801
- mean_io_u: 0.6489 - val_loss: 1.5381 - val_mean_io_u: 0.5788
```

```
Epoch 138/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0799
- mean_io_u: 0.6493 - val_loss: 1.5580 - val_mean_io_u: 0.5870
Epoch 139/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0787
- mean_io_u: 0.6550 - val_loss: 1.5593 - val_mean_io_u: 0.5818
Epoch 140/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0794
- mean_io_u: 0.6551 - val_loss: 1.5592 - val_mean_io_u: 0.5761
Epoch 141/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0807
- mean_io_u: 0.6453 - val_loss: 1.4867 - val_mean_io_u: 0.5737
Epoch 142/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0809
- mean_io_u: 0.6418 - val_loss: 1.5158 - val_mean_io_u: 0.5701
Epoch 143/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0801
- mean_io_u: 0.6405 - val_loss: 1.5752 - val_mean_io_u: 0.5788
Epoch 144/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0800
- mean_io_u: 0.6438 - val_loss: 1.5065 - val_mean_io_u: 0.5716
Epoch 145/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0785
- mean_io_u: 0.6541 - val_loss: 1.5558 - val_mean_io_u: 0.5863
Epoch 146/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0793
- mean_io_u: 0.6512 - val_loss: 1.5535 - val_mean_io_u: 0.5804
Epoch 147/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0812
- mean_io_u: 0.6380 - val_loss: 1.5551 - val_mean_io_u: 0.5785
Epoch 148/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0784
- mean_io_u: 0.6516 - val_loss: 1.5881 - val_mean_io_u: 0.5899
Epoch 149/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0777
- mean_io_u: 0.6612 - val_loss: 1.5911 - val_mean_io_u: 0.5904
Epoch 150/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0779
- mean_io_u: 0.6628 - val_loss: 1.6030 - val_mean_io_u: 0.5898
```

```
Epoch 151/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0782
- mean_io_u: 0.6633 - val_loss: 1.5277 - val_mean_io_u: 0.5787
Epoch 152/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0788
- mean_io_u: 0.6536 - val_loss: 1.5678 - val_mean_io_u: 0.5854
Epoch 153/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0798
- mean_io_u: 0.6528 - val_loss: 1.5487 - val_mean_io_u: 0.5850
Epoch 154/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0787
- mean_io_u: 0.6528 - val_loss: 1.5604 - val_mean_io_u: 0.5880
Epoch 155/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0796
- mean_io_u: 0.6503 - val_loss: 1.5382 - val_mean_io_u: 0.5722
Epoch 156/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0796
- mean_io_u: 0.6464 - val_loss: 1.5623 - val_mean_io_u: 0.5781
Epoch 157/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0798
- mean_io_u: 0.6446 - val_loss: 1.5423 - val_mean_io_u: 0.5827
Epoch 158/300
43/43 [==============================] - 24s 564ms/step - loss: 0.0774
- mean_io_u: 0.6585 - val_loss: 1.5985 - val_mean_io_u: 0.5887
Epoch 159/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0778
- mean_io_u: 0.6612 - val_loss: 1.5954 - val_mean_io_u: 0.5945
Epoch 160/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0776
- mean_io_u: 0.6605 - val_loss: 1.5636 - val_mean_io_u: 0.5809
Epoch 161/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0773
- mean_io_u: 0.6632 - val_loss: 1.5843 - val_mean_io_u: 0.5903
Epoch 162/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0775
- mean_io_u: 0.6658 - val_loss: 1.6257 - val_mean_io_u: 0.5892
Epoch 163/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0792
- mean_io_u: 0.6522 - val_loss: 1.5017 - val_mean_io_u: 0.5794
```

