In [28]:

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [29]:
```
import tensorflow as tf
import tensorflow.keras.layers as tfl
import os
from tqdm import tqdm
import cv2
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from PIL import Image
from os.path import splitext
```

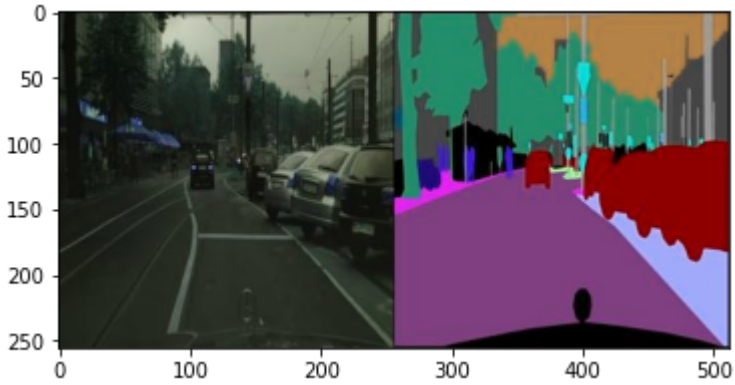
In [30]:
```
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
path="/content/drive/MyDrive/cityscapes_data"
def images_upload(path):
    images=[]
    for root,subfolders,files in os.walk(path):
        for file in tqdm(files):
            filename=root+os.sep+file
            if filename.endswith('jpg') or filename.endwith('png'):
                images.append(filename)
    return images
images=images_upload(path)
```

```
0it [00:00, ?it/s]
100%|██████████| 500/500 [00:00<00:00, 347440.69it/s]
100%|██████████| 1330/1330 [00:00<00:00, 681043.14it/s]
```

In [31]:
```
def convert_image_rgb(data):
    imgs=[]
    for i in tqdm(data):
        img = cv2.imread(i,cv2.COLOR_BGR2RGB)
        del i
        imgs.append(img)
    return imgs
img=convert_image_rgb(images)
```

```
100%|██████████| 1830/1830 [00:17<00:00, 106.43it/s]
```

In [32]:
```
plt.imshow(img[np.random.randint(0,len(img))]);
```

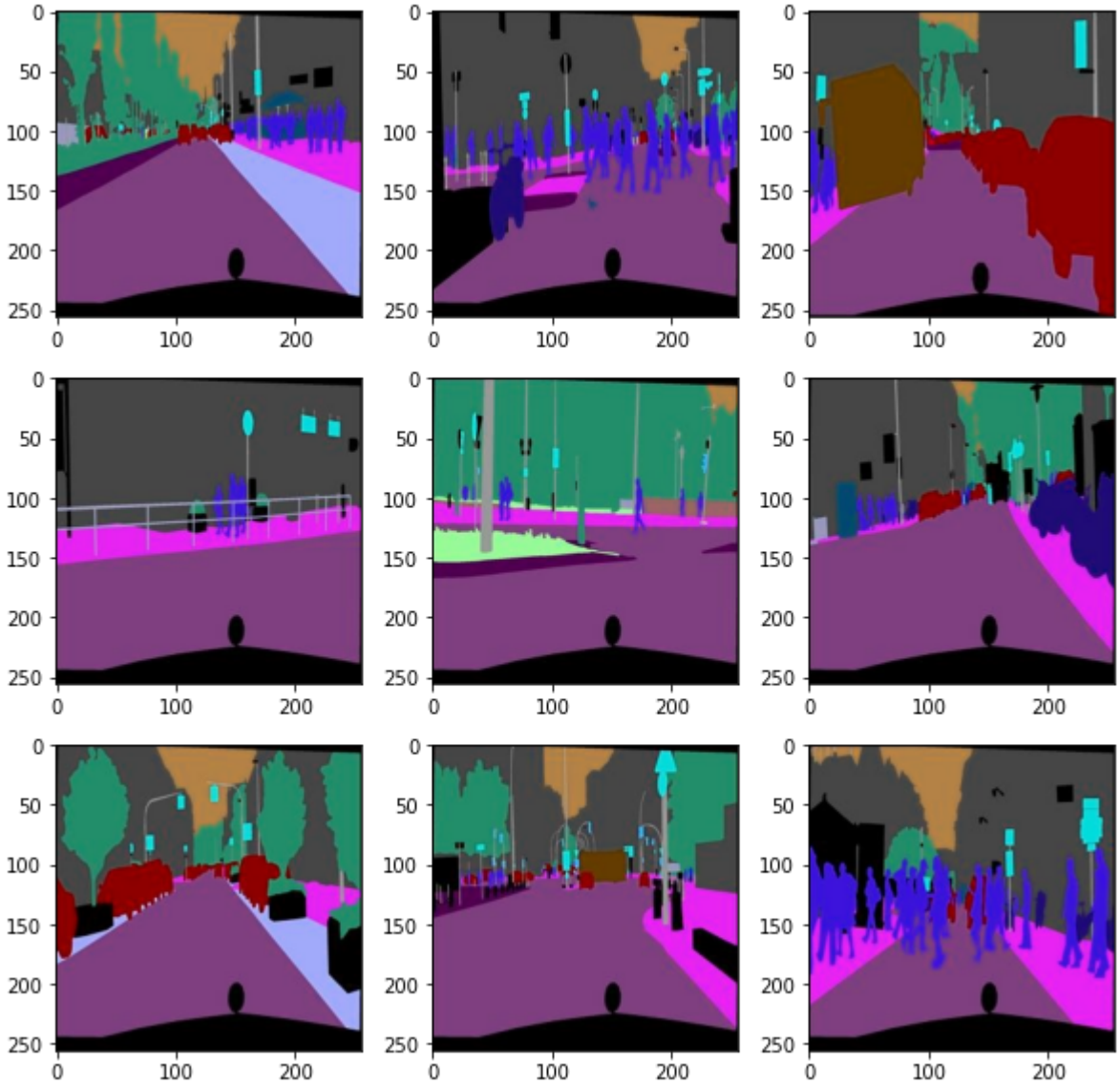


In [34]:
```
def split_input_mask(data):
    inputs=[]
    mask=[]
    for i in data:
        a=i[:,:256]
        inputs.append(a)
        b=i[:,256:]
        mask.append(b)
    return inputs,mask
inputs,mask=split_input_mask(img)
```

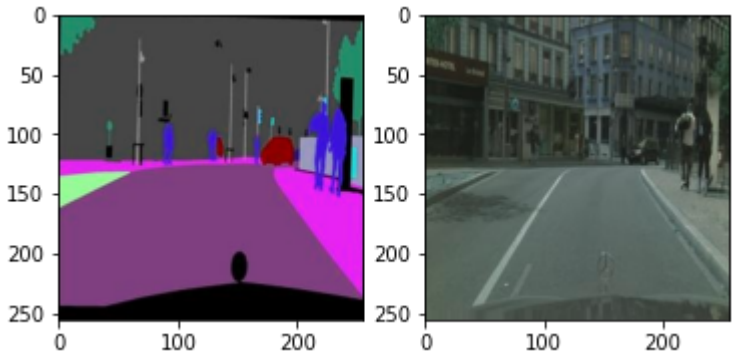
In [35]:
```
del images
del img
```

In [36]:
```
def show_images(data):
    plt.figure(figsize=(10,10))
    for i in range(9):
        idx=np.random.randint(0,len(data))
        plt.subplot(3,3,i+1)
        img=data[idx]
        plt.imshow(img)
show_images(inputs)
```

