Instance Segmentation with Mask R-CNN

AI Computer Vision Project, AI Innovation Square

2020.03.23 – 2020.04.10

JIHUN KIM
YUJIN NAM
YOONSUNG LEE

AI Innovation Square

1. Mask R-CNN

1-1. What’s Mask R-CNN?

1-2. Problem

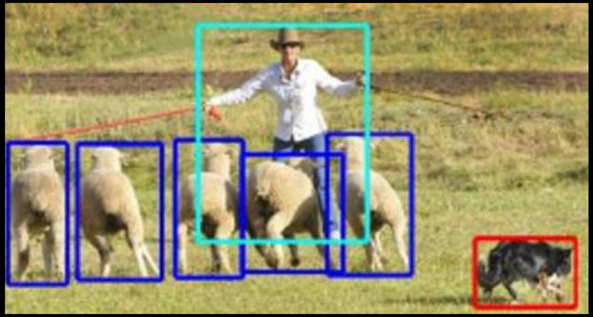
2. Modified Mask R-CNN

2-1. What’s the difference with original Mask R-CNN?

2-2. Specified Architecture

3. Conclusion

1-1. What’s Mask R-CNN?

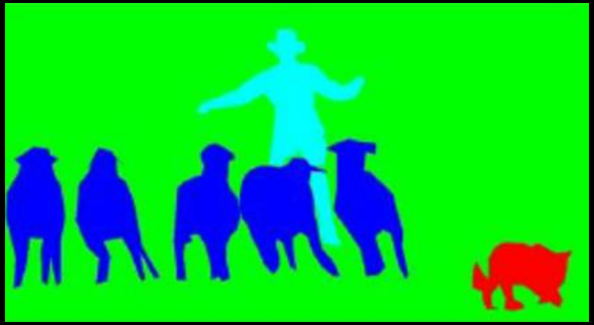


BBOX
Classification

Can Separate
Cannot Segment

Faster R-CNN

+

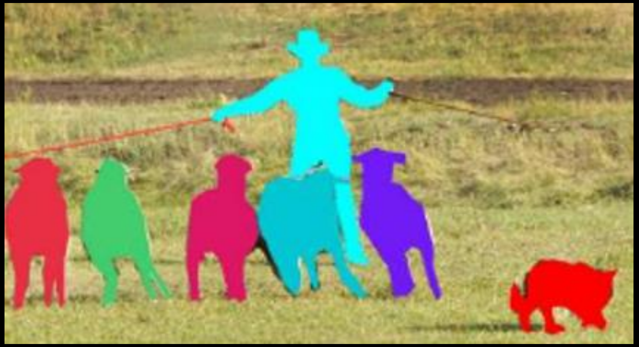


