Instance Level Salient Object Segmentation

Project Mid Evaluation

Team Name : DeepView

Members :

- Aniket Joshi (20161166)
- Prakyath Madadi (20161236)
- Vashist Madiraju (20161222)

Github Project Link:

https://github.com/prakyath-04/Instance-Level-Salient-Object-Segmentation

Introduction :-

Salient object detection attempts to locate the most noticeable and eye-attracting object regions in images. It is a fundamental problem in computer vision and has served as a pre-processing step to facilitate a wide range of vision applications. This paper presents a salient instance segmentation method. It broadly consists of 3 steps.

1) Estimating binary saliency map.

2) Detecting salient object contours.

3) Identifying salient object instances (Salient instance generation and salient instance refinement).

We build a deep multi-scale refinement network and use it for salient region detection and salient object contour detection. Then, MCG (Multiscale combinatorial grouping) algorithm is used to find object proposals from the salient object contours. We generate a fixed number of salient object proposals on the basis of the results of salient object contour detection and further apply a subset optimization method to reduce the number of object proposals. Finally, the results from the previous three steps are integrated in a CRF model to generate the final salient instance segmentation.

Targets :-

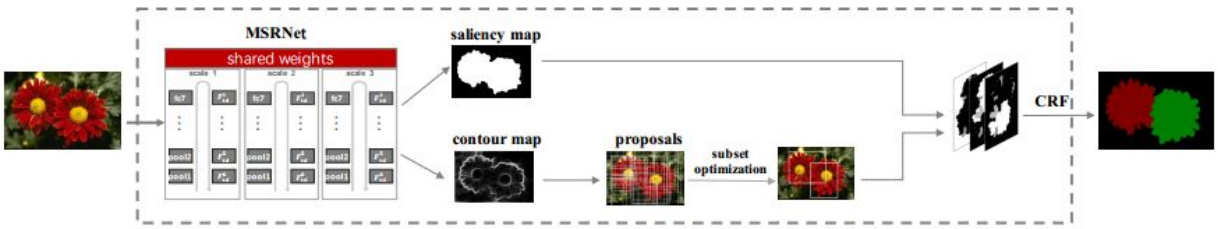
- ➔ To implement a network to generate saliency map and contour maps.
- ➔ Implement MSRnet by combining bottom up and top down using Refinement Modules ($\mathbf{R}_i$).
- ➔ Implement the attention module to compute final output of MSRNet as weighted sum of the probability maps of the inputs of different scale.

Overview :-

We formulate both salient region detection and salient object contour detection as a binary pixel labeling problem.

Since salient objects could have different scales, we propose a multiscale refinement network (MSRNet). MSRNet is composed of three refined VGG network streams with shared parameters and a learned attentional model for fusing results at different scales.

Architecture :-



Bottom-Up Network :-

- We use a modified VGG network for this with additional Convolutional layers for this.
- Takes low level features as input, such as colours and texture, and propagates it up through the layers.
- Information from an input image needs to be passed from the bottom layers up in a deep network before being transformed into high-level semantic information.

