

고려대학교
빅데이터 연구회

KU-BIG

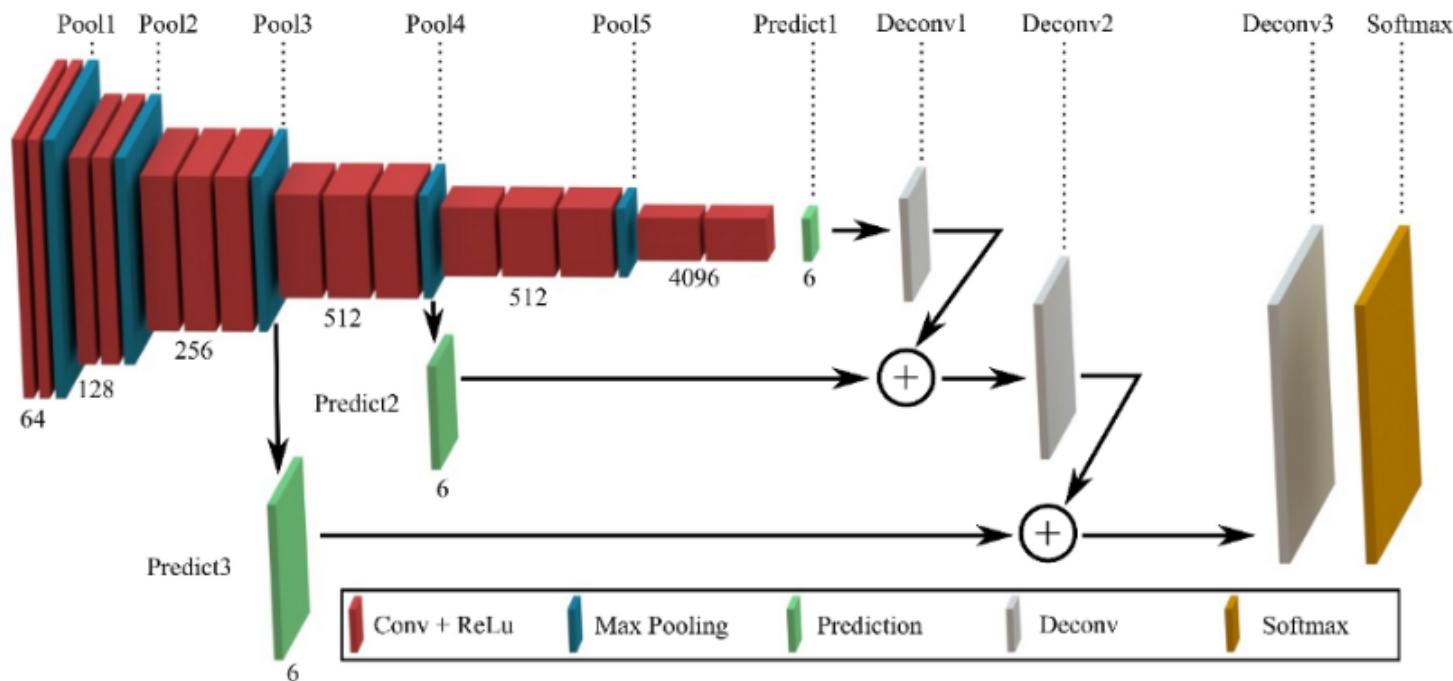
프로젝트 발표

- 딥러닝 1조 -

조원 : 박지윤 윤빈나 이동빈 조송현



Architecture



Convolution+Pooling Layer

```
# Function to create to a series of CONV layers followed by Max pooling layer
def Convblock(channel_dimension, block_no, no_of_convs):
    Layers = []
    for i in range(no_of_convs):
        Conv_name = "conv"+str(block_no)+"_"+str(i+1)
        Layers.append(Convolution2D(channel_dimension,kernel_size = (3,3),padding = "same",activation = "relu",name = Conv_name))

        Max_pooling_name = "pool"+str(block_no)
        Layers.append(MaxPooling2D(pool_size=(2, 2), strides=(2, 2),name = Max_pooling_name))
    return Layers
```