```
Epoch 164/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0800
- mean_io_u: 0.6443 - val_loss: 1.5261 - val_mean_io_u: 0.5783
Epoch 165/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0777
- mean_io_u: 0.6546 - val_loss: 1.5942 - val_mean_io_u: 0.5880
Epoch 166/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0785
- mean_io_u: 0.6559 - val_loss: 1.5416 - val_mean_io_u: 0.5796
Epoch 167/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0778
- mean_io_u: 0.6575 - val_loss: 1.5623 - val_mean_io_u: 0.5807
Epoch 168/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0784
- mean_io_u: 0.6566 - val_loss: 1.5149 - val_mean_io_u: 0.5805
Epoch 169/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0786
- mean_io_u: 0.6538 - val_loss: 1.5165 - val_mean_io_u: 0.5795
Epoch 170/300
43/43 [==============================] - 24s 564ms/step - loss: 0.0768
- mean_io_u: 0.6622 - val_loss: 1.5880 - val_mean_io_u: 0.5878
Epoch 171/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0759
- mean_io_u: 0.6727 - val_loss: 1.5961 - val_mean_io_u: 0.5951
Epoch 172/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0764
- mean_io_u: 0.6732 - val_loss: 1.5915 - val_mean_io_u: 0.5960
Epoch 173/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0772
- mean_io_u: 0.6670 - val_loss: 1.5332 - val_mean_io_u: 0.5853
Epoch 174/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0766
- mean_io_u: 0.6687 - val_loss: 1.5660 - val_mean_io_u: 0.5870
Epoch 175/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0769
- mean_io_u: 0.6682 - val_loss: 1.5823 - val_mean_io_u: 0.5915
Epoch 176/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0771
- mean_io_u: 0.6679 - val_loss: 1.5590 - val_mean_io_u: 0.5890
```

```
Epoch 177/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0780
- mean_io_u: 0.6608 - val_loss: 1.5408 - val_mean_io_u: 0.5817
Epoch 178/300
43/43 [==============================] - 24s 550ms/step - loss: 0.0787
- mean_io_u: 0.6521 - val_loss: 1.5173 - val_mean_io_u: 0.5772
Epoch 179/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0777
- mean_io_u: 0.6546 - val_loss: 1.5530 - val_mean_io_u: 0.5839
Epoch 180/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0759
- mean_io_u: 0.6666 - val_loss: 1.5937 - val_mean_io_u: 0.5954
Epoch 181/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0763
- mean_io_u: 0.6716 - val_loss: 1.6100 - val_mean_io_u: 0.5950
Epoch 182/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0761
- mean_io_u: 0.6722 - val_loss: 1.5891 - val_mean_io_u: 0.5884
Epoch 183/300
43/43 [==============================] - 24s 566ms/step - loss: 0.0759
- mean_io_u: 0.6765 - val_loss: 1.5975 - val_mean_io_u: 0.5954
Epoch 184/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0755
- mean_io_u: 0.6768 - val_loss: 1.5748 - val_mean_io_u: 0.5901
Epoch 185/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0765
- mean_io_u: 0.6720 - val_loss: 1.6126 - val_mean_io_u: 0.5982
Epoch 186/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0788
- mean_io_u: 0.6575 - val_loss: 1.5746 - val_mean_io_u: 0.5831
Epoch 187/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0796
- mean_io_u: 0.6481 - val_loss: 1.4956 - val_mean_io_u: 0.5751
Epoch 188/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0769
- mean_io_u: 0.6604 - val_loss: 1.5943 - val_mean_io_u: 0.5975
Epoch 189/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0760
- mean_io_u: 0.6697 - val_loss: 1.5632 - val_mean_io_u: 0.5884
```

```
Epoch 190/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0756
- mean_io_u: 0.6767 - val_loss: 1.5914 - val_mean_io_u: 0.5928
Epoch 191/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0753
- mean_io_u: 0.6789 - val_loss: 1.6066 - val_mean_io_u: 0.5951
Epoch 192/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0768
- mean_io_u: 0.6706 - val_loss: 1.5510 - val_mean_io_u: 0.5844
Epoch 193/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0773
- mean_io_u: 0.6616 - val_loss: 1.5810 - val_mean_io_u: 0.5883
Epoch 194/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0766
- mean_io_u: 0.6678 - val_loss: 1.5636 - val_mean_io_u: 0.5865
Epoch 195/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0751
- mean_io_u: 0.6741 - val_loss: 1.5545 - val_mean_io_u: 0.5895
Epoch 196/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0755
- mean_io_u: 0.6766 - val_loss: 1.5576 - val_mean_io_u: 0.5932
Epoch 197/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0752
- mean_io_u: 0.6779 - val_loss: 1.5947 - val_mean_io_u: 0.5949
Epoch 198/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0755
- mean_io_u: 0.6783 - val_loss: 1.5962 - val_mean_io_u: 0.5909
Epoch 199/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0762
- mean_io_u: 0.6745 - val_loss: 1.5331 - val_mean_io_u: 0.5863
Epoch 200/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0758
- mean_io_u: 0.6743 - val_loss: 1.5803 - val_mean_io_u: 0.5941
Epoch 201/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0757
- mean_io_u: 0.6755 - val_loss: 1.6395 - val_mean_io_u: 0.6014
Epoch 202/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0754
- mean_io_u: 0.6798 - val_loss: 1.5453 - val_mean_io_u: 0.5843
```