Top-Down Network :-

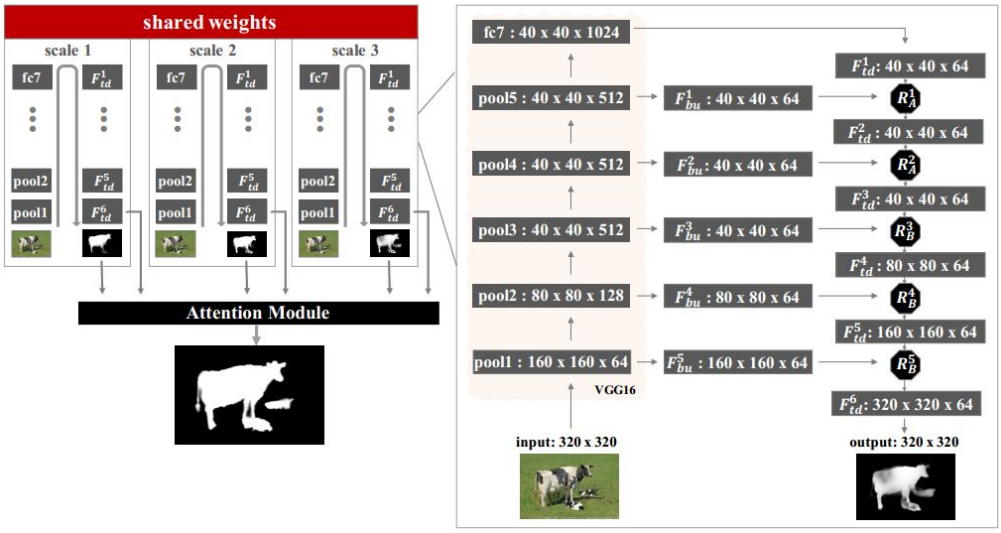
- We use a top down model to use and incorporate high level semantic information.
- It is propagated from the top layers further down, and is integrated with low-level information obtained from the intermediate stages of the Bottom-up network.
- This integration of high level information with low level features, results in high precision contour detection results.

MSRNet :-

- As we can conclude from the properties of the above networks, a network should consider both bottom-up and top-down information propagation and output a label map.
- We want the saliency map and contour size to have the same resolution as the input image. Thus the output should be a label map of same resolution as the input image.

Implementation and Results:-

To implement the MSRNet we have used Keras-Tensorflow libraries. Results of the implementation and some particular layers are given below:-



- We transform the original VGG16 into a fully convolutional network, which serves as our bottom-up backbone network.
    - We modify multiple layers of the the VGG16 network, with the resulting network having 19 layers.
    - The output resolution of the transformed VGG network is 1/8 of the original input resolution i.e **output at fc7 should be of 1/8 resolution as input image.**

Output at fc7 is 1/8 resolution as input image (320x 320).

```
max_pooling2d_5 (MaxPooling2 (None, 40, 40, 512)       0
zero_padding2d_19 (ZeroPaddi (None, 64, 64, 512)       0
conv2d_14 (Conv2D)           (None, 40, 40, 1024)      4719616
dropout_1 (Dropout)          (None, 40, 40, 1024)      0
conv2d_15 (Conv2D)           (None, 40, 40, 1024)      1049600
dropout_2 (Dropout)          (None, 40, 40, 1024)      0
```

```
scale0_pool_5 = MaxPooling2D((3,3), strides=(1,1))(scale0_layer_34)
##############scale0_pool_5 = scale0_layer_35#################

scale0_layer_36 = ZeroPadding2D((12,12))(scale0_pool_5)
scale0_layer_37 = Convolution2D(1024, (3, 3), activation='relu',dilation_rate=(12,12))(scale0_layer_36)
scale0_layer_38 = Dropout(rate = 0.5, noise_shape=None, seed=None)(scale0_layer_37)
scale0_layer_39 = Convolution2D(1024, (1, 1), activation='relu')(scale0_layer_38)
scale0_fc7 = Dropout(rate= 0.5, noise_shape=None, seed=None)(scale0_layer_39)
```

- We use refinement modules to combine the top-down and bottom-up information.
    - We integrate a “refinement module” R to invert the effect of each pooling layer and double the resolution of its input feature map if necessary.
    - Each refinement module R_i takes as input the output feature map F_it_d of the refinement module in the top-down pass before.
    - The output feature map F_ib_u of the aforementioned extra convolutional layer attached to the corresponding pooling layer in the bottom-up pass.
    - The refinement module R_i works by first concatenating F_it_d and F_ib_u and then feeding them to another 3 × 3 convolutional layer with 64 channels.
    - The final output of the refinement stream is a probability map with the same resolution as the original input image.