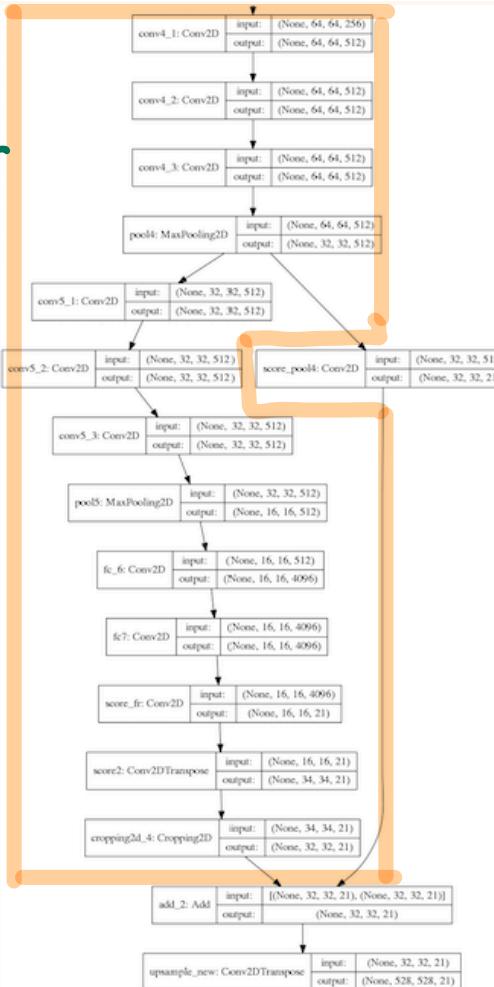
↳ 이 만큼의 conv. layers + 하나의 pooling layer를 이루어짐

Layer (type)	Output Shape	Param #
<hr/>		
conv1_1 (Conv2D)	(None, 512, 512, 64)	1792
<hr/>		
conv1_2 (Conv2D)	(None, 512, 512, 64)	36928
<hr/>		
pool1 (MaxPooling2D)	(None, 256, 256, 64)	0

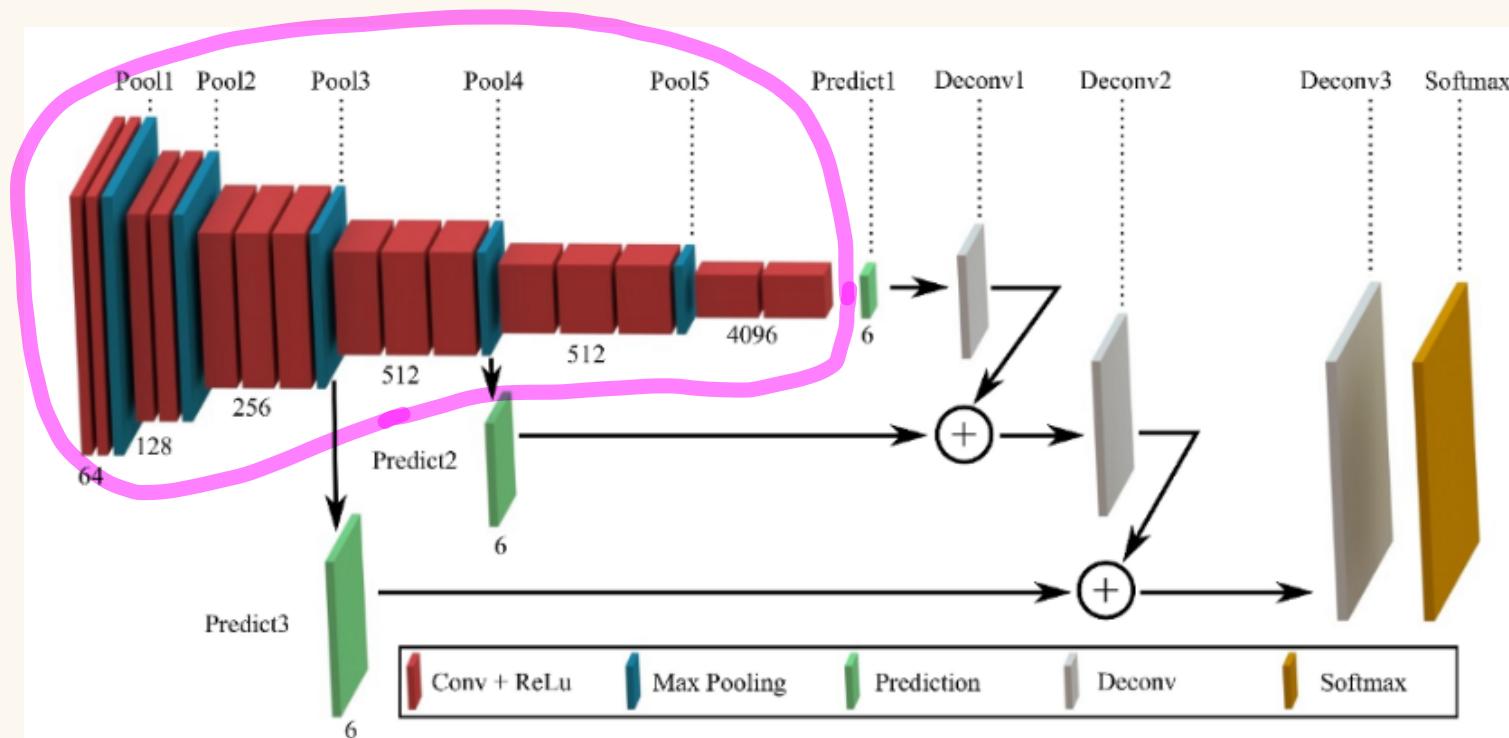
"FCN helper"