In [37]:
```
show_images(mask)
```

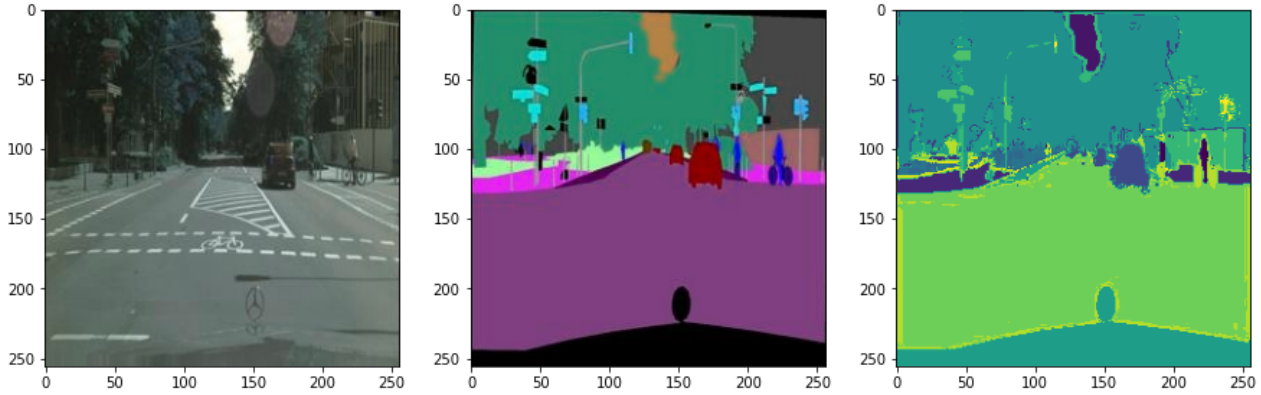


In [38]:
```
def images_compare(inputs,mask):
    idx_new=np.random.randint(0,len(mask))
    fig = plt.figure()
    ax1 = fig.add_subplot(1,2,1)
    ax1.imshow(mask[idx_new])
    ax2 = fig.add_subplot(1,2,2)
    ax2.imshow(inputs[idx_new],cmap='gray')
    plt.show()
images_compare(inputs,mask)
```



In [39]:
```
num_items = 1000
```

```
color_array = np.random.choice(range(256), 3*num_items).reshape(-1,3)
num_classes = 20
label_model = KMeans(n_clusters = num_classes)
label_model.fit(color_array)
label_class = label_model.predict(mask[10].reshape(-1,3)).reshape(256,256)
fig, axes = plt.subplots(1,3,figsize=(15,5))
axes[0].imshow(inputs[10]);
axes[1].imshow(mask[10]);
axes[2].imshow(label_class);
```

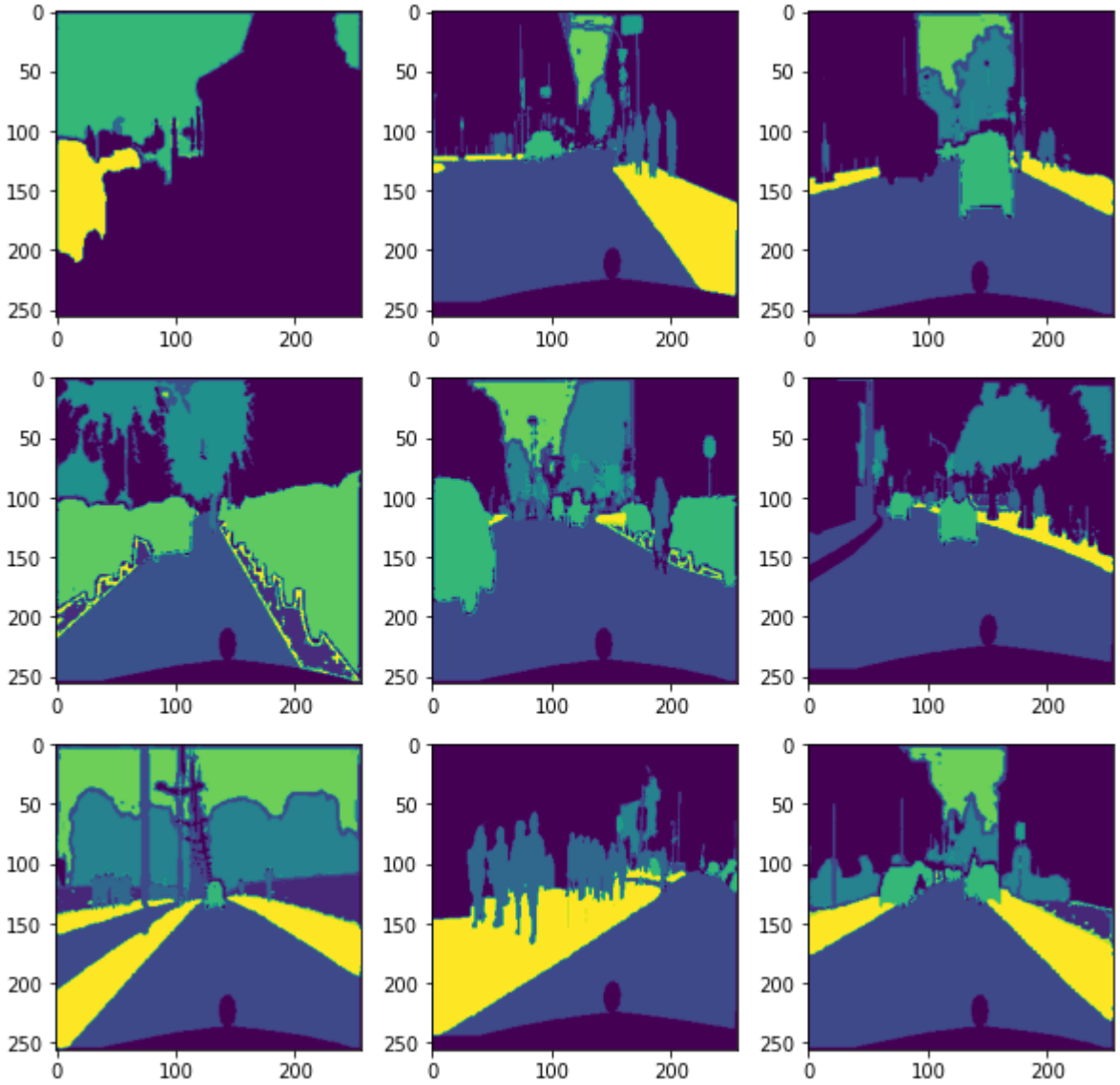


In [40]:
```
def new_labels(mask):
    num_items = 1000
    color_array = np.random.choice(range(256), 3*num_items).reshape(-1,3)
    num_classes = 10
    label_model = KMeans(n_clusters = num_classes)
    label_model.fit(color_array)
    labels=[]
    for i in tqdm(range(len(mask))):
        label_class = label_model.predict(mask[i].reshape(-1,3)).reshape(256,256)
        labels.append(label_class)
    return labels
```

In [41]:
```
labels=new_labels(mask)
idx=np.random.randint(0,len(labels))
classes,freq=np.unique(labels[idx],return_counts=True)
print(f'number of classes :{len(classes)}')
```

```
100%|██████████| 1830/1830 [00:08<00:00, 224.34it/s]
number of classes :10
```

In [42]:
```
plt.figure(figsize=(10,10))
for i in range(9):
        plt.subplot(3,3,i+1)
        im=labels[i]
        plt.imshow(im)
# plt.imshow(a)
```



In [43]:
```
def rescale(data):
    rescaled=[]
    for i in tqdm(data):
        img=tf.image.convert_image_dtype(i, tf.float32)
        del i
        rescaled.append(img)
    return rescaled
```

In [44]:
```
rescaled_input=rescale(inputs)
```

```
100%|██████████| 1830/1830 [00:01<00:00, 1513.65it/s]
```

In [45]:
```
def build_model(inputsize=(256,256,3),classes=None):
    inputs = tf.keras.Input(shape=(inputsize))

    conv = tfl.Conv2D(32, (3, 3), padding="same",kernel_initializer='he_normal', name='
        inputs)
    x=tfl.BatchNormalization()(conv)
    x=tfl.LeakyReLU()(x)
    x1 = tfl.Conv2D(32, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
        x)
    x=tfl.BatchNormalization()(x1)
    x=tfl.LeakyReLU()(x)
```

```
    x = tfl.MaxPool2D(pool_size=(2, 2), strides=(2, 2), name='MaxPool1')(x)

    x = tfl.Conv2D(64, (3, 3), padding="same",kernel_initializer='he_normal', name='Con
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x2 = tfl.Conv2D(64, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x2)
    x=tfl.LeakyReLU()(x)
    x = tfl.MaxPool2D(pool_size=(2, 2), name='MaxPool2')(x)

    x = tfl.Conv2D(128, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x3 = tfl.Conv2D(128, (3, 3), padding="same",kernel_initializer='he_normal', name='C
    x=tfl.BatchNormalization()(x3)
    x=tfl.LeakyReLU()(x)
    x = tfl.MaxPool2D(pool_size=(2, 2), strides=(2, 2), name='MaxPool3')(x)


    x = tfl.Conv2D(256, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x = tfl.Conv2D(256, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    x = tfl.Conv2DTranspose(128, (3, 3), strides=2, padding="same")(x)
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    x = tfl.concatenate([x, x3], axis=3)

    x = tfl.Conv2D(128, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x = tfl.Conv2D(128, (3, 3), padding="same",kernel_initializer='he_normal', name='Co
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x = tfl.Conv2DTranspose(64, (3, 3), strides=2, padding="same")(x)
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    x = tfl.concatenate([x, x2], axis=3)

    x = tfl.Conv2D(64, (3, 3), padding="same",kernel_initializer='he_normal', name='Con
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
    x = tfl.Conv2D(64, (3, 3), padding="same",kernel_initializer='he_normal', name='Con
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    x = tfl.Conv2DTranspose(32, (3, 3), strides=2, padding="same")(x)
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    x = tfl.concatenate([x, x1], axis=3)

    x = tfl.Conv2D(32, (3, 3), padding="same",kernel_initializer='he_normal', name='Con
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)
```

```
    x = tfl.Conv2D(32, (3, 3), padding="same",kernel_initializer='he_normal', name='Con
    x=tfl.BatchNormalization()(x)
    x=tfl.LeakyReLU()(x)

    outputs = tfl.Conv2D(classes, (1, 1), padding="same", activation='softmax', name='O
    final_model = tf.keras.Model(inputs=inputs, outputs=outputs)
    final_model.summary()
    return final_model
```