Output dimension is same as the input Dimension(320 x 3)

```
concatenate_6 (Concatenate)     (None, 160, 160, 128 0           dense_12[0][0]
                                                                 conv2d_transpose_2[0][0]
zero_padding2d_31 (ZeroPadding2 (None, 162, 162, 128 0           concatenate_6[0][0]
conv2d_28 (Conv2D)              (None, 160, 160, 64) 73792       zero_padding2d_31[0][0]
dense_13 (Dense)                (None, 160, 160, 64) 4160        conv2d_28[0][0]
conv2d_transpose_3 (Conv2DTrans (None, 320, 320, 64) 65600       dense_13[0][0]
```

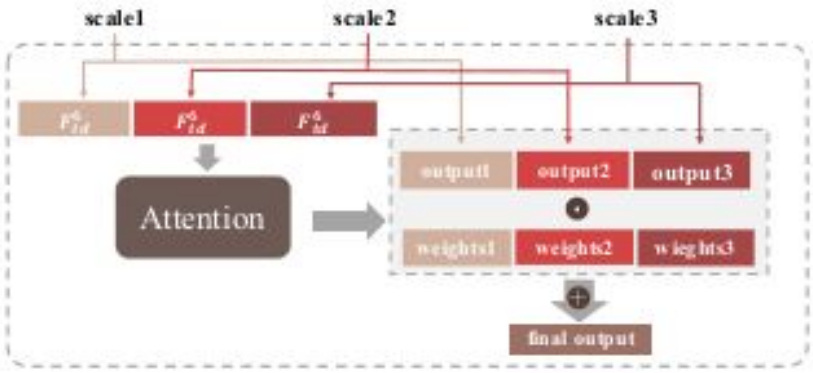
```
scale0_concat_6 = Concatenate()([scale0_skip_6, scale0_mask_6_up])
scale0_mask_7 = ZeroPadding2D((1,1))(scale0_concat_6)
scale0_mask_7 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_7)
scale0_mask_7 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_7)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
scale0_mask_7_up = Conv2DTranspose(64,(4,4),strides=2,padding = 'same',activation = 'relu')(scale0_mask_7)


model0 = Model(inputs = main_input0,outputs = scale0_mask_7_up)
print(model0.summary())
```

To Do :-

- *Attention module:*
  - The final output from our MSRNet is computed as a weighted sum of the three probability maps in a pixel-wise manner.
  - Let F_c be the fused probability map of class C and W_s be the weight map for scale S.
  - The fused map is calculated by summing the element wise multiplication between each probability map and its corresponding weight map as:

$$F_c = \Sigma_s W_s * M_c^s .$$



- To train MSRNet using the ECSSD, DUT-OMRON datasets.
- The loss functions of these two subtasks have different weights.
    - The penalty for misclassifying contour pixels is 10 times the penalty for misclassifying non-contour pixels while, for salient region detection.
    - The penalty for misclassifying salient pixels is twice the penalty for misclassifying non-salient pixels.
- Implement MCG and MAP-based optimization to generate and filter out object proposals.
    - MCG is used to generate a fixed number of salient object proposals on the basis of the results of salient object contour detection.
    - MAP-based subset optimization is used to optimize both the number and locations of detection windows given the set of salient object proposals.
- Finally use CRF based filtering to improve spatial coherence and combine the results from the previous steps to generate the final salient instance level segmentation of the input image.

Model Summary: (Scale = 1)