```
Epoch 203/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0757
- mean_io_u: 0.6751 - val_loss: 1.5771 - val_mean_io_u: 0.5904
Epoch 204/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0759
- mean_io_u: 0.6730 - val_loss: 1.5967 - val_mean_io_u: 0.5966
Epoch 205/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0761
- mean_io_u: 0.6725 - val_loss: 1.5494 - val_mean_io_u: 0.5923
Epoch 206/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0766
- mean_io_u: 0.6682 - val_loss: 1.5947 - val_mean_io_u: 0.5882
Epoch 207/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0760
- mean_io_u: 0.6711 - val_loss: 1.5954 - val_mean_io_u: 0.5946
Epoch 208/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0746
- mean_io_u: 0.6819 - val_loss: 1.5548 - val_mean_io_u: 0.5906
Epoch 209/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0749
- mean_io_u: 0.6817 - val_loss: 1.5876 - val_mean_io_u: 0.5927
Epoch 210/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0748
- mean_io_u: 0.6831 - val_loss: 1.6139 - val_mean_io_u: 0.6005
Epoch 211/300
43/43 [==============================] - 24s 550ms/step - loss: 0.0746
- mean_io_u: 0.6831 - val_loss: 1.5830 - val_mean_io_u: 0.5921
Epoch 212/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0746
- mean_io_u: 0.6861 - val_loss: 1.5998 - val_mean_io_u: 0.5961
Epoch 213/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0752
- mean_io_u: 0.6832 - val_loss: 1.5818 - val_mean_io_u: 0.5929
Epoch 214/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0752
- mean_io_u: 0.6807 - val_loss: 1.5591 - val_mean_io_u: 0.5931
Epoch 215/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0764
- mean_io_u: 0.6712 - val_loss: 1.5660 - val_mean_io_u: 0.5888
```

```
Epoch 216/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0748
- mean_io_u: 0.6794 - val_loss: 1.5820 - val_mean_io_u: 0.5977
Epoch 217/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0736
- mean_io_u: 0.6869 - val_loss: 1.6140 - val_mean_io_u: 0.5983
Epoch 218/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0747
- mean_io_u: 0.6867 - val_loss: 1.5996 - val_mean_io_u: 0.5952
Epoch 219/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0753
- mean_io_u: 0.6829 - val_loss: 1.5570 - val_mean_io_u: 0.5917
Epoch 220/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0748
- mean_io_u: 0.6820 - val_loss: 1.6200 - val_mean_io_u: 0.5965
Epoch 221/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0750
- mean_io_u: 0.6836 - val_loss: 1.5908 - val_mean_io_u: 0.5898
Epoch 222/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0758
- mean_io_u: 0.6772 - val_loss: 1.5544 - val_mean_io_u: 0.5870
Epoch 223/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0759
- mean_io_u: 0.6723 - val_loss: 1.5780 - val_mean_io_u: 0.5925
Epoch 224/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0750
- mean_io_u: 0.6784 - val_loss: 1.5644 - val_mean_io_u: 0.5937
Epoch 225/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0749
- mean_io_u: 0.6806 - val_loss: 1.5660 - val_mean_io_u: 0.5889
Epoch 226/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0749
- mean_io_u: 0.6809 - val_loss: 1.6261 - val_mean_io_u: 0.6045
Epoch 227/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0744
- mean_io_u: 0.6842 - val_loss: 1.5936 - val_mean_io_u: 0.5972
Epoch 228/300
43/43 [==============================] - 25s 570ms/step - loss: 0.0748
- mean_io_u: 0.6824 - val_loss: 1.5772 - val_mean_io_u: 0.5942
```