In [46]:
```
mymodel=build_model(classes=10)
```

```
Model: "model_1"
_________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=========================================================================================
 input_2 (InputLayer)        [(None, 256, 256, 3       0         []
                             )]

 Conv1 (Conv2D)              (None, 256, 256, 32       896       ['input_2[0][0]']
                             )

 batch_normalization_17 (BatchN  (None, 256, 256, 32   128       ['Conv1[0][0]']
 ormalization)               )

 leaky_re_lu_17 (LeakyReLU)  (None, 256, 256, 32       0         ['batch_normalization_1
7[0][0]']
                             )

 Conv2 (Conv2D)              (None, 256, 256, 32       9248      ['leaky_re_lu_17[0]
[0]']
                             )

 batch_normalization_18 (BatchN  (None, 256, 256, 32   128       ['Conv2[0][0]']
 ormalization)               )

 leaky_re_lu_18 (LeakyReLU)  (None, 256, 256, 32       0         ['batch_normalization_1
8[0][0]']
                             )

 MaxPool1 (MaxPooling2D)     (None, 128, 128, 32       0         ['leaky_re_lu_18[0]
[0]']
                             )

 Conv3 (Conv2D)              (None, 128, 128, 64       18496     ['MaxPool1[0][0]']
                             )

 batch_normalization_19 (BatchN  (None, 128, 128, 64   256       ['Conv3[0][0]']
 ormalization)               )

 leaky_re_lu_19 (LeakyReLU)  (None, 128, 128, 64       0         ['batch_normalization_1
9[0][0]']
                             )

 Conv4 (Conv2D)              (None, 128, 128, 64       36928     ['leaky_re_lu_19[0]
[0]']
                             )

 batch_normalization_20 (BatchN  (None, 128, 128, 64   256       ['Conv4[0][0]']
```

```
 ormalization)                  )

 leaky_re_lu_20 (LeakyReLU)     (None, 128, 128, 64  0           ['batch_normalization_2
0[0][0]']
                                )

 MaxPool2 (MaxPooling2D)        (None, 64, 64, 64)   0           ['leaky_re_lu_20[0]
[0]']

 Conv5 (Conv2D)                 (None, 64, 64, 128)  73856       ['MaxPool2[0][0]']

 batch_normalization_21 (BatchN  (None, 64, 64, 128)  512        ['Conv5[0][0]']
 ormalization)

 leaky_re_lu_21 (LeakyReLU)     (None, 64, 64, 128)  0           ['batch_normalization_2
1[0][0]']

 Conv6 (Conv2D)                 (None, 64, 64, 128)  147584      ['leaky_re_lu_21[0]
[0]']

 batch_normalization_22 (BatchN  (None, 64, 64, 128)  512        ['Conv6[0][0]']
 ormalization)

 leaky_re_lu_22 (LeakyReLU)     (None, 64, 64, 128)  0           ['batch_normalization_2
2[0][0]']

 MaxPool3 (MaxPooling2D)        (None, 32, 32, 128)  0           ['leaky_re_lu_22[0]
[0]']

 Conv7 (Conv2D)                 (None, 32, 32, 256)  295168      ['MaxPool3[0][0]']

 batch_normalization_23 (BatchN  (None, 32, 32, 256)  1024       ['Conv7[0][0]']
 ormalization)

 leaky_re_lu_23 (LeakyReLU)     (None, 32, 32, 256)  0           ['batch_normalization_2
3[0][0]']

 Conv8 (Conv2D)                 (None, 32, 32, 256)  590080      ['leaky_re_lu_23[0]
[0]']

 batch_normalization_24 (BatchN  (None, 32, 32, 256)  1024       ['Conv8[0][0]']
 ormalization)

 leaky_re_lu_24 (LeakyReLU)     (None, 32, 32, 256)  0           ['batch_normalization_2
4[0][0]']

 conv2d_transpose_3 (Conv2DTran  (None, 64, 64, 128)  295040     ['leaky_re_lu_24[0]
[0]']
 spose)

 batch_normalization_25 (BatchN  (None, 64, 64, 128)  512        ['conv2d_transpose_3[0]
[0]']
 ormalization)

 leaky_re_lu_25 (LeakyReLU)     (None, 64, 64, 128)  0           ['batch_normalization_2
5[0][0]']

 concatenate_3 (Concatenate)    (None, 64, 64, 256)  0           ['leaky_re_lu_25[0]
[0]',
                                                                  'Conv6[0][0]']
```

```
 Conv9 (Conv2D)                (None, 64, 64, 128)  295040      ['concatenate_3[0][0]']

 batch_normalization_26 (BatchN  (None, 64, 64, 128)  512        ['Conv9[0][0]']
 ormalization)

 leaky_re_lu_26 (LeakyReLU)    (None, 64, 64, 128)  0          ['batch_normalization_2
6[0][0]']

 Conv10 (Conv2D)               (None, 64, 64, 128)  147584      ['leaky_re_lu_26[0]
[0]']

 batch_normalization_27 (BatchN  (None, 64, 64, 128)  512        ['Conv10[0][0]']
 ormalization)

 leaky_re_lu_27 (LeakyReLU)    (None, 64, 64, 128)  0          ['batch_normalization_2
7[0][0]']

 conv2d_transpose_4 (Conv2DTran  (None, 128, 128, 64  73792      ['leaky_re_lu_27[0]
[0]']
 spose)                        )

 batch_normalization_28 (BatchN  (None, 128, 128, 64  256        ['conv2d_transpose_4[0]
[0]']
 ormalization)                 )

 leaky_re_lu_28 (LeakyReLU)    (None, 128, 128, 64  0          ['batch_normalization_2
8[0][0]']
                               )

 concatenate_4 (Concatenate)   (None, 128, 128, 12  0          ['leaky_re_lu_28[0]
[0]',
                               8)                               'Conv4[0][0]']

 Conv11 (Conv2D)               (None, 128, 128, 64  73792       ['concatenate_4[0][0]']
                               )

 batch_normalization_29 (BatchN  (None, 128, 128, 64  256        ['Conv11[0][0]']
 ormalization)                 )

 leaky_re_lu_29 (LeakyReLU)    (None, 128, 128, 64  0          ['batch_normalization_2
9[0][0]']
                               )

 Conv12 (Conv2D)               (None, 128, 128, 64  36928       ['leaky_re_lu_29[0]
[0]']
                               )

 batch_normalization_30 (BatchN  (None, 128, 128, 64  256        ['Conv12[0][0]']
 ormalization)                 )

 leaky_re_lu_30 (LeakyReLU)    (None, 128, 128, 64  0          ['batch_normalization_3
0[0][0]']
                               )

 conv2d_transpose_5 (Conv2DTran  (None, 256, 256, 32  18464      ['leaky_re_lu_30[0]
[0]']
 spose)                        )

 batch_normalization_31 (BatchN  (None, 256, 256, 32  128        ['conv2d_transpose_5[0]
```

```
[0]']
 ormalization)                  )

 leaky_re_lu_31 (LeakyReLU)     (None, 256, 256, 32  0           ['batch_normalization_3
1[0][0]']
                                )

 concatenate_5 (Concatenate)    (None, 256, 256, 64  0           ['leaky_re_lu_31[0]
[0]',
                                )                                 'Conv2[0][0]']

 Conv25 (Conv2D)                (None, 256, 256, 32  18464       ['concatenate_5[0][0]']
                                )

 batch_normalization_32 (BatchN  (None, 256, 256, 32  128        ['Conv25[0][0]']
 ormalization)                  )

 leaky_re_lu_32 (LeakyReLU)     (None, 256, 256, 32  0           ['batch_normalization_3
2[0][0]']
                                )

 Conv26 (Conv2D)                (None, 256, 256, 32  9248        ['leaky_re_lu_32[0]
[0]']
                                )

 batch_normalization_33 (BatchN  (None, 256, 256, 32  128        ['Conv26[0][0]']
 ormalization)                  )

 leaky_re_lu_33 (LeakyReLU)     (None, 256, 256, 32  0           ['batch_normalization_3
3[0][0]']
                                )

 Outputs (Conv2D)               (None, 256, 256, 10  330         ['leaky_re_lu_33[0]
[0]']
                                )

==================================================================================
==========
Total params: 2,147,466
Trainable params: 2,144,202
Non-trainable params: 3,264
__________________________________________________________________________________
_________
```