Layer (type)	Output Shape	Param #	Connected to
main_input (InputLayer)	(None, 320, 320, 3)	0	
zero_padding2d_1 (ZeroPadding2D	(None, 322, 322, 3)	0	main_input[0][0]
conv2d_1 (Conv2D)	(None, 320, 320, 64)	1792	zero_padding2d_1[0][0]
zero_padding2d_2 (ZeroPadding2D	(None, 322, 322, 64)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 320, 320, 64)	36928	zero_padding2d_2[0][0]
zero_padding2d_3 (ZeroPadding2D	(None, 322, 322, 64)	0	conv2d_2[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 160, 160, 64)	0	zero_padding2d_3[0][0]
zero_padding2d_4 (ZeroPadding2D	(None, 162, 162, 64)	0	max_pooling2d_1[0][0]
conv2d_3 (Conv2D)	(None, 160, 160, 128	73856	zero_padding2d_4[0][0]
zero_padding2d_5 (ZeroPadding2D	(None, 162, 162, 128	0	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 160, 160, 128	147584	zero_padding2d_5[0][0]
zero_padding2d_6 (ZeroPadding2D	(None, 162, 162, 128	0	conv2d_4[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 80, 80, 128)	0	zero_padding2d_6[0][0]
zero_padding2d_7 (ZeroPadding2D	(None, 82, 82, 128)	0	max_pooling2d_2[0][0]
conv2d_5 (Conv2D)	(None, 80, 80, 256)	295168	zero_padding2d_7[0][0]
zero_padding2d_8 (ZeroPadding2D	(None, 82, 82, 256)	0	conv2d_5[0][0]
conv2d_6 (Conv2D)	(None, 80, 80, 256)	590080	zero_padding2d_8[0][0]
zero_padding2d_9 (ZeroPadding2D	(None, 82, 82, 256)	0	conv2d_6[0][0]
conv2d_7 (Conv2D)	(None, 80, 80, 256)	590080	zero_padding2d_9[0][0]
zero_padding2d_10 (ZeroPadding2	(None, 82, 82, 256)	0	conv2d_7[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 40, 40, 256)	0	zero_padding2d_10[0][0]

```
zero_padding2d_11 (ZeroPadding2 (None, 42, 42, 256)  0           max_pooling2d_3[0][0]
conv2d_8 (Conv2D)               (None, 40, 40, 512)  1180160     zero_padding2d_11[0][0]
zero_padding2d_12 (ZeroPadding2 (None, 42, 42, 512)  0           conv2d_8[0][0]
conv2d_9 (Conv2D)               (None, 40, 40, 512)  2359808     zero_padding2d_12[0][0]
zero_padding2d_13 (ZeroPadding2 (None, 42, 42, 512)  0           conv2d_9[0][0]
conv2d_10 (Conv2D)              (None, 40, 40, 512)  2359808     zero_padding2d_13[0][0]
zero_padding2d_14 (ZeroPadding2 (None, 42, 42, 512)  0           conv2d_10[0][0]
max_pooling2d_4 (MaxPooling2D)  (None, 20, 20, 512)  0           zero_padding2d_14[0][0]
zero_padding2d_15 (ZeroPadding2 (None, 24, 24, 512)  0           max_pooling2d_4[0][0]
conv2d_11 (Conv2D)              (None, 20, 20, 512)  2359808     zero_padding2d_15[0][0]
zero_padding2d_16 (ZeroPadding2 (None, 24, 24, 512)  0           conv2d_11[0][0]
conv2d_12 (Conv2D)              (None, 20, 20, 512)  2359808     zero_padding2d_16[0][0]
zero_padding2d_17 (ZeroPadding2 (None, 24, 24, 512)  0           conv2d_12[0][0]
conv2d_13 (Conv2D)              (None, 20, 20, 512)  2359808     zero_padding2d_17[0][0]
zero_padding2d_18 (ZeroPadding2 (None, 22, 22, 512)  0           conv2d_13[0][0]
max_pooling2d_5 (MaxPooling2D)  (None, 20, 20, 512)  0           zero_padding2d_18[0][0]
zero_padding2d_19 (ZeroPadding2 (None, 44, 44, 512)  0           max_pooling2d_5[0][0]
conv2d_14 (Conv2D)              (None, 20, 20, 1024) 4719616     zero_padding2d_19[0][0]
dropout_1 (Dropout)             (None, 20, 20, 1024) 0           conv2d_14[0][0]
conv2d_15 (Conv2D)              (None, 20, 20, 1024) 1049600     dropout_1[0][0]
dropout_2 (Dropout)             (None, 20, 20, 1024) 0           conv2d_15[0][0]
zero_padding2d_20 (ZeroPadding2 (None, 22, 22, 1024) 0           dropout_2[0][0]

conv2d_17 (Conv2D)              (None, 20, 20, 64)   589888      zero_padding2d_20[0][0]
conv2d_16 (Conv2D)              (None, 20, 20, 64)   65600       dropout_2[0][0]