- conv. + pooling layers 3
이루어진 5개 blocks

+ 다른 fc layer는
convolution 형태로 따로
놓여짐



Architecture



```
def FCN_16_helper(image_size):
    model = Sequential()
    model.add(Permute((1,2,3),input_shape = (image_size,image_size,3)))

    for l in Convblock(64,1,2): → channel dimension : 64
        model.add(l)
        block # : 1

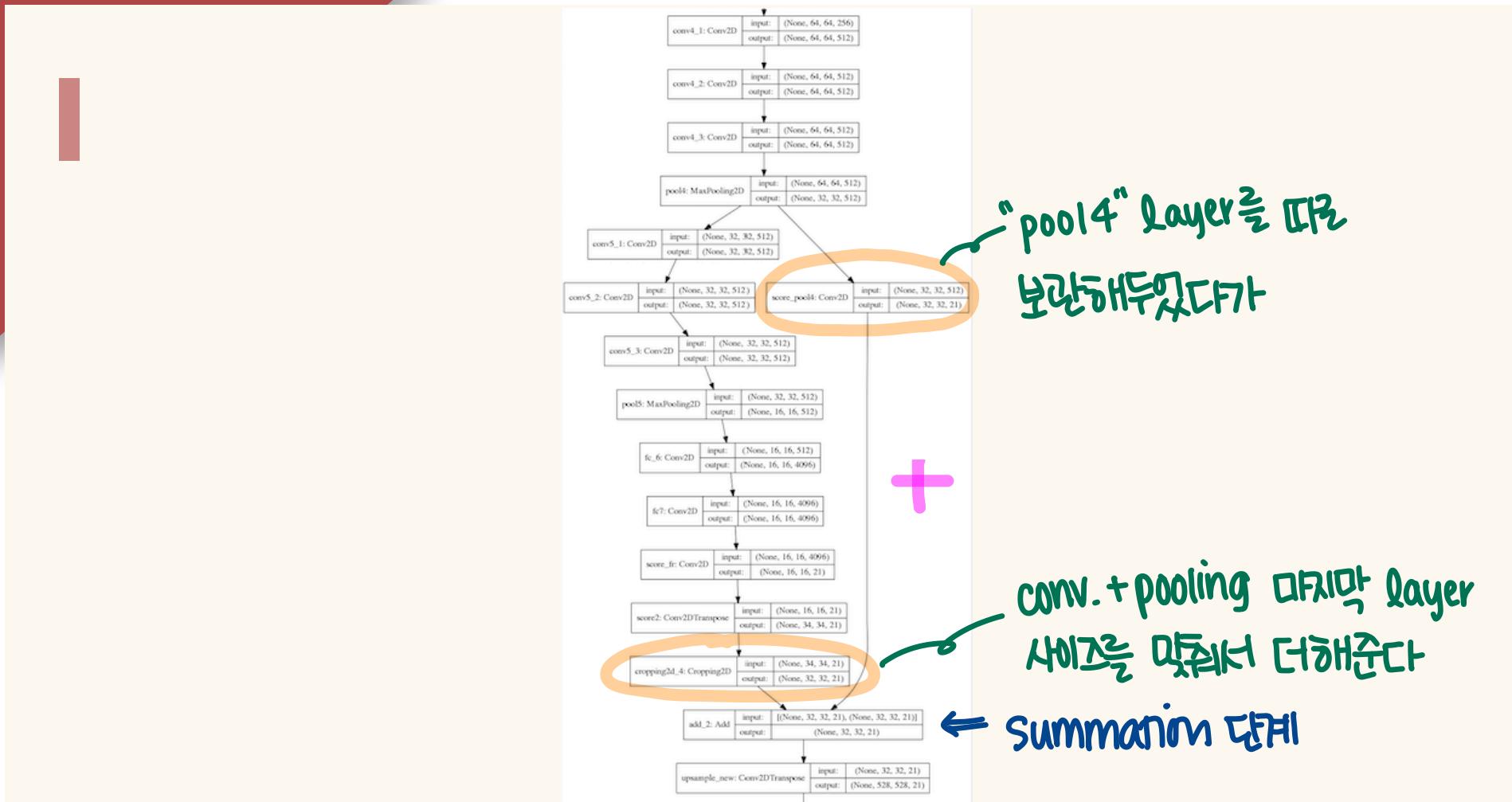
    for l in Convblock(128,2,2): → convolution layer 개수 : 2
        model.add(l)
```