In [ ]:
```
img_file = './model_arch.png'

tf.keras.utils.plot_model(mymodel, to_file=img_file, show_shapes=True, show_layer_names
```

```
('Failed to import pydot. You must `pip install pydot` and install graphviz (https://gra
phviz.gitlab.io/download/), ', 'for `pydotprint` to work.')
```

In [47]:
```
del inputs
```

In [48]:
```
def split_data(x,y,test_size=0.2):
    x1=np.array(x)
    del x
    y1=np.array(y)
    del y
```

```
    x_train, x_test, y_train, y_test = train_test_split(x1, y1, test_size=test_size)
    return  x_train, x_test, y_train, y_test
```

In [49]:
```
x_train, x_test, y_train, y_test=split_data(rescaled_input[:1300],labels[:1300],test_s
```

In [50]:
```
def callbacks(patience=5):
    checkpoint = tf.keras.callbacks.ModelCheckpoint('seg_model.h5', monitor='loss', ver
    early=tf.keras.callbacks.EarlyStopping(monitor='loss',patience=patience)
    callbacks_list=[checkpoint, early]
    return callbacks_list
```

In [51]:
```
mymodel.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),loss=tf.keras.lo
hist=mymodel.fit(x_train,y_train,batch_size=16,epochs=200,callbacks=callbacks())
```

```
Epoch 1/200
65/65 [==============================] - ETA: 0s - loss: 1.1677 - acc: 0.6240
Epoch 00001: loss improved from inf to 1.16775, saving model to seg_model.h5
65/65 [==============================] - 46s 674ms/step - loss: 1.1677 - acc: 0.6240
Epoch 2/200
65/65 [==============================] - ETA: 0s - loss: 0.9437 - acc: 0.6896
Epoch 00002: loss improved from 1.16775 to 0.94367, saving model to seg_model.h5
65/65 [==============================] - 44s 672ms/step - loss: 0.9437 - acc: 0.6896
Epoch 3/200
65/65 [==============================] - ETA: 0s - loss: 0.8458 - acc: 0.7169
Epoch 00003: loss improved from 0.94367 to 0.84578, saving model to seg_model.h5
65/65 [==============================] - 44s 673ms/step - loss: 0.8458 - acc: 0.7169
Epoch 4/200
65/65 [==============================] - ETA: 0s - loss: 0.7845 - acc: 0.7372
Epoch 00004: loss improved from 0.84578 to 0.78446, saving model to seg_model.h5
65/65 [==============================] - 44s 674ms/step - loss: 0.7845 - acc: 0.7372
Epoch 5/200
65/65 [==============================] - ETA: 0s - loss: 0.7567 - acc: 0.7453
Epoch 00005: loss improved from 0.78446 to 0.75665, saving model to seg_model.h5
65/65 [==============================] - 44s 673ms/step - loss: 0.7567 - acc: 0.7453
Epoch 6/200
65/65 [==============================] - ETA: 0s - loss: 0.7119 - acc: 0.7608
Epoch 00006: loss improved from 0.75665 to 0.71187, saving model to seg_model.h5
65/65 [==============================] - 44s 674ms/step - loss: 0.7119 - acc: 0.7608
Epoch 7/200
65/65 [==============================] - ETA: 0s - loss: 0.6921 - acc: 0.7676
Epoch 00007: loss improved from 0.71187 to 0.69205, saving model to seg_model.h5
65/65 [==============================] - 44s 673ms/step - loss: 0.6921 - acc: 0.7676
Epoch 8/200
65/65 [==============================] - ETA: 0s - loss: 0.6685 - acc: 0.7763
Epoch 00008: loss improved from 0.69205 to 0.66852, saving model to seg_model.h5
65/65 [==============================] - 44s 677ms/step - loss: 0.6685 - acc: 0.7763
Epoch 9/200
65/65 [==============================] - ETA: 0s - loss: 0.6522 - acc: 0.7813
Epoch 00009: loss improved from 0.66852 to 0.65219, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.6522 - acc: 0.7813
Epoch 10/200
65/65 [==============================] - ETA: 0s - loss: 0.6283 - acc: 0.7899
Epoch 00010: loss improved from 0.65219 to 0.62832, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.6283 - acc: 0.7899
Epoch 11/200
65/65 [==============================] - ETA: 0s - loss: 0.6177 - acc: 0.7942
Epoch 00011: loss improved from 0.62832 to 0.61772, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.6177 - acc: 0.7942
Epoch 12/200
```

```
65/65 [==============================] - ETA: 0s - loss: 0.6093 - acc: 0.7967
Epoch 00012: loss improved from 0.61772 to 0.60933, saving model to seg_model.h5
65/65 [==============================] - 44s 678ms/step - loss: 0.6093 - acc: 0.7967
Epoch 13/200
65/65 [==============================] - ETA: 0s - loss: 0.5943 - acc: 0.8027
Epoch 00013: loss improved from 0.60933 to 0.59427, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5943 - acc: 0.8027
Epoch 14/200
65/65 [==============================] - ETA: 0s - loss: 0.5786 - acc: 0.8079
Epoch 00014: loss improved from 0.59427 to 0.57863, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5786 - acc: 0.8079
Epoch 15/200
65/65 [==============================] - ETA: 0s - loss: 0.5787 - acc: 0.8071
Epoch 00015: loss did not improve from 0.57863
65/65 [==============================] - 44s 677ms/step - loss: 0.5787 - acc: 0.8071
Epoch 16/200
65/65 [==============================] - ETA: 0s - loss: 0.5723 - acc: 0.8088
Epoch 00016: loss improved from 0.57863 to 0.57225, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5723 - acc: 0.8088
Epoch 17/200
65/65 [==============================] - ETA: 0s - loss: 0.5516 - acc: 0.8167
Epoch 00017: loss improved from 0.57225 to 0.55157, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.5516 - acc: 0.8167
Epoch 18/200
65/65 [==============================] - ETA: 0s - loss: 0.5485 - acc: 0.8167
Epoch 00018: loss improved from 0.55157 to 0.54847, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5485 - acc: 0.8167
Epoch 19/200
65/65 [==============================] - ETA: 0s - loss: 0.5526 - acc: 0.8159
Epoch 00019: loss did not improve from 0.54847
65/65 [==============================] - 44s 677ms/step - loss: 0.5526 - acc: 0.8159
Epoch 20/200
65/65 [==============================] - ETA: 0s - loss: 0.5395 - acc: 0.8210
Epoch 00020: loss improved from 0.54847 to 0.53954, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5395 - acc: 0.8210
Epoch 21/200
65/65 [==============================] - ETA: 0s - loss: 0.5321 - acc: 0.8227
Epoch 00021: loss improved from 0.53954 to 0.53210, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5321 - acc: 0.8227
Epoch 22/200
65/65 [==============================] - ETA: 0s - loss: 0.5221 - acc: 0.8261
Epoch 00022: loss improved from 0.53210 to 0.52209, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5221 - acc: 0.8261
Epoch 23/200
65/65 [==============================] - ETA: 0s - loss: 0.5169 - acc: 0.8281
Epoch 00023: loss improved from 0.52209 to 0.51691, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5169 - acc: 0.8281
Epoch 24/200
65/65 [==============================] - ETA: 0s - loss: 0.5107 - acc: 0.8305
Epoch 00024: loss improved from 0.51691 to 0.51066, saving model to seg_model.h5
65/65 [==============================] - 44s 678ms/step - loss: 0.5107 - acc: 0.8305
Epoch 25/200
65/65 [==============================] - ETA: 0s - loss: 0.5078 - acc: 0.8319