```
Epoch 229/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0755
- mean_io_u: 0.6786 - val_loss: 1.5476 - val_mean_io_u: 0.5932
Epoch 230/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0786
- mean_io_u: 0.6549 - val_loss: 1.4755 - val_mean_io_u: 0.5678
Epoch 231/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0768
- mean_io_u: 0.6600 - val_loss: 1.5490 - val_mean_io_u: 0.5884
Epoch 232/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0741
- mean_io_u: 0.6798 - val_loss: 1.5879 - val_mean_io_u: 0.5942
Epoch 233/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0726
- mean_io_u: 0.6906 - val_loss: 1.6193 - val_mean_io_u: 0.5991
Epoch 234/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0740
- mean_io_u: 0.6876 - val_loss: 1.5939 - val_mean_io_u: 0.5951
Epoch 235/300
43/43 [==============================] - 24s 549ms/step - loss: 0.0739
- mean_io_u: 0.6899 - val_loss: 1.5826 - val_mean_io_u: 0.5984
Epoch 236/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0725
- mean_io_u: 0.6958 - val_loss: 1.6216 - val_mean_io_u: 0.6036
Epoch 237/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0736
- mean_io_u: 0.6947 - val_loss: 1.6133 - val_mean_io_u: 0.6042
Epoch 238/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0728
- mean_io_u: 0.6981 - val_loss: 1.5973 - val_mean_io_u: 0.6024
Epoch 239/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0740
- mean_io_u: 0.6947 - val_loss: 1.6257 - val_mean_io_u: 0.6022
Epoch 240/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0740
- mean_io_u: 0.6930 - val_loss: 1.5645 - val_mean_io_u: 0.5970
Epoch 241/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0739
- mean_io_u: 0.6891 - val_loss: 1.5885 - val_mean_io_u: 0.5983
```

```
Epoch 242/300
43/43 [==============================] - 24s 553ms/step - loss: 0.0745
- mean_io_u: 0.6889 - val_loss: 1.5802 - val_mean_io_u: 0.6014
Epoch 243/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0743
- mean_io_u: 0.6851 - val_loss: 1.5510 - val_mean_io_u: 0.5889
Epoch 244/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0754
- mean_io_u: 0.6794 - val_loss: 1.5777 - val_mean_io_u: 0.5875
Epoch 245/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0743
- mean_io_u: 0.6819 - val_loss: 1.6010 - val_mean_io_u: 0.6027
Epoch 246/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0739
- mean_io_u: 0.6865 - val_loss: 1.5883 - val_mean_io_u: 0.5959
Epoch 247/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0735
- mean_io_u: 0.6910 - val_loss: 1.6060 - val_mean_io_u: 0.6026
Epoch 248/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0728
- mean_io_u: 0.6980 - val_loss: 1.6038 - val_mean_io_u: 0.6003
Epoch 249/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0742
- mean_io_u: 0.6916 - val_loss: 1.5837 - val_mean_io_u: 0.5972
Epoch 250/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0737
- mean_io_u: 0.6913 - val_loss: 1.5531 - val_mean_io_u: 0.5900
Epoch 251/300
43/43 [==============================] - 24s 550ms/step - loss: 0.0740
- mean_io_u: 0.6901 - val_loss: 1.5723 - val_mean_io_u: 0.5909
Epoch 252/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0743
- mean_io_u: 0.6861 - val_loss: 1.6026 - val_mean_io_u: 0.6008
Epoch 253/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0740
- mean_io_u: 0.6877 - val_loss: 1.5789 - val_mean_io_u: 0.5924
Epoch 254/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0733
- mean_io_u: 0.6919 - val_loss: 1.6054 - val_mean_io_u: 0.6017
```

```
Epoch 255/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0735
- mean_io_u: 0.6945 - val_loss: 1.6083 - val_mean_io_u: 0.6028
Epoch 256/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0731
- mean_io_u: 0.6958 - val_loss: 1.6097 - val_mean_io_u: 0.6039
Epoch 257/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0733
- mean_io_u: 0.6965 - val_loss: 1.5840 - val_mean_io_u: 0.5979
Epoch 258/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0728
- mean_io_u: 0.6986 - val_loss: 1.6218 - val_mean_io_u: 0.6064
Epoch 259/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0722
- mean_io_u: 0.7015 - val_loss: 1.6427 - val_mean_io_u: 0.6101
Epoch 260/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0734
- mean_io_u: 0.6974 - val_loss: 1.5691 - val_mean_io_u: 0.5943
Epoch 261/300
43/43 [==============================] - 24s 551ms/step - loss: 0.0742
- mean_io_u: 0.6896 - val_loss: 1.5900 - val_mean_io_u: 0.6020
Epoch 262/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0725
- mean_io_u: 0.6987 - val_loss: 1.6249 - val_mean_io_u: 0.6072
Epoch 263/300
43/43 [==============================] - 24s 564ms/step - loss: 0.0721
- mean_io_u: 0.7041 - val_loss: 1.6157 - val_mean_io_u: 0.6102
Epoch 264/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0734
- mean_io_u: 0.7012 - val_loss: 1.5659 - val_mean_io_u: 0.5988
Epoch 265/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0751
- mean_io_u: 0.6829 - val_loss: 1.5995 - val_mean_io_u: 0.6025
Epoch 266/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0733
- mean_io_u: 0.6916 - val_loss: 1.5879 - val_mean_io_u: 0.6008
Epoch 267/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0730
- mean_io_u: 0.6971 - val_loss: 1.5893 - val_mean_io_u: 0.6007
```