Segmentation
Classification

Cannot Separate
Can Segment

FCN

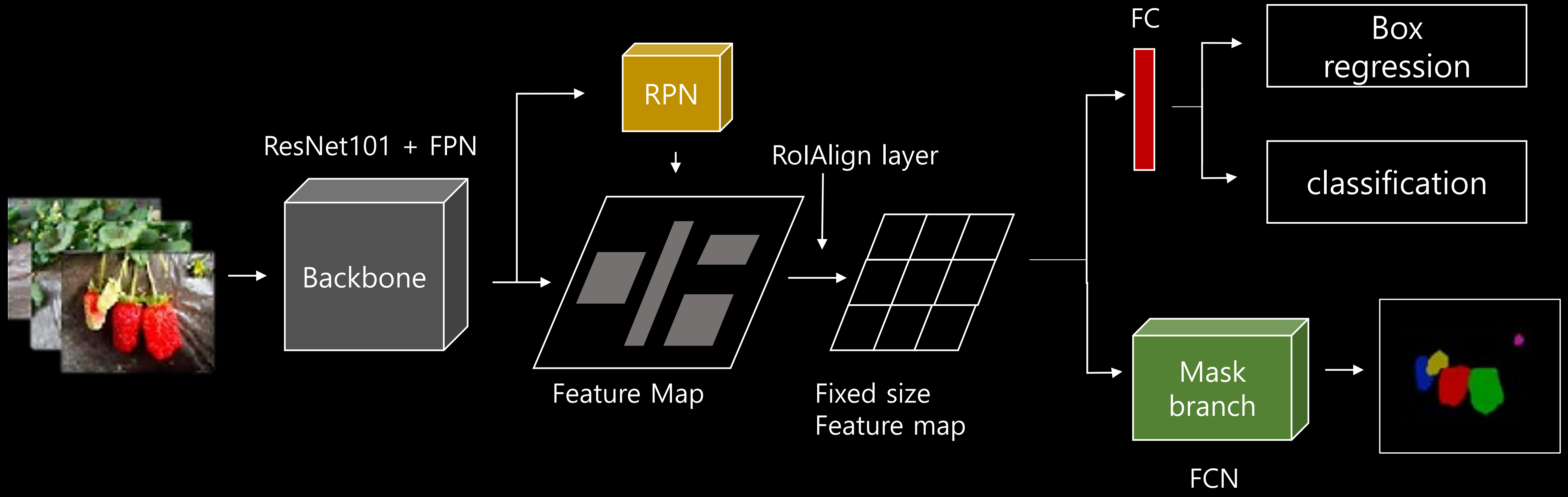
=



Segmentation
In Bbox
Classification

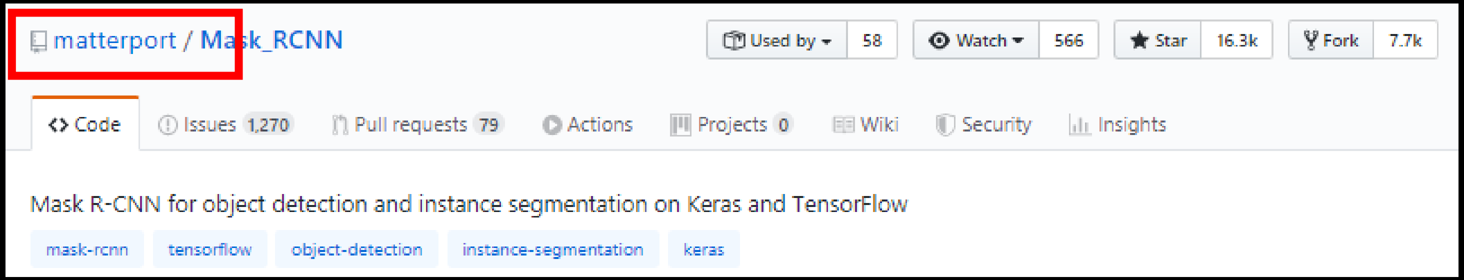
Head Architecture

Faster R-CNN + Binary Mask Prediction + FCN + RoI Align



1-1. What's Mask R-CNN?

How to use Mask R-CNN?

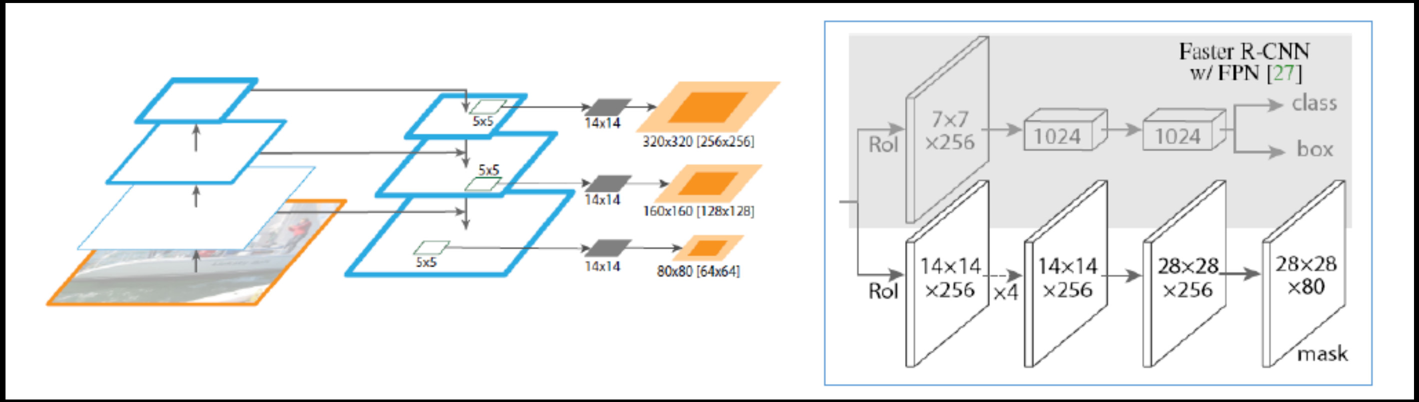


matterport / Mask_RCNN

Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow

mask-rcnn tensorflow object-detection instance-segmentation keras

ResNet101(backbone) + FPN(Binary Masking)