Epoch 00025: loss improved from 0.51066 to 0.50784, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.5078 - acc: 0.8319
Epoch 26/200
65/65 [==============================] - ETA: 0s - loss: 0.5095 - acc: 0.8302
Epoch 00026: loss did not improve from 0.50784
65/65 [==============================] - 44s 676ms/step - loss: 0.5095 - acc: 0.8302
Epoch 27/200
```

```
65/65 [==============================] - ETA: 0s - loss: 0.4996 - acc: 0.8342
Epoch 00027: loss improved from 0.50784 to 0.49959, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4996 - acc: 0.8342
Epoch 28/200
65/65 [==============================] - ETA: 0s - loss: 0.4870 - acc: 0.8382
Epoch 00028: loss improved from 0.49959 to 0.48703, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4870 - acc: 0.8382
Epoch 29/200
65/65 [==============================] - ETA: 0s - loss: 0.4792 - acc: 0.8402
Epoch 00029: loss improved from 0.48703 to 0.47921, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4792 - acc: 0.8402
Epoch 30/200
65/65 [==============================] - ETA: 0s - loss: 0.4797 - acc: 0.8393
Epoch 00030: loss did not improve from 0.47921
65/65 [==============================] - 44s 677ms/step - loss: 0.4797 - acc: 0.8393
Epoch 31/200
65/65 [==============================] - ETA: 0s - loss: 0.4840 - acc: 0.8385
Epoch 00031: loss did not improve from 0.47921
65/65 [==============================] - 44s 677ms/step - loss: 0.4840 - acc: 0.8385
Epoch 32/200
65/65 [==============================] - ETA: 0s - loss: 0.4715 - acc: 0.8419
Epoch 00032: loss improved from 0.47921 to 0.47148, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4715 - acc: 0.8419
Epoch 33/200
65/65 [==============================] - ETA: 0s - loss: 0.4672 - acc: 0.8435
Epoch 00033: loss improved from 0.47148 to 0.46719, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4672 - acc: 0.8435
Epoch 34/200
65/65 [==============================] - ETA: 0s - loss: 0.4699 - acc: 0.8432
Epoch 00034: loss did not improve from 0.46719
65/65 [==============================] - 44s 677ms/step - loss: 0.4699 - acc: 0.8432
Epoch 35/200
65/65 [==============================] - ETA: 0s - loss: 0.4636 - acc: 0.8454
Epoch 00035: loss improved from 0.46719 to 0.46363, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.4636 - acc: 0.8454
Epoch 36/200
65/65 [==============================] - ETA: 0s - loss: 0.4627 - acc: 0.8447
Epoch 00036: loss improved from 0.46363 to 0.46267, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4627 - acc: 0.8447
Epoch 37/200
65/65 [==============================] - ETA: 0s - loss: 0.4539 - acc: 0.8474
Epoch 00037: loss improved from 0.46267 to 0.45387, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4539 - acc: 0.8474
Epoch 38/200
65/65 [==============================] - ETA: 0s - loss: 0.4483 - acc: 0.8493
Epoch 00038: loss improved from 0.45387 to 0.44832, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4483 - acc: 0.8493
Epoch 39/200
65/65 [==============================] - ETA: 0s - loss: 0.4390 - acc: 0.8530
Epoch 00039: loss improved from 0.44832 to 0.43905, saving model to seg_model.h5
65/65 [==============================] - 44s 678ms/step - loss: 0.4390 - acc: 0.8530
Epoch 40/200
65/65 [==============================] - ETA: 0s - loss: 0.4340 - acc: 0.8539
Epoch 00040: loss improved from 0.43905 to 0.43397, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4340 - acc: 0.8539
Epoch 41/200
65/65 [==============================] - ETA: 0s - loss: 0.4442 - acc: 0.8508
Epoch 00041: loss did not improve from 0.43397
65/65 [==============================] - 44s 677ms/step - loss: 0.4442 - acc: 0.8508
Epoch 42/200
```

```
65/65 [==============================] - ETA: 0s - loss: 0.4293 - acc: 0.8550
Epoch 00042: loss improved from 0.43397 to 0.42935, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4293 - acc: 0.8550
Epoch 43/200
65/65 [==============================] - ETA: 0s - loss: 0.4313 - acc: 0.8543
Epoch 00043: loss did not improve from 0.42935
65/65 [==============================] - 44s 677ms/step - loss: 0.4313 - acc: 0.8543
Epoch 44/200
65/65 [==============================] - ETA: 0s - loss: 0.4322 - acc: 0.8548
Epoch 00044: loss did not improve from 0.42935
65/65 [==============================] - 44s 677ms/step - loss: 0.4322 - acc: 0.8548
Epoch 45/200
65/65 [==============================] - ETA: 0s - loss: 0.4210 - acc: 0.8580
Epoch 00045: loss improved from 0.42935 to 0.42099, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.4210 - acc: 0.8580
Epoch 46/200
65/65 [==============================] - ETA: 0s - loss: 0.4174 - acc: 0.8591
Epoch 00046: loss improved from 0.42099 to 0.41741, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.4174 - acc: 0.8591
Epoch 47/200
65/65 [==============================] - ETA: 0s - loss: 0.4242 - acc: 0.8568
Epoch 00047: loss did not improve from 0.41741
65/65 [==============================] - 44s 677ms/step - loss: 0.4242 - acc: 0.8568
Epoch 48/200
65/65 [==============================] - ETA: 0s - loss: 0.4167 - acc: 0.8586
Epoch 00048: loss improved from 0.41741 to 0.41672, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.4167 - acc: 0.8586
Epoch 49/200
65/65 [==============================] - ETA: 0s - loss: 0.4090 - acc: 0.8617
Epoch 00049: loss improved from 0.41672 to 0.40897, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.4090 - acc: 0.8617
Epoch 50/200
65/65 [==============================] - ETA: 0s - loss: 0.3977 - acc: 0.8648
Epoch 00050: loss improved from 0.40897 to 0.39766, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3977 - acc: 0.8648
Epoch 51/200
65/65 [==============================] - ETA: 0s - loss: 0.3961 - acc: 0.8652
Epoch 00051: loss improved from 0.39766 to 0.39610, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3961 - acc: 0.8652
Epoch 52/200
65/65 [==============================] - ETA: 0s - loss: 0.3942 - acc: 0.8660
Epoch 00052: loss improved from 0.39610 to 0.39418, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3942 - acc: 0.8660
Epoch 53/200
65/65 [==============================] - ETA: 0s - loss: 0.3934 - acc: 0.8658
Epoch 00053: loss improved from 0.39418 to 0.39342, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3934 - acc: 0.8658
Epoch 54/200
65/65 [==============================] - ETA: 0s - loss: 0.3943 - acc: 0.8656
Epoch 00054: loss did not improve from 0.39342
65/65 [==============================] - 44s 678ms/step - loss: 0.3943 - acc: 0.8656
Epoch 55/200
65/65 [==============================] - ETA: 0s - loss: 0.3790 - acc: 0.8703
Epoch 00055: loss improved from 0.39342 to 0.37898, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3790 - acc: 0.8703
Epoch 56/200
65/65 [==============================] - ETA: 0s - loss: 0.3771 - acc: 0.8713
Epoch 00056: loss improved from 0.37898 to 0.37713, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3771 - acc: 0.8713
Epoch 57/200
```