```
Epoch 268/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0724
- mean_io_u: 0.6981 - val_loss: 1.6049 - val_mean_io_u: 0.6043
Epoch 269/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0727
- mean_io_u: 0.7016 - val_loss: 1.5784 - val_mean_io_u: 0.6002
Epoch 270/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0724
- mean_io_u: 0.7005 - val_loss: 1.6034 - val_mean_io_u: 0.6057
Epoch 271/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0723
- mean_io_u: 0.7012 - val_loss: 1.6296 - val_mean_io_u: 0.6066
Epoch 272/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0726
- mean_io_u: 0.7024 - val_loss: 1.6229 - val_mean_io_u: 0.6056
Epoch 273/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0726
- mean_io_u: 0.7013 - val_loss: 1.5944 - val_mean_io_u: 0.6044
Epoch 274/300
43/43 [==============================] - 24s 550ms/step - loss: 0.0730
- mean_io_u: 0.7014 - val_loss: 1.6156 - val_mean_io_u: 0.6089
Epoch 275/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0721
- mean_io_u: 0.7022 - val_loss: 1.6443 - val_mean_io_u: 0.6127
Epoch 276/300
43/43 [==============================] - 24s 565ms/step - loss: 0.0719
- mean_io_u: 0.7076 - val_loss: 1.6174 - val_mean_io_u: 0.6059
Epoch 277/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0727
- mean_io_u: 0.7048 - val_loss: 1.5904 - val_mean_io_u: 0.6049
Epoch 278/300
43/43 [==============================] - 24s 547ms/step - loss: 0.0727
- mean_io_u: 0.7013 - val_loss: 1.6032 - val_mean_io_u: 0.6051
Epoch 279/300
43/43 [==============================] - 24s 560ms/step - loss: 0.0727
- mean_io_u: 0.7043 - val_loss: 1.6180 - val_mean_io_u: 0.6064
Epoch 280/300
43/43 [==============================] - 24s 549ms/step - loss: 0.0718
- mean_io_u: 0.7074 - val_loss: 1.6092 - val_mean_io_u: 0.6060
```

```
Epoch 281/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0720
- mean_io_u: 0.7079 - val_loss: 1.6337 - val_mean_io_u: 0.6096
Epoch 282/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0728
- mean_io_u: 0.7054 - val_loss: 1.5776 - val_mean_io_u: 0.5992
Epoch 283/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0736
- mean_io_u: 0.6981 - val_loss: 1.5912 - val_mean_io_u: 0.5997
Epoch 284/300
43/43 [==============================] - 24s 552ms/step - loss: 0.0725
- mean_io_u: 0.7009 - val_loss: 1.6255 - val_mean_io_u: 0.6128
Epoch 285/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0718
- mean_io_u: 0.7081 - val_loss: 1.6134 - val_mean_io_u: 0.6068
Epoch 286/300
43/43 [==============================] - 24s 561ms/step - loss: 0.0716
- mean_io_u: 0.7097 - val_loss: 1.6254 - val_mean_io_u: 0.6111
Epoch 287/300
43/43 [==============================] - 24s 547ms/step - loss: 0.0722
- mean_io_u: 0.7088 - val_loss: 1.6292 - val_mean_io_u: 0.6127
Epoch 288/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0732
- mean_io_u: 0.7022 - val_loss: 1.5814 - val_mean_io_u: 0.6005
Epoch 289/300
43/43 [==============================] - 24s 562ms/step - loss: 0.0731
- mean_io_u: 0.7010 - val_loss: 1.6056 - val_mean_io_u: 0.6068
Epoch 290/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0722
- mean_io_u: 0.7029 - val_loss: 1.6117 - val_mean_io_u: 0.6074
Epoch 291/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0726
- mean_io_u: 0.7033 - val_loss: 1.6059 - val_mean_io_u: 0.6091
Epoch 292/300
43/43 [==============================] - 24s 558ms/step - loss: 0.0729
- mean_io_u: 0.7024 - val_loss: 1.5841 - val_mean_io_u: 0.6047
Epoch 293/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0741
- mean_io_u: 0.6901 - val_loss: 1.5771 - val_mean_io_u: 0.6012
```