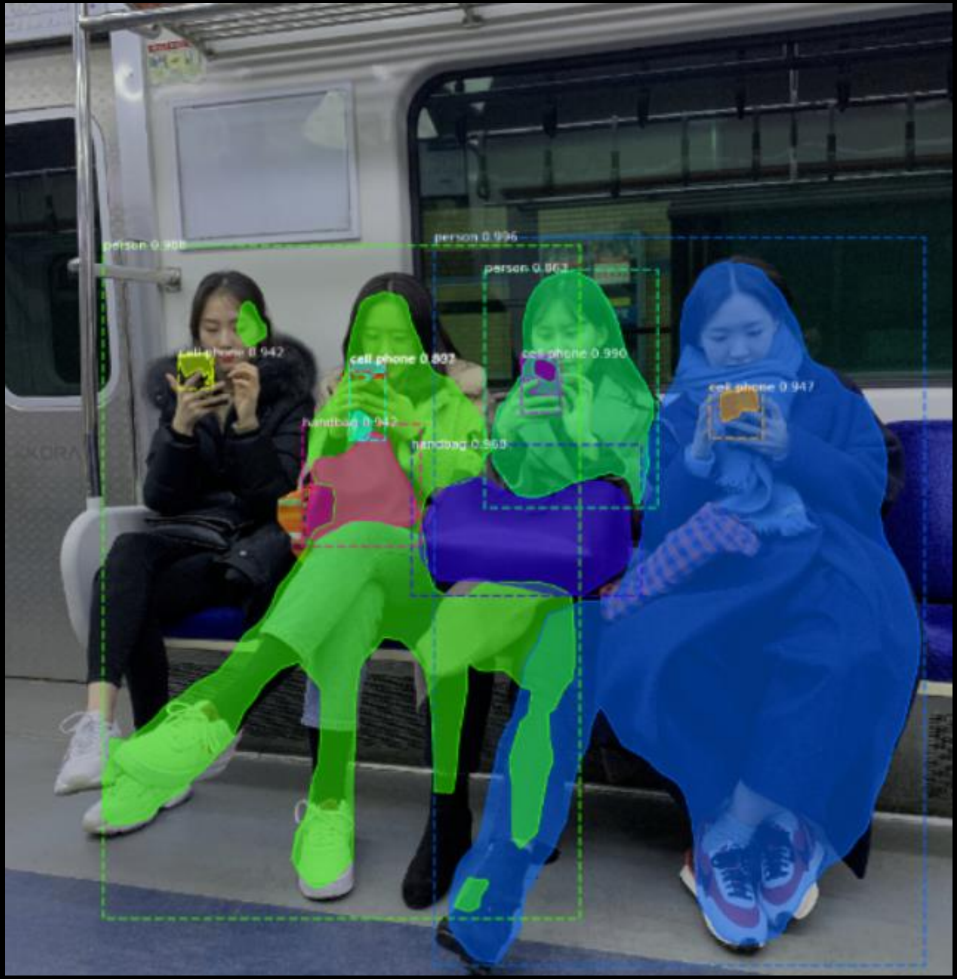


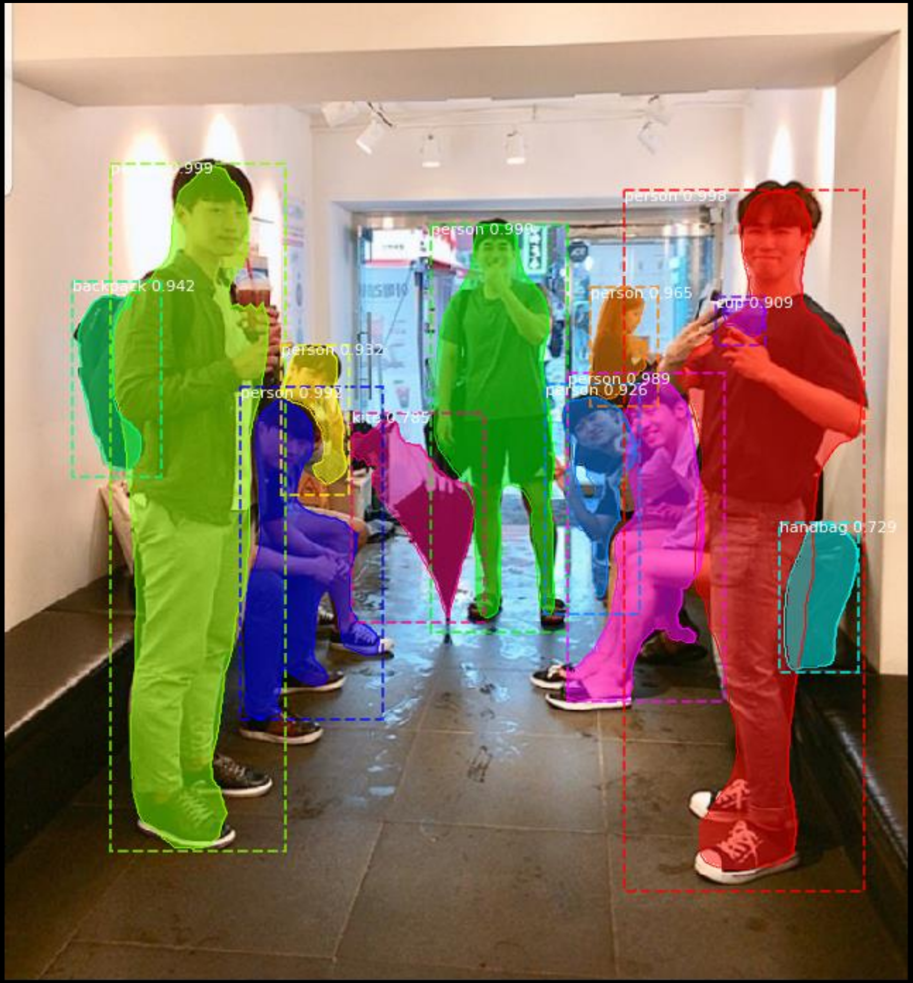
COCO Dataset