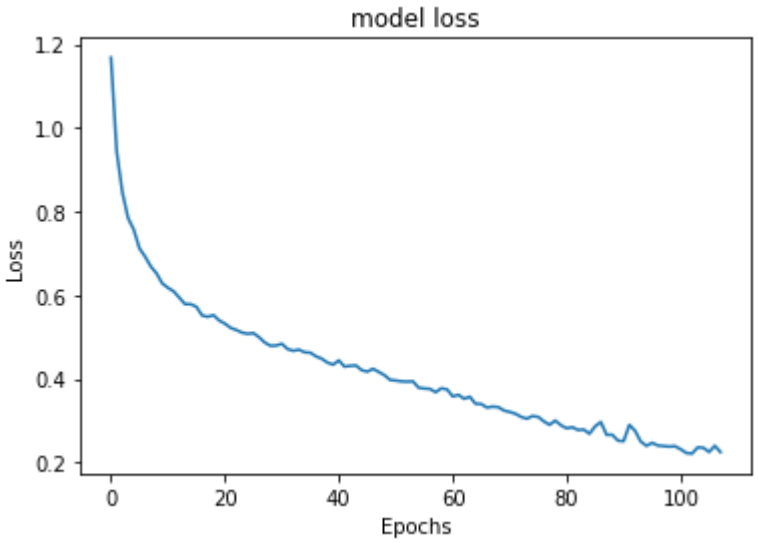
```
65/65 [==============================] - ETA: 0s - loss: 0.3762 - acc: 0.8713
Epoch 00057: loss improved from 0.37713 to 0.37617, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3762 - acc: 0.8713
Epoch 58/200
65/65 [==============================] - ETA: 0s - loss: 0.3683 - acc: 0.8738
Epoch 00058: loss improved from 0.37617 to 0.36833, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3683 - acc: 0.8738
Epoch 59/200
65/65 [==============================] - ETA: 0s - loss: 0.3775 - acc: 0.8709
Epoch 00059: loss did not improve from 0.36833
65/65 [==============================] - 44s 677ms/step - loss: 0.3775 - acc: 0.8709
Epoch 60/200
65/65 [==============================] - ETA: 0s - loss: 0.3749 - acc: 0.8712
Epoch 00060: loss did not improve from 0.36833
65/65 [==============================] - 44s 677ms/step - loss: 0.3749 - acc: 0.8712
Epoch 61/200
65/65 [==============================] - ETA: 0s - loss: 0.3581 - acc: 0.8773
Epoch 00061: loss improved from 0.36833 to 0.35810, saving model to seg_model.h5
65/65 [==============================] - 44s 681ms/step - loss: 0.3581 - acc: 0.8773
Epoch 62/200
65/65 [==============================] - ETA: 0s - loss: 0.3620 - acc: 0.8755
Epoch 00062: loss did not improve from 0.35810
65/65 [==============================] - 44s 677ms/step - loss: 0.3620 - acc: 0.8755
Epoch 63/200
65/65 [==============================] - ETA: 0s - loss: 0.3525 - acc: 0.8789
Epoch 00063: loss improved from 0.35810 to 0.35247, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3525 - acc: 0.8789
Epoch 64/200
65/65 [==============================] - ETA: 0s - loss: 0.3574 - acc: 0.8767
Epoch 00064: loss did not improve from 0.35247
65/65 [==============================] - 44s 678ms/step - loss: 0.3574 - acc: 0.8767
Epoch 65/200
65/65 [==============================] - ETA: 0s - loss: 0.3406 - acc: 0.8833
Epoch 00065: loss improved from 0.35247 to 0.34064, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3406 - acc: 0.8833
Epoch 66/200
65/65 [==============================] - ETA: 0s - loss: 0.3401 - acc: 0.8833
Epoch 00066: loss improved from 0.34064 to 0.34008, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3401 - acc: 0.8833
Epoch 67/200
65/65 [==============================] - ETA: 0s - loss: 0.3315 - acc: 0.8865
Epoch 00067: loss improved from 0.34008 to 0.33154, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3315 - acc: 0.8865
Epoch 68/200
65/65 [==============================] - ETA: 0s - loss: 0.3339 - acc: 0.8849
Epoch 00068: loss did not improve from 0.33154
65/65 [==============================] - 44s 677ms/step - loss: 0.3339 - acc: 0.8849
Epoch 69/200
65/65 [==============================] - ETA: 0s - loss: 0.3325 - acc: 0.8850
Epoch 00069: loss did not improve from 0.33154
65/65 [==============================] - 44s 678ms/step - loss: 0.3325 - acc: 0.8850
Epoch 70/200
65/65 [==============================] - ETA: 0s - loss: 0.3244 - acc: 0.8884
Epoch 00070: loss improved from 0.33154 to 0.32441, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3244 - acc: 0.8884
Epoch 71/200
65/65 [==============================] - ETA: 0s - loss: 0.3211 - acc: 0.8895
Epoch 00071: loss improved from 0.32441 to 0.32106, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.3211 - acc: 0.8895
Epoch 72/200
```

```
65/65 [==============================] - ETA: 0s - loss: 0.3168 - acc: 0.8910
Epoch 00072: loss improved from 0.32106 to 0.31678, saving model to seg_model.h5
65/65 [==============================] - 44s 681ms/step - loss: 0.3168 - acc: 0.8910
Epoch 73/200
65/65 [==============================] - ETA: 0s - loss: 0.3093 - acc: 0.8935
Epoch 00073: loss improved from 0.31678 to 0.30926, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.3093 - acc: 0.8935
Epoch 74/200
65/65 [==============================] - ETA: 0s - loss: 0.3049 - acc: 0.8951
Epoch 00074: loss improved from 0.30926 to 0.30487, saving model to seg_model.h5
65/65 [==============================] - 44s 682ms/step - loss: 0.3049 - acc: 0.8951
Epoch 75/200
65/65 [==============================] - ETA: 0s - loss: 0.3113 - acc: 0.8927
Epoch 00075: loss did not improve from 0.30487
65/65 [==============================] - 44s 679ms/step - loss: 0.3113 - acc: 0.8927
Epoch 76/200
65/65 [==============================] - ETA: 0s - loss: 0.3092 - acc: 0.8933
Epoch 00076: loss did not improve from 0.30487
65/65 [==============================] - 44s 676ms/step - loss: 0.3092 - acc: 0.8933
Epoch 77/200
65/65 [==============================] - ETA: 0s - loss: 0.2985 - acc: 0.8973
Epoch 00077: loss improved from 0.30487 to 0.29853, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.2985 - acc: 0.8973
Epoch 78/200
65/65 [==============================] - ETA: 0s - loss: 0.2904 - acc: 0.9002
Epoch 00078: loss improved from 0.29853 to 0.29042, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2904 - acc: 0.9002
Epoch 79/200
65/65 [==============================] - ETA: 0s - loss: 0.3005 - acc: 0.8960
Epoch 00079: loss did not improve from 0.29042
65/65 [==============================] - 44s 677ms/step - loss: 0.3005 - acc: 0.8960
Epoch 80/200
65/65 [==============================] - ETA: 0s - loss: 0.2891 - acc: 0.9004
Epoch 00080: loss improved from 0.29042 to 0.28912, saving model to seg_model.h5
65/65 [==============================] - 44s 681ms/step - loss: 0.2891 - acc: 0.9004
Epoch 81/200
65/65 [==============================] - ETA: 0s - loss: 0.2821 - acc: 0.9026
Epoch 00081: loss improved from 0.28912 to 0.28206, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2821 - acc: 0.9026
Epoch 82/200
65/65 [==============================] - ETA: 0s - loss: 0.2845 - acc: 0.9016
Epoch 00082: loss did not improve from 0.28206
65/65 [==============================] - 44s 678ms/step - loss: 0.2845 - acc: 0.9016
Epoch 83/200
65/65 [==============================] - ETA: 0s - loss: 0.2779 - acc: 0.9039
Epoch 00083: loss improved from 0.28206 to 0.27793, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2779 - acc: 0.9039
Epoch 84/200
65/65 [==============================] - ETA: 0s - loss: 0.2790 - acc: 0.9035
Epoch 00084: loss did not improve from 0.27793
65/65 [==============================] - 44s 678ms/step - loss: 0.2790 - acc: 0.9035
Epoch 85/200
65/65 [==============================] - ETA: 0s - loss: 0.2689 - acc: 0.9067
Epoch 00085: loss improved from 0.27793 to 0.26894, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2689 - acc: 0.9067
Epoch 86/200
65/65 [==============================] - ETA: 0s - loss: 0.2867 - acc: 0.9006
Epoch 00086: loss did not improve from 0.26894
65/65 [==============================] - 44s 678ms/step - loss: 0.2867 - acc: 0.9006
Epoch 87/200
```