```
Epoch 294/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0723
- mean_io_u: 0.6989 - val_loss: 1.6102 - val_mean_io_u: 0.6037
Epoch 295/300
43/43 [==============================] - 24s 563ms/step - loss: 0.0718
- mean_io_u: 0.7069 - val_loss: 1.6138 - val_mean_io_u: 0.6067
Epoch 296/300
43/43 [==============================] - 24s 559ms/step - loss: 0.0712
- mean_io_u: 0.7102 - val_loss: 1.6232 - val_mean_io_u: 0.6099
Epoch 297/300
43/43 [==============================] - 24s 555ms/step - loss: 0.0717
- mean_io_u: 0.7089 - val_loss: 1.6489 - val_mean_io_u: 0.6128
Epoch 298/300
43/43 [==============================] - 24s 557ms/step - loss: 0.0720
- mean_io_u: 0.7088 - val_loss: 1.6266 - val_mean_io_u: 0.6098
Epoch 299/300
43/43 [==============================] - 24s 554ms/step - loss: 0.0711
- mean_io_u: 0.7130 - val_loss: 1.6464 - val_mean_io_u: 0.6146
Epoch 300/300
43/43 [==============================] - 24s 556ms/step - loss: 0.0724
- mean_io_u: 0.7073 - val_loss: 1.5954 - val_mean_io_u: 0.6051
```

In [42]:

```
# Saving model
from keras.models import load_model
model.save('my_model.h5py')  # creates a h5py file 'my_model.h5'
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorfl
ow/python/ops/resource_variable_ops.py:1817: calling BaseResourceVariab
le.__init__ (from tensorflow.python.ops.resource_variable_ops) with con
straint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
INFO:tensorflow:Assets written to: my_model.h5py/assets
```

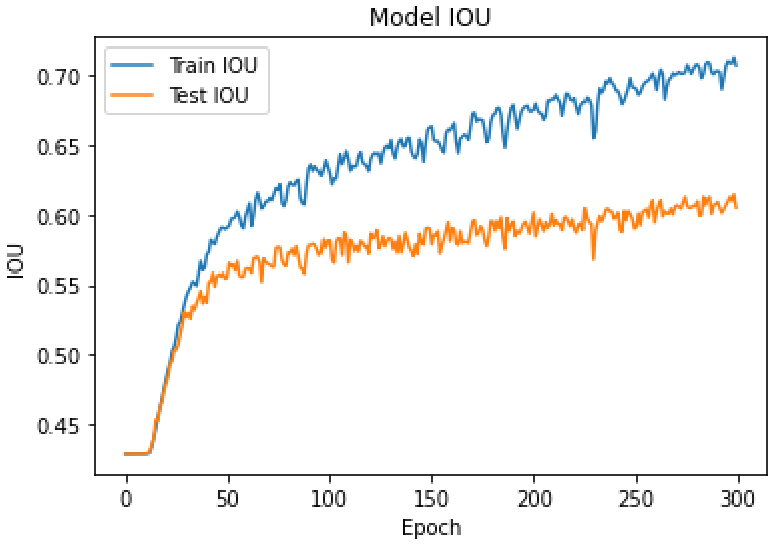
In [43]:

```
print(hist2.history.keys())
```

```
dict_keys(['loss', 'mean_io_u', 'val_loss', 'val_mean_io_u'])
```

In [44]:

```
plt.plot(hist2.history['mean_io_u'])
plt.plot(hist2.history['val_mean_io_u'])
plt.title('Model IOU')
plt.ylabel('IOU')
plt.xlabel('Epoch')
plt.legend(['Train IOU', 'Test IOU'], loc='upper left')
plt.show()
```

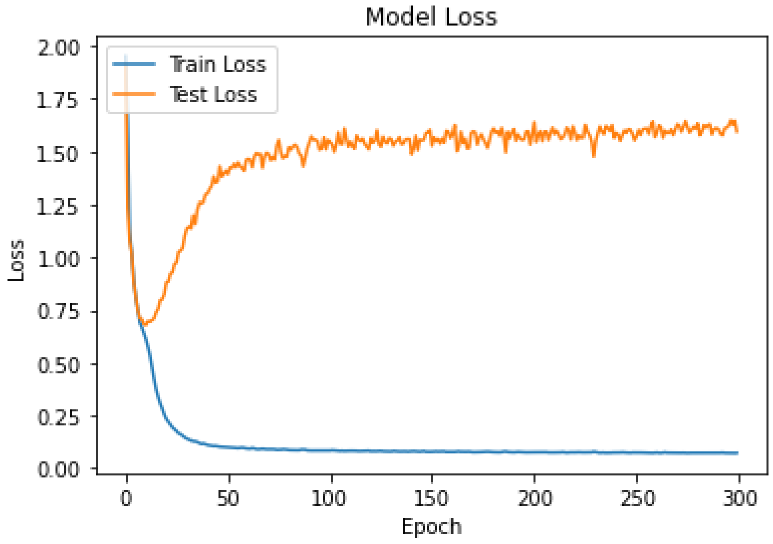


Conclusion:

1.Looks good as we are getting 60% Mean IOU after 300 epochs

In [45]:

```
plt.plot(hist2.history['loss'])
plt.plot(hist2.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train Loss', 'Test Loss'], loc='upper left')
plt.show()
```



In [0]:

```
#Prediction on test dataset
y_pred = model.predict(test_dataset)
```

In [0]:

```
y_predi = tf.argmax(y_pred, axis=3)
```

In [78]:

```
y_predi.shape
```

Out[78]:

```
TensorShape([204, 224, 224])
```

In [2]:

```
import numpy as np
# Assigning some RGB colors for the 7 + 1 (Misc) classes
colors = np.array([
    [255, 192, 203],      # Drivable
    [244, 35, 232],     # Non Drivable
    [220, 20, 60],      # Living Things
    [0, 0, 230],        # Vehicles
    [220, 190, 40],     # Road Side Objects
    [70, 70, 70],       # Far Objects
    [70, 130, 180],     # Sky
    [0, 0, 0]           # Misc
], dtype=np.int)
```

In [0]:

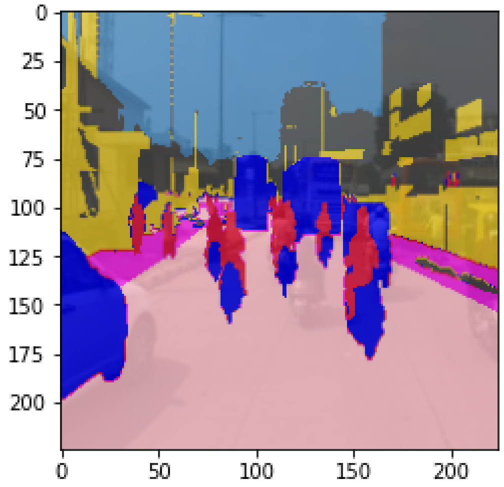
```
import matplotlib.pyplot as plt
from matplotlib.pyplot import imread
import numpy as np
from glob import glob
image_paths = glob('idd20k_lite/leftImg8bit/train/*/*_image.jpg')
label_paths = [p.replace('leftImg8bit', 'gtFine').replace('_image.jpg', '_label.png') for p in image_paths]
image_paths_val = glob('idd20k_lite/leftImg8bit/val/*/*_image.jpg')
label_paths_val = [p.replace('leftImg8bit', 'gtFine').replace('_image.jpg', '_label.png') for p in image_paths_val]

#This function returns the original image, ground truth and predicted output
def output(image_no):
    label_map = imread(label_paths_val[image_no])
    image_frame = imread(image_paths_val[image_no])
    color_image = np.zeros((label_map.shape[0], label_map.shape[1], 3), dtype=np.int)
    for i in range(7):
        color_image[label_map == i] = colors[i]
    color_image[label_map == 255] = colors[7]
    plt.imshow(image_frame)
    #plt.imshow(color_image, alpha=0.5)
    print('Original image is')
    plt.show()


    label_map=label_paths_val[image_no]
    image_frame = imread(image_paths_val[image_no])
    color_image = np.zeros((label_map.shape[0], label_map.shape[1], 3), dtype=np.int)
    for i in range(7):
        color_image[label_map == i] = colors[i]
    color_image[label_map == 255] = colors[7]
    plt.imshow(image_frame)
    plt.imshow(color_image, alpha=0.8)
    print('Original masked image is')
    plt.show()