AI Innovation Square

1-1. What's Mask R-CNN?

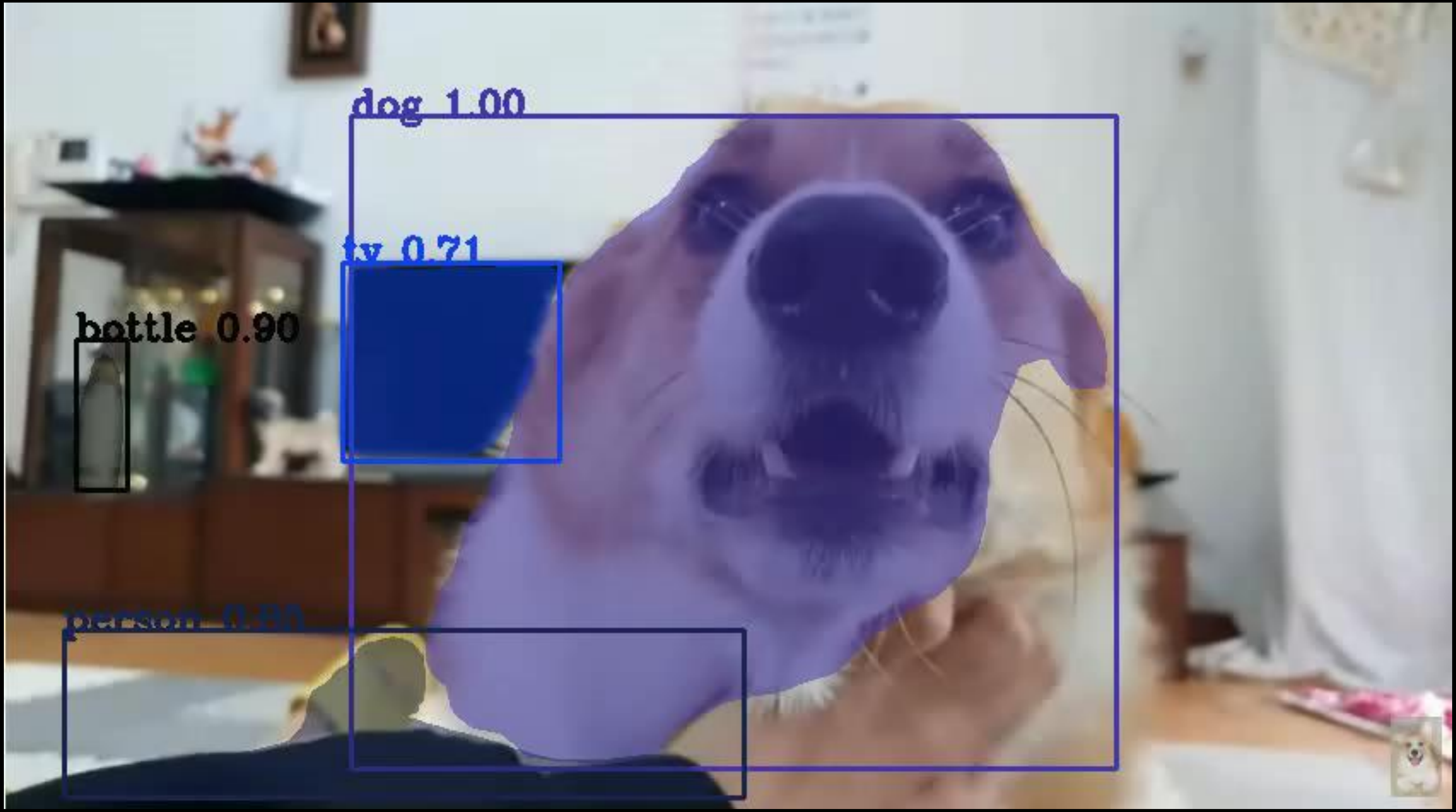
How it works?