dense_2 (Dense)                 (None, 20, 20, 64)   4160        conv2d_17[0][0]
dense_1 (Dense)                 (None, 20, 20, 64)   4160        conv2d_16[0][0]
concatenate_1 (Concatenate)     (None, 20, 20, 128)  0           dense_2[0][0]
                                                                 dense_1[0][0]
zero_padding2d_22 (ZeroPadding2 (None, 22, 22, 512)  0           max_pooling2d_5[0][0]
zero_padding2d_21 (ZeroPadding2 (None, 22, 22, 128)  0           concatenate_1[0][0]
conv2d_19 (Conv2D)              (None, 20, 20, 64)   294976      zero_padding2d_22[0][0]
conv2d_18 (Conv2D)              (None, 20, 20, 64)   73792       zero_padding2d_21[0][0]
dense_4 (Dense)                 (None, 20, 20, 64)   4160        conv2d_19[0][0]
dense_3 (Dense)                 (None, 20, 20, 64)   4160        conv2d_18[0][0]
```

```
____________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 20, 20, 128)  0           dense_4[0][0]
                                                                 dense_3[0][0]
____________________________________________________________________________________________________
zero_padding2d_24 (ZeroPadding2 (None, 22, 22, 512)  0           max_pooling2d_4[0][0]
____________________________________________________________________________________________________
zero_padding2d_23 (ZeroPadding2 (None, 22, 22, 128)  0           concatenate_2[0][0]
____________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 20, 20, 64)   294976      zero_padding2d_24[0][0]
____________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 20, 20, 64)   73792       zero_padding2d_23[0][0]
____________________________________________________________________________________________________
dense_6 (Dense)                 (None, 20, 20, 64)   4160        conv2d_21[0][0]
____________________________________________________________________________________________________
dense_5 (Dense)                 (None, 20, 20, 64)   4160        conv2d_20[0][0]
____________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 20, 20, 128)  0           dense_6[0][0]
                                                                 dense_5[0][0]
____________________________________________________________________________________________________
zero_padding2d_25 (ZeroPadding2 (None, 22, 22, 128)  0           concatenate_3[0][0]
____________________________________________________________________________________________________
zero_padding2d_26 (ZeroPadding2 (None, 42, 42, 256)  0           max_pooling2d_3[0][0]
____________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 20, 20, 64)   73792       zero_padding2d_25[0][0]
____________________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 40, 40, 64)   147520      zero_padding2d_26[0][0]
____________________________________________________________________________________________________
dense_7 (Dense)                 (None, 20, 20, 64)   4160        conv2d_22[0][0]
____________________________________________________________________________________________________
dense_8 (Dense)                 (None, 40, 40, 64)   4160        conv2d_23[0][0]
____________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 40, 40, 64)   0           dense_7[0][0]
____________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 40, 40, 128)  0           dense_8[0][0]
                                                                 up_sampling2d_1[0][0]
____________________________________________________________________________________________________
zero_padding2d_27 (ZeroPadding2 (None, 42, 42, 128)  0           concatenate_4[0][0]