    label_map=y_predi[image_no]
    image_frame = imread(image_paths_val[image_no])
    color_image = np.zeros((label_map.shape[0], label_map.shape[1], 3),
dtype=np.int)
    for i in range(7):
        color_image[label_map == i] = colors[i]
    color_image[label_map == 255] = colors[7]
    plt.imshow(image_frame)
    plt.imshow(color_image, alpha=0.8)
    print('Predicted masked image')
    plt.show()
```

In [126]:

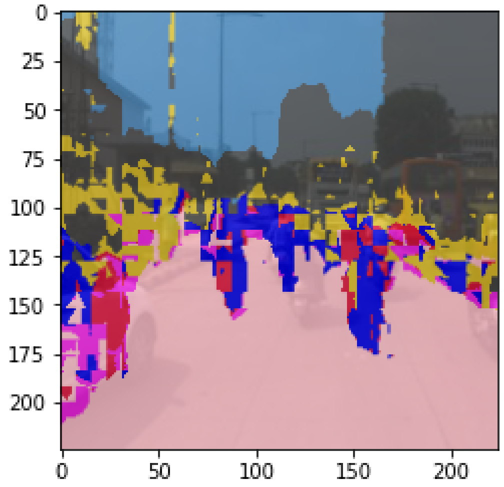
```
output(1)
```

```
Original image is
```

```
Original masked image is
```



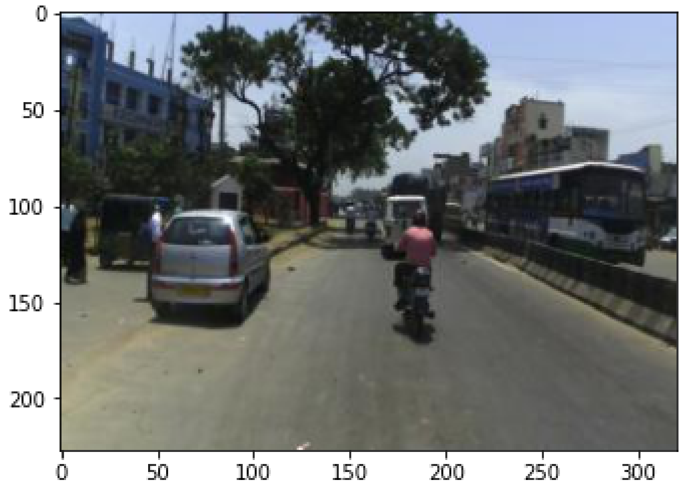
```
Predicted masked image
```



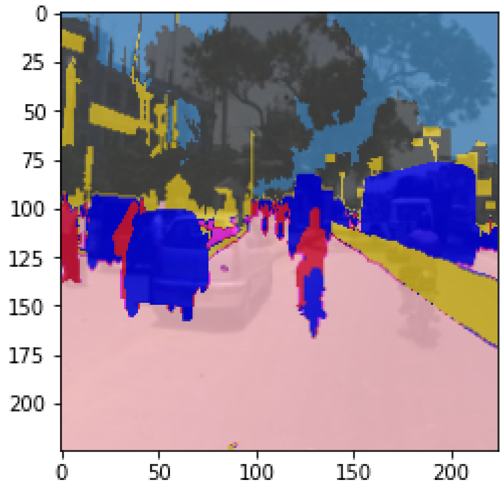
In [128]:

```
output(106)
```

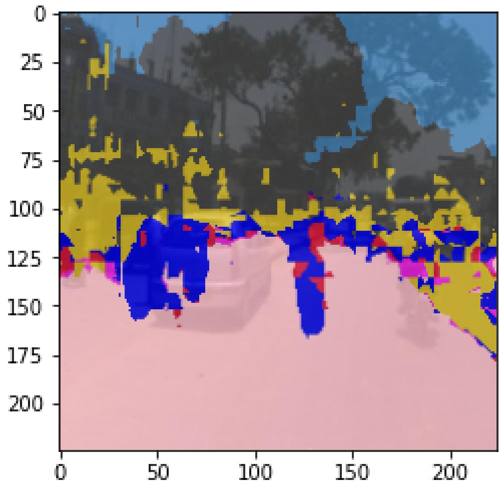
```
Original image is
```



```
Original masked image is
```



```
Predicted masked image
```



Conclusion:

1. Good predictions for highly weighted classes

```
In [ ]:
```