Layer (type)	Output Shape	Param #
=====		
permute_1 (Permute)	(None, 512, 512, 3)	0
conv1_1 (Conv2D)	(None, 512, 512, 64)	1792
conv1_2 (Conv2D)	(None, 512, 512, 64)	36928
pool1 (MaxPooling2D)	(None, 256, 256, 64)	0

CNN -> FCN

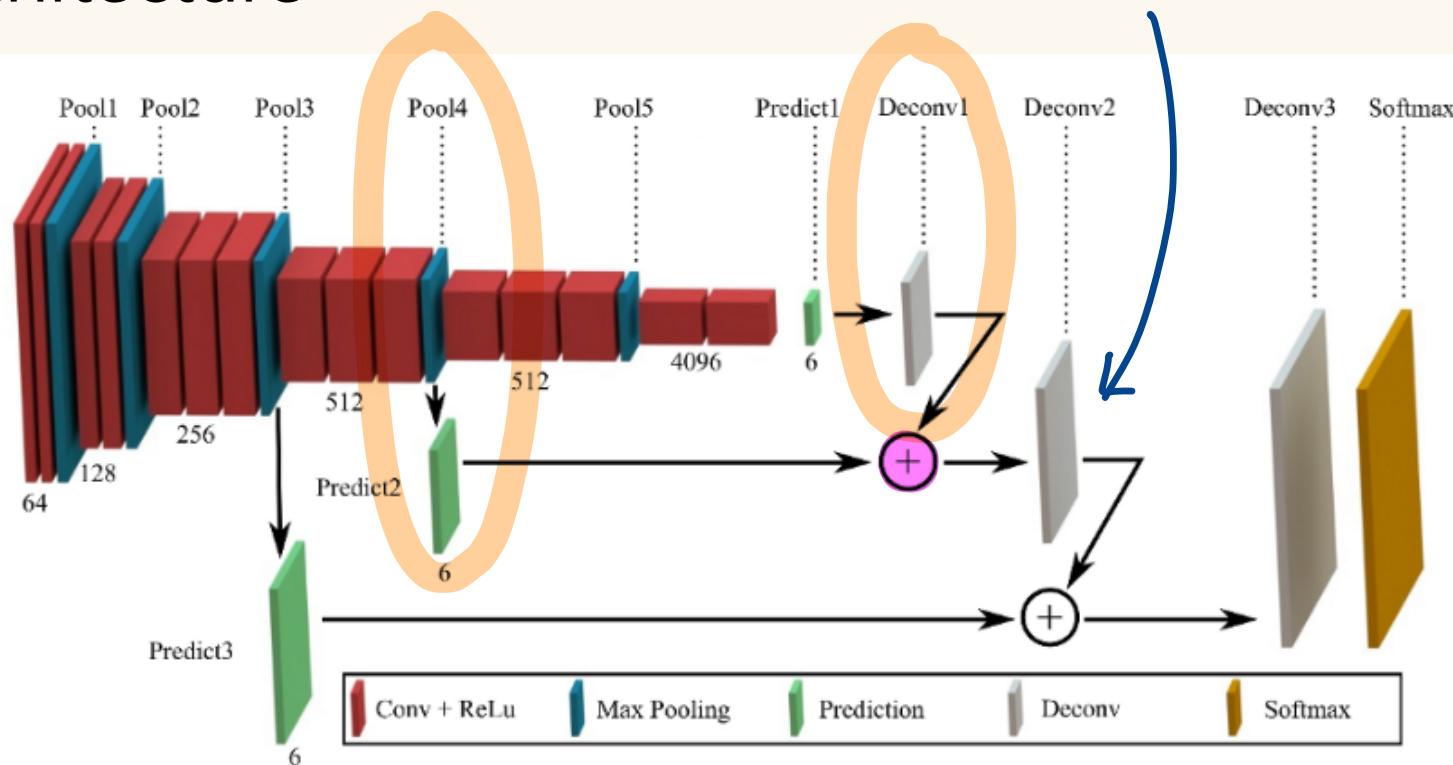
```
model.add(Convolution2D(128,kernel_size=(7,7),padding = "same",activation = "relu",name = "fc_6"))
model.add(Convolution2D(128,kernel_size=(1,1),padding = "same",activation = "relu",name = "fc7"))
```

conv5_1 (Conv2D)	(None, 32, 32, 512)	2359808
conv5_2 (Conv2D)	(None, 32, 32, 512)	2359808
conv5_3 (Conv2D)	(None, 32, 32, 512)	2359808
pool5 (MaxPooling2D)	(None, 16, 16, 512)	0
fc_6 (Conv2D)	(None, 16, 16, 128)	3211392
fc7 (Conv2D)	(None, 16, 16, 128)	16512



Architecture

summation output



Skip Layer

```

def FCN_16(image_size):
    fcn_16 = FCN_16_helper(128)

    skip_con = Convolution2D(1,kernel_size=(1,1),padding = "same",activation=None, name = "score_pool4")

    ① #Addig skip connection which adds the output of Max pooling layer 4 to current layer
    Summed = add(inputs = [skip_con(fcn_16.layers[14].output),fcn_16.layers[-1].output])

    ② Up = Deconvolution2D(1,kernel_size=(32,32),strides = (16,16),padding = "valid",activation = None,name = "upsample_new")

    #Cropping to get the original size of the image
    crop = Cropping2D(cropping = ((0,extra_margin),(0,extra_margin)))
    return Model(fcn_16.input, crop(Up(Summed)))

    output = FCN_16(128)

    print(output.summary())

```

↳ upsampling 단계

①	add_1 (Add)	(None, 8, 8, 1)	0	score_pool4[0][0]
				cropping2d_2[0][0]
②	upsample_new (Conv2DTranspose)	(None, 144, 144, 1)	1025	add_1[0][0]
				cropping2d_3 (Cropping2D) (None, 128, 128, 1) 0 upsample_new[0][0]

문제점 및 갈등...

KU-BIG, 빅데이터 연구회

