이름 : 강시온

학번 : 201802045

학과 : 컴퓨터공학과

코드 설명

```
import tensorflow as tf
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras import Input

inputs = Input(shape=(224, 224, 3))
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv1')(inputs)
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv2')(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool')(x)
f1 = x

# Block 2
x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv1')(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block2_pool')(x)
f2 = x

# Block 3
x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv1')(x)
x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv2')(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block3_pool')(x)
pool3 = x

# Block 4
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv1')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv2')(x)
pool4 = MaxPooling2D((2, 2), strides=(2, 2), name='block4_pool')(x)## (None, 14, 14, 512)

# Block 5
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv1')(pool4)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv2')(x)
pool5 = MaxPooling2D((2, 2), strides=(2, 2), name='block5_pool')(x)## (None, 7, 7, 512)

vgg = tf.keras.Model(inputs, pool5)

o = Conv2D(4096, (7, 7), activation='relu', padding='same', name="conv6")(pool5)
conv7 = Conv2D(3, (1, 1), activation='relu', padding='same', name="conv7")(o)

conv7_4 = UpSampling2D(size=(2, 2), interpolation='bilinear')(conv7)
conv7_4 = Conv2D(3, (3, 3), (1, 1), activation='relu', padding='same')(conv7_4)
## (None, 224, 224, 10)
## 2 times upsampling for pool411
pool411 = Conv2D(3, (1, 1), activation='relu', padding='same', name="pool4_11")(pool4)
pool411 = Add(name="add")([pool411, conv7_4])
pool411_2 = UpSampling2D(size=(2, 2), interpolation='bilinear')(pool411)
pool411_2 = Conv2D(3, (3, 3), (1, 1), activation='relu', padding='same')(pool411_2)

pool311 = Conv2D(3, (1, 1), activation='relu' , padding='same', name="pool3_11")(pool3)
o = Add(name="add2")([pool411_2, pool311])
o = UpSampling2D(size=(8, 8), interpolation='bilinear')(o)
o = Conv2D(3, (3, 3), (1, 1), activation='relu', padding='same')(o)

fcn_model = tf.keras.Model(inputs, o)

fcn_model.summary()
```

Encode하는 부분은 vgg와 비슷하지만 conv레이어의 갯수가 달라져 이미지와 같이 맞춰주었고, 나중에 add해줘야하므로 따로 변수에 저장해두었다.

그리고 decode하는 부분에서는 사이즈를 키울때는 upsampling2d 레이어를 활용하였다.

결과 이미지

```
from tensorflow import keras
# model_fcn.compile(optimizer=keras.optimizers.SGD(momentum=0.9,nesterov=True),loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True))
fcn_model.compile(optimizer=keras.optimizers.Adam(),loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True))

EPOCHS = 100
VAL_SUBSPLITS = 5
VALIDATION_STEPS = info.splits['test'].num_examples//BATCH_SIZE//VAL_SUBSPLITS

model_history = fcn_model.fit(train_dataset, epochs=EPOCHS,
                          steps_per_epoch=STEPS_PER_EPOCH,
                          validation_steps=VALIDATION_STEPS,
                          validation_data=test_dataset,
                          callbacks=[DisplayCallback()])
```


```
에포크 이후 예측 예시 72

57/57 [==============================] - 165s 3s/step - loss: 0.1198 - val_loss: 0.5213
Epoch 73/100
17/57 [=======>......................] - ETA: 1:45 - loss: 0.1177
```

난이도

생각보다 난이도가 있는과제였다.

Upsampling을 하는 방법도 여러가지가있었고 vgg 모델도 약간 다른부분이있어 그 부분도 맞춰주고 중간 레이어를 나중에 다시 add해주는 작업도 있어 꽤나 복잡한 모델이었다.