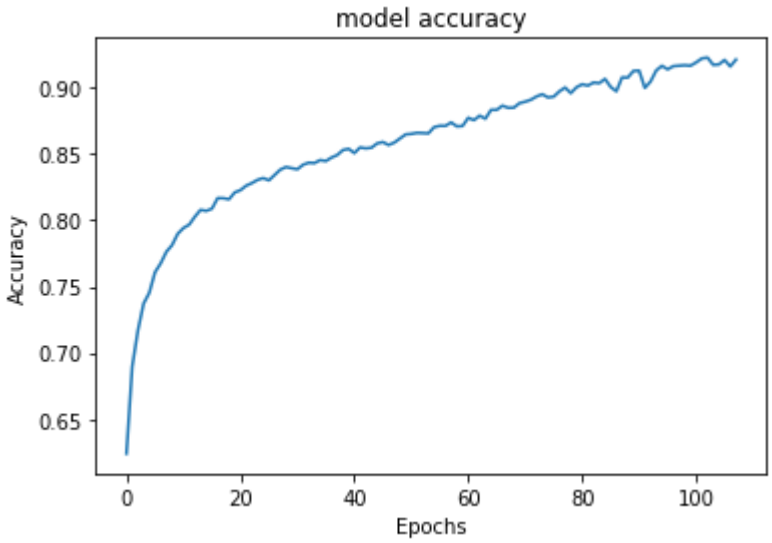
```
65/65 [==============================] - ETA: 0s - loss: 0.2970 - acc: 0.8973
Epoch 00087: loss did not improve from 0.26894
65/65 [==============================] - 44s 678ms/step - loss: 0.2970 - acc: 0.8973
Epoch 88/200
65/65 [==============================] - ETA: 0s - loss: 0.2665 - acc: 0.9078
Epoch 00088: loss improved from 0.26894 to 0.26647, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2665 - acc: 0.9078
Epoch 89/200
65/65 [==============================] - ETA: 0s - loss: 0.2666 - acc: 0.9077
Epoch 00089: loss did not improve from 0.26647
65/65 [==============================] - 44s 677ms/step - loss: 0.2666 - acc: 0.9077
Epoch 90/200
65/65 [==============================] - ETA: 0s - loss: 0.2526 - acc: 0.9127
Epoch 00090: loss improved from 0.26647 to 0.25255, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2526 - acc: 0.9127
Epoch 91/200
65/65 [==============================] - ETA: 0s - loss: 0.2512 - acc: 0.9129
Epoch 00091: loss improved from 0.25255 to 0.25125, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2512 - acc: 0.9129
Epoch 92/200
65/65 [==============================] - ETA: 0s - loss: 0.2902 - acc: 0.8999
Epoch 00092: loss did not improve from 0.25125
65/65 [==============================] - 44s 678ms/step - loss: 0.2902 - acc: 0.8999
Epoch 93/200
65/65 [==============================] - ETA: 0s - loss: 0.2762 - acc: 0.9047
Epoch 00093: loss did not improve from 0.25125
65/65 [==============================] - 44s 677ms/step - loss: 0.2762 - acc: 0.9047
Epoch 94/200
65/65 [==============================] - ETA: 0s - loss: 0.2504 - acc: 0.9131
Epoch 00094: loss improved from 0.25125 to 0.25037, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2504 - acc: 0.9131
Epoch 95/200
65/65 [==============================] - ETA: 0s - loss: 0.2401 - acc: 0.9166
Epoch 00095: loss improved from 0.25037 to 0.24014, saving model to seg_model.h5
65/65 [==============================] - 44s 679ms/step - loss: 0.2401 - acc: 0.9166
Epoch 96/200
65/65 [==============================] - ETA: 0s - loss: 0.2473 - acc: 0.9139
Epoch 00096: loss did not improve from 0.24014
65/65 [==============================] - 44s 677ms/step - loss: 0.2473 - acc: 0.9139
Epoch 97/200
65/65 [==============================] - ETA: 0s - loss: 0.2408 - acc: 0.9163
Epoch 00097: loss did not improve from 0.24014
65/65 [==============================] - 44s 677ms/step - loss: 0.2408 - acc: 0.9163
Epoch 98/200
65/65 [==============================] - ETA: 0s - loss: 0.2396 - acc: 0.9167
Epoch 00098: loss improved from 0.24014 to 0.23961, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2396 - acc: 0.9167
Epoch 99/200
65/65 [==============================] - ETA: 0s - loss: 0.2383 - acc: 0.9170
Epoch 00099: loss improved from 0.23961 to 0.23828, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2383 - acc: 0.9170
Epoch 100/200
65/65 [==============================] - ETA: 0s - loss: 0.2393 - acc: 0.9166
Epoch 00100: loss did not improve from 0.23828
65/65 [==============================] - 44s 679ms/step - loss: 0.2393 - acc: 0.9166
Epoch 101/200
65/65 [==============================] - ETA: 0s - loss: 0.2321 - acc: 0.9191
Epoch 00101: loss improved from 0.23828 to 0.23209, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2321 - acc: 0.9191
Epoch 102/200
```

```
65/65 [==============================] - ETA: 0s - loss: 0.2231 - acc: 0.9221
Epoch 00102: loss improved from 0.23209 to 0.22307, saving model to seg_model.h5
65/65 [==============================] - 44s 680ms/step - loss: 0.2231 - acc: 0.9221
Epoch 103/200
65/65 [==============================] - ETA: 0s - loss: 0.2214 - acc: 0.9228
Epoch 00103: loss improved from 0.22307 to 0.22144, saving model to seg_model.h5
65/65 [==============================] - 44s 681ms/step - loss: 0.2214 - acc: 0.9228
Epoch 104/200
65/65 [==============================] - ETA: 0s - loss: 0.2364 - acc: 0.9173
Epoch 00104: loss did not improve from 0.22144
65/65 [==============================] - 44s 677ms/step - loss: 0.2364 - acc: 0.9173
Epoch 105/200
65/65 [==============================] - ETA: 0s - loss: 0.2354 - acc: 0.9176
Epoch 00105: loss did not improve from 0.22144
65/65 [==============================] - 44s 678ms/step - loss: 0.2354 - acc: 0.9176
Epoch 106/200
65/65 [==============================] - ETA: 0s - loss: 0.2254 - acc: 0.9210
Epoch 00106: loss did not improve from 0.22144
65/65 [==============================] - 44s 678ms/step - loss: 0.2254 - acc: 0.9210
Epoch 107/200
65/65 [==============================] - ETA: 0s - loss: 0.2402 - acc: 0.9161
Epoch 00107: loss did not improve from 0.22144
65/65 [==============================] - 44s 678ms/step - loss: 0.2402 - acc: 0.9161
Epoch 108/200
65/65 [==============================] - ETA: 0s - loss: 0.2253 - acc: 0.9213
Epoch 00108: loss did not improve from 0.22144
65/65 [==============================] - 44s 678ms/step - loss: 0.2253 - acc: 0.9213
```

In [52]:
```
plt.plot(hist.history['loss'])
plt.title("model loss")
plt.ylabel("Loss")
plt.xlabel("Epochs")
plt.show()
```



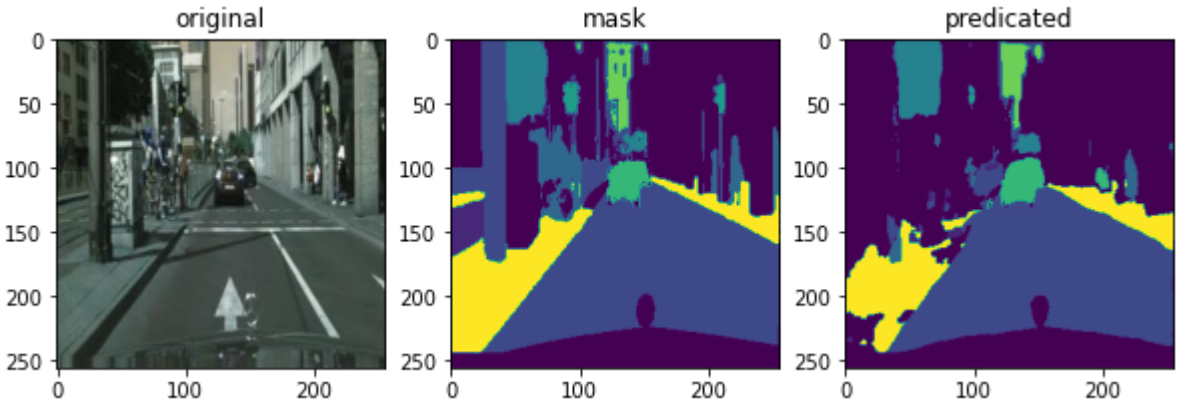
In [53]:
```
plt.plot(hist.history["acc"])
plt.title("model accuracy")
plt.ylabel("Accuracy")
plt.xlabel("Epochs")
plt.show()
```