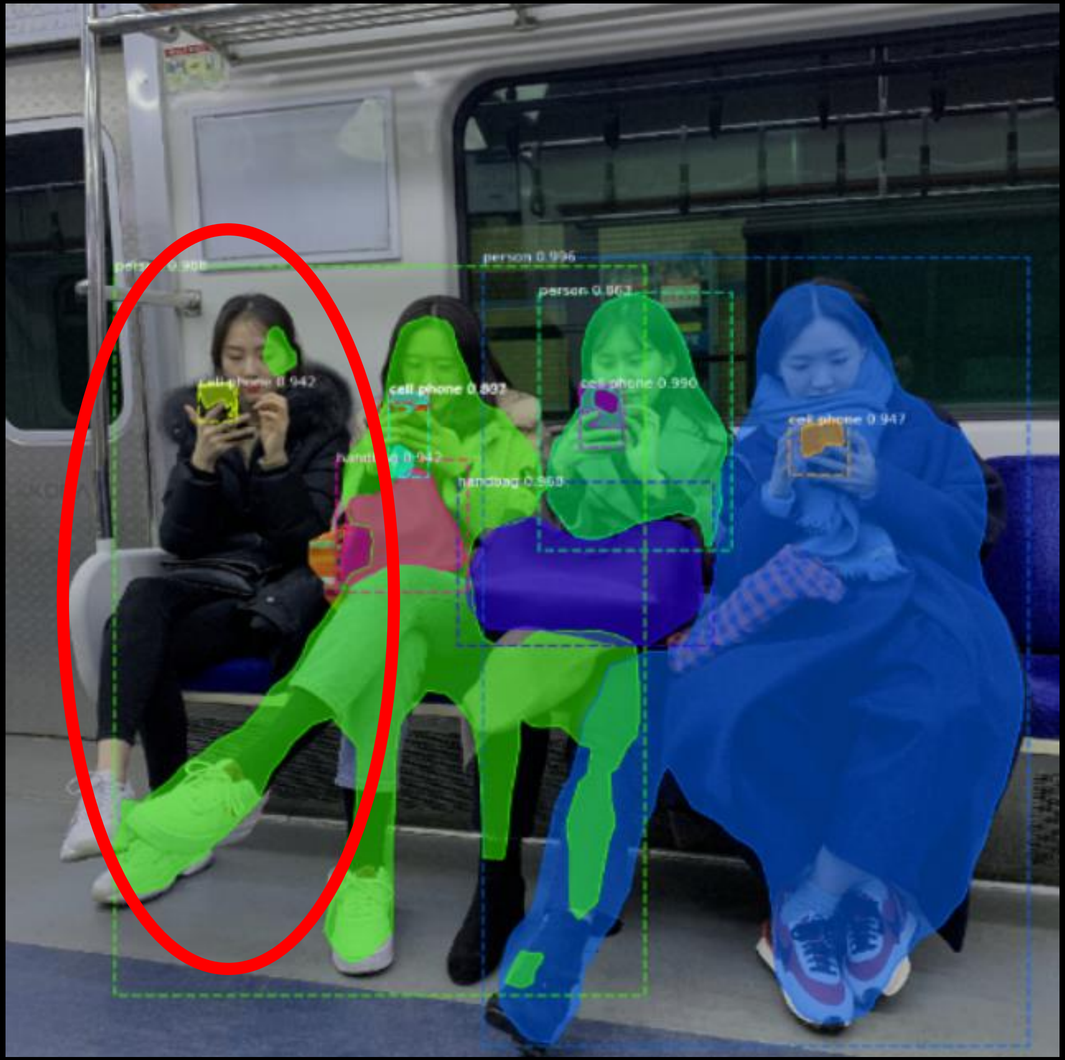