____________________________________________________________________________________________________
zero_padding2d_28 (ZeroPadding2 (None, 82, 82, 128)  0           max_pooling2d_2[0][0]
____________________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 40, 40, 64)   73792       zero_padding2d_27[0][0]
____________________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, 80, 80, 64)   73792       zero_padding2d_28[0][0]


____________________________________________________________________________________________________
dense_9 (Dense)                 (None, 40, 40, 64)   4160        conv2d_24[0][0]
____________________________________________________________________________________________________
dense_10 (Dense)                (None, 80, 80, 64)   4160        conv2d_25[0][0]
____________________________________________________________________________________________________
conv2d_transpose_1 (Conv2DTrans (None, 80, 80, 64)   65600       dense_9[0][0]
____________________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 80, 80, 128)  0           dense_10[0][0]
                                                                 conv2d_transpose_1[0][0]
____________________________________________________________________________________________________
zero_padding2d_29 (ZeroPadding2 (None, 82, 82, 128)  0           concatenate_5[0][0]
____________________________________________________________________________________________________
zero_padding2d_30 (ZeroPadding2 (None, 162, 162, 64) 0           max_pooling2d_1[0][0]
____________________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, 80, 80, 64)   73792       zero_padding2d_29[0][0]
____________________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, 160, 160, 64) 36928       zero_padding2d_30[0][0]
____________________________________________________________________________________________________
dense_11 (Dense)                (None, 80, 80, 64)   4160        conv2d_26[0][0]
____________________________________________________________________________________________________
dense_12 (Dense)                (None, 160, 160, 64) 4160        conv2d_27[0][0]
```

```
____________________________________________________________________________________________________
conv2d_transpose_2 (Conv2DTrans (None, 160, 160, 64) 65600       dense_11[0][0]
____________________________________________________________________________________________________
concatenate_6 (Concatenate)     (None, 160, 160, 128 0           dense_12[0][0]
                                                                 conv2d_transpose_2[0][0]
____________________________________________________________________________________________________
zero_padding2d_31 (ZeroPadding2 (None, 162, 162, 128 0           concatenate_6[0][0]
____________________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, 160, 160, 64) 73792       zero_padding2d_31[0][0]
____________________________________________________________________________________________________
dense_13 (Dense)                (None, 160, 160, 64) 4160        conv2d_28[0][0]
____________________________________________________________________________________________________
conv2d_transpose_3 (Conv2DTrans (None, 320, 320, 64) 65600       dense_13[0][0]
====================================================================================================
Total params: 22,681,216
Trainable params: 22,681,216
Non-trainable params: 0
```