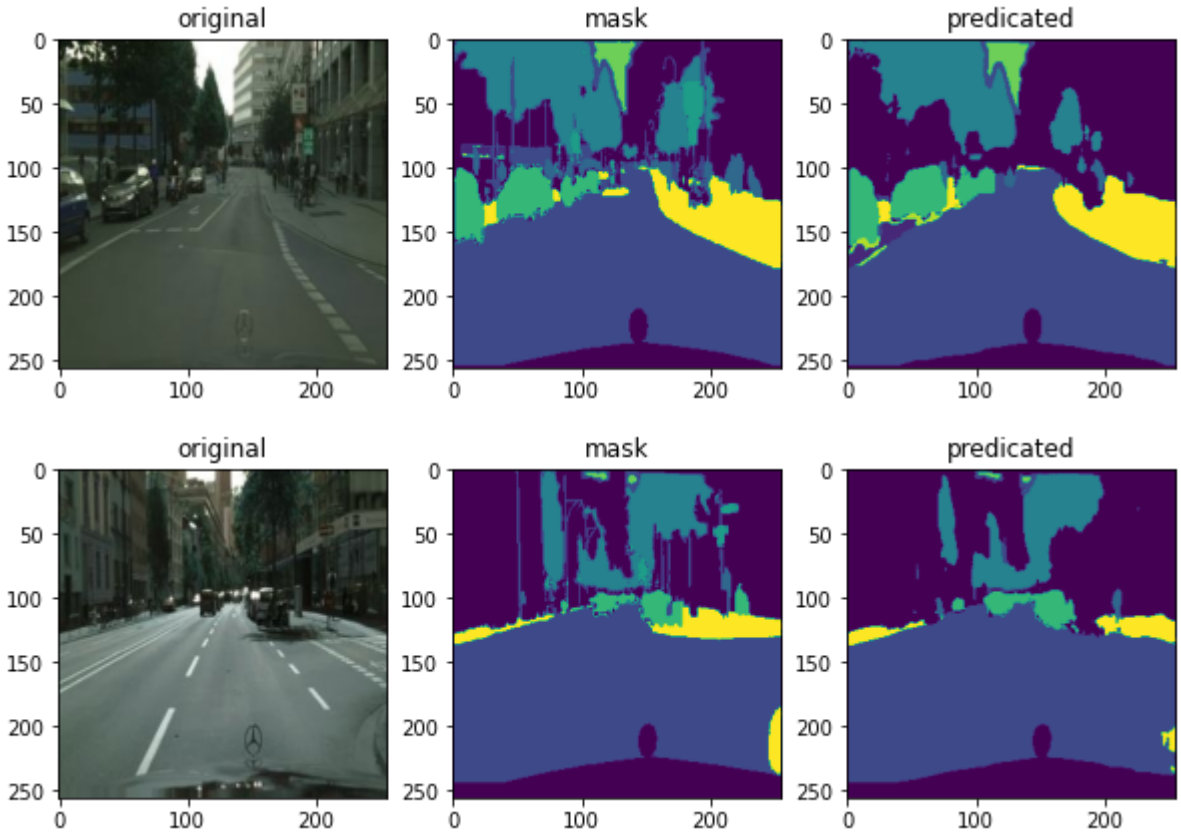


In [54]:
```
pred=mymodel.predict(x_test)
y_pred=tf.argmax(pred,axis=-1)
```

In [55]:
```
def show_predications(x_test,y_test,y_pred):
    idx=np.random.randint(0,len(y_pred))
    fig, axes = plt.subplots(1,3,figsize=(10,10))
    axes[0].imshow(x_test[idx])
    axes[0].set_title("original")
    axes[1].imshow(y_test[idx])
    axes[1].set_title("mask")
    axes[2].imshow(y_pred[idx])
    axes[2].set_title("predicated")
```

In [56]:
```
for i in range(3):
    show_predications(x_test,y_test,y_pred)
```





In [58]:
```
import pandas as pd

df = pd.DataFrame(hist.history)
df
```

Out[58]:

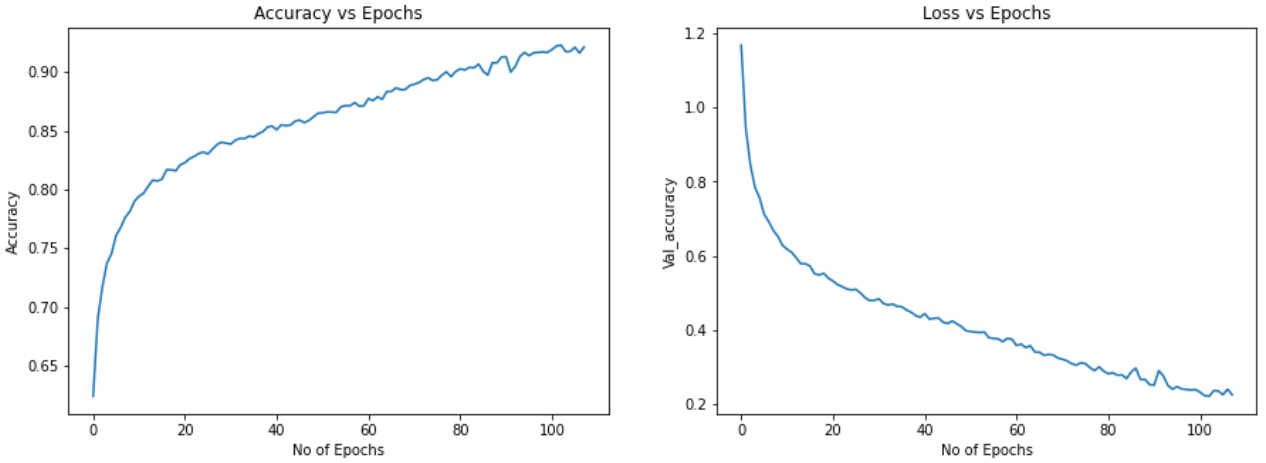
	loss	acc
0	1.167745	0.624000
1	0.943668	0.689557
2	0.845775	0.716860
3	0.784463	0.737166
4	0.756650	0.745333
...	...	...
103	0.236433	0.917284
104	0.235418	0.917608
105	0.225444	0.920963
106	0.240224	0.916070
107	0.225275	0.921252

108 rows × 2 columns

In [64]:
```
import matplotlib.pyplot as plt
fig, axes = plt.subplots(1,2, figsize=(15, 5))
```

```
axes[0].plot(df.acc)
axes[0].set_xlabel('No of Epochs')
axes[0].set_ylabel('Accuracy')
axes[0].set_title('Accuracy vs Epochs')

axes[1].plot(df.loss)
axes[1].set_xlabel('No of Epochs')
axes[1].set_ylabel('Val_accuracy')
axes[1].set_title('Loss vs Epochs')
plt.show()
```



In [78]:

In [79]:

```
---------------------------------------------------------------------------
InvalidArgumentError                      Traceback (most recent call last)
<ipython-input-79-6a1ed4a6fc89> in <module>()
----> 1 print(iou_coef(y_test,y_pred))

<ipython-input-78-007ee8784880> in iou_coef(y_true, y_pred, smooth)
      1 from keras import backend as K
      2 def iou_coef(y_true, y_pred, smooth=1):
----> 3   intersection = K.sum(K.abs(y_true * y_pred), axis=[1,2,3])
      4   union = K.sum(y_true)+K.sum(y_pred)-intersection
      5   iou = K.mean((intersection + smooth) / (union + smooth), axis=3)

/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/traceback_utils.py in erro
r_handler(*args, **kwargs)
    151     except Exception as e:
    152       filtered_tb = _process_traceback_frames(e.__traceback__)
--> 153       raise e.with_traceback(filtered_tb) from None
    154     finally:
    155       del filtered_tb

/usr/local/lib/python3.7/dist-packages/keras/backend.py in sum(x, axis, keepdims)
   2453       A tensor with sum of `x`.
   2454   """
-> 2455   return tf.reduce_sum(x, axis, keepdims)
   2456
   2457
```

```
InvalidArgumentError: Invalid reduction dimension (3 for input with 3 dimension(s) [Op:Sum]
```