Code Snippet: (Socle =1)

```python
from keras.models import Sequential, Model
from keras.layers.core import Flatten, Dense, Dropout
from keras.layers.convolutional import Convolution2D, MaxPooling2D, ZeroPadding2D
from keras.optimizers import SGD
from keras.layers import Input, Concatenate, Conv2DTranspose, Lambda
from keras.utils import plot_model
from keras.backend import dot

#from tensorflow.python.keras.models import Model

import keras
import cv2
import numpy as np

# def VGG(weight_path=None):
model0 = Sequential()
model1 = Sequential()
model2 = Sequential()

#########################################################################################33
#########################################################################################33
#########################################################################################33

main_input0 = Input(shape=(320,320,3), dtype='float32', name='main_input0')
scale0_layer_1 = ZeroPadding2D((1,1))(main_input0)
scale0_layer_2 = Convolution2D(64, (3, 3), activation='relu')(scale0_layer_1)
scale0_layer_3 = ZeroPadding2D((1,1))(scale0_layer_2)
scale0_layer_4 = Convolution2D(64, (3, 3), activation='relu')(scale0_layer_3)
scale0_layer_5 = ZeroPadding2D((1,1))(scale0_layer_4)
scale0_pool_1 = MaxPooling2D((3,3), strides=(2,2))(scale0_layer_5)
################ scale0_pool_1 = scale0_layer_5 ##################

scale0_layer_6 = ZeroPadding2D((1,1))(scale0_pool_1)
scale0_layer_7 = Convolution2D(128, (3, 3), activation='relu')(scale0_layer_6)
scale0_layer_8 = ZeroPadding2D((1,1))(scale0_layer_7)
scale0_layer_9 = Convolution2D(128, (3, 3), activation='relu')(scale0_layer_8)
scale0_layer_10 = ZeroPadding2D((1,1))(scale0_layer_9)
scale0_pool_2 = MaxPooling2D((3,3), strides=(2,2))(scale0_layer_10)
################ scale0_pool_2 = scale0_layer_11 ##################

scale0_layer_12 = ZeroPadding2D((1,1))(scale0_pool_2)
scale0_layer_13 = Convolution2D(256, (3, 3), activation='relu')(scale0_layer_12)
scale0_layer_14 = ZeroPadding2D((1,1))(scale0_layer_13)
scale0_layer_15 = Convolution2D(256, (3, 3), activation='relu')(scale0_layer_14)
scale0_layer_16 = ZeroPadding2D((1,1))(scale0_layer_15)
scale0_layer_17 = Convolution2D(256, (3, 3), activation='relu')(scale0_layer_16)
scale0_layer_18 = ZeroPadding2D((1,1))(scale0_layer_17)
scale0_pool_3 = MaxPooling2D((3,3), strides=(2,2))(scale0_layer_18)
################ scale0_pool_3 = scale0_layer_19 ##################
```

```
scale0_layer_20 = ZeroPadding2D((1,1))(scale0_pool_3)
scale0_layer_21 = Convolution2D(512, (3, 3), activation='relu')(scale0_layer_20)
scale0_layer_22 = ZeroPadding2D((1,1))(scale0_layer_21)
scale0_layer_23 = Convolution2D(512, (3, 3), activation='relu')(scale0_layer_22)
scale0_layer_24 = ZeroPadding2D((1,1))(scale0_layer_23)
scale0_layer_25 = Convolution2D(512, (3, 3), activation='relu')(scale0_layer_24)
scale0_layer_26 = ZeroPadding2D((1,1))(scale0_layer_25)
scale0_pool_4 = MaxPooling2D((3,3), strides=(1,1))(scale0_layer_26)
################ scale0_pool_4 = scale0_layer_27 ###############

scale0_layer_28 = ZeroPadding2D((2,2))(scale0_pool_4)
scale0_layer_29 = Convolution2D(512, (3, 3), activation='relu', dilation_rate=(2,2))(scale0_layer_28)
scale0_layer_30 = ZeroPadding2D((2,2))(scale0_layer_29)
scale0_layer_31 = Convolution2D(512, (3, 3), activation='relu', dilation_rate=(2,2))(scale0_layer_30)
scale0_layer_32 = ZeroPadding2D((2,2))(scale0_layer_31)
scale0_layer_33 = Convolution2D(512, (3, 3), activation='relu', dilation_rate=(2,2))(scale0_layer_32)
scale0_layer_34 = ZeroPadding2D((1,1))(scale0_layer_33)
scale0_pool_5 = MaxPooling2D((3,3), strides=(1,1))(scale0_layer_34)
#############scale0_pool_5 = scale0_layer_35#################

scale0_layer_36 = ZeroPadding2D((12,12))(scale0_pool_5)
scale0_layer_37 = Convolution2D(1024, (3, 3), activation='relu',dilation_rate=(12,12))(scale0_layer_36)
scale0_layer_38 = Dropout(rate = 0.5, noise_shape=None, seed=None)(scale0_layer_37)
scale0_layer_39 = Convolution2D(1024, (1, 1), activation='relu')(scale0_layer_38)
scale0_fc7 = Dropout(rate= 0.5, noise_shape=None, seed=None)(scale0_layer_39)

scale0_mask_1 = Convolution2D(64, (1, 1), strides=(1,1),activation='relu')(scale0_fc7)
scale0_mask_1 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_1)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

scale0_skip_1 = ZeroPadding2D((1,1))(scale0_fc7)
scale0_skip_1 = Convolution2D(64, (3, 3),activation='relu')(scale0_skip_1)
scale0_skip_1 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_1)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

#############################################################
scale0_concat_1 = Concatenate()([scale0_skip_1, scale0_mask_1])
scale0_concat_1 = ZeroPadding2D((1,1))(scale0_concat_1)
scale0_mask_2 = Convolution2D(64, (3, 3), activation='relu')(scale0_concat_1)
scale0_mask_2 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_2)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

scale0_skip_2 = ZeroPadding2D((1,1))(scale0_pool_5)
scale0_skip_2 = Convolution2D(64, (3, 3), activation='relu')(scale0_skip_2)
scale0_skip_2 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_2)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

#################################################3
scale0_concat_2 = Concatenate()([scale0_skip_2, scale0_mask_2])
scale0_mask_3 = ZeroPadding2D((1,1))(scale0_concat_2)
scale0_mask_3 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_3)
scale0_mask_3 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_3)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

scale0_skip_3 = ZeroPadding2D((1,1))(scale0_pool_4)
scale0_skip_3 = Convolution2D(64, (3, 3), activation='relu')(scale0_skip_3)
scale0_skip_3 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_3)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
```

```
#############################################333
scale0_concat_3 = Concatenate()([scale0_skip_3, scale0_mask_3])
scale0_mask_4 = ZeroPadding2D((1,1))(scale0_concat_3)
scale0_mask_4 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_4)
scale0_mask_4 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_4)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
## upsampling to match dimensions

scale0_skip_4 = ZeroPadding2D((1,1))(scale0_pool_3)
scale0_skip_4 = Convolution2D(64, (3, 3), activation='relu')(scale0_skip_4)
scale0_skip_4 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_4)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
```

```
############################################################
scale0_concat_4 = Concatenate()([scale0_skip_4, scale0_mask_4])
scale0_mask_5 = ZeroPadding2D((1,1))(scale0_concat_4)
scale0_mask_5 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_5)
scale0_mask_5 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_5)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
scale0_mask_5_up = Conv2DTranspose(64,(4,4),strides=2,padding = 'same',activation = 'relu')(scale0_mask_5)
# scale0_mask_5_up = keras.layers.UpSampling2D(size=(2, 2),interpolation = 'bilinear',data_format=None)
#(scale0_mask_5)

scale0_skip_5 = ZeroPadding2D((1,1))(scale0_pool_2)
scale0_skip_5 = Convolution2D(64, (3, 3), activation='relu')(scale0_skip_5)
scale0_skip_5 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_5)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

############################################################
scale0_concat_5 = Concatenate()([scale0_skip_5, scale0_mask_5_up])
scale0_mask_6 = ZeroPadding2D((1,1))(scale0_concat_5)
scale0_mask_6 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_6)
scale0_mask_6 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_6)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
scale0_mask_6_up = Conv2DTranspose(64,(4,4),strides=2,padding = 'same',activation = 'relu')(scale0_mask_6)

scale0_skip_6 = ZeroPadding2D((1,1))(scale0_pool_1)
scale0_skip_6 = Convolution2D(64, (3, 3), activation='relu')(scale0_skip_6)
scale0_skip_6 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_skip_6)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)

############################################################
scale0_concat_6 = Concatenate()([scale0_skip_6, scale0_mask_6_up])
scale0_mask_7 = ZeroPadding2D((1,1))(scale0_concat_6)
scale0_mask_7 = Convolution2D(64, (3, 3), activation='relu')(scale0_mask_7)
scale0_mask_7 = Dense(64,kernel_initializer='random_uniform',bias_initializer='zeros')(scale0_mask_7)
keras.initializers.RandomNormal(mean=0.0, stddev=0.01, seed=None)
scale0_mask_7_up = Conv2DTranspose(64,(4,4),strides=2,padding = 'same',activation = 'relu')(scale0_mask_7)


# model0 = Model(inputs = main_input0,outputs = scale0_mask_7_up)
# print(model0.summary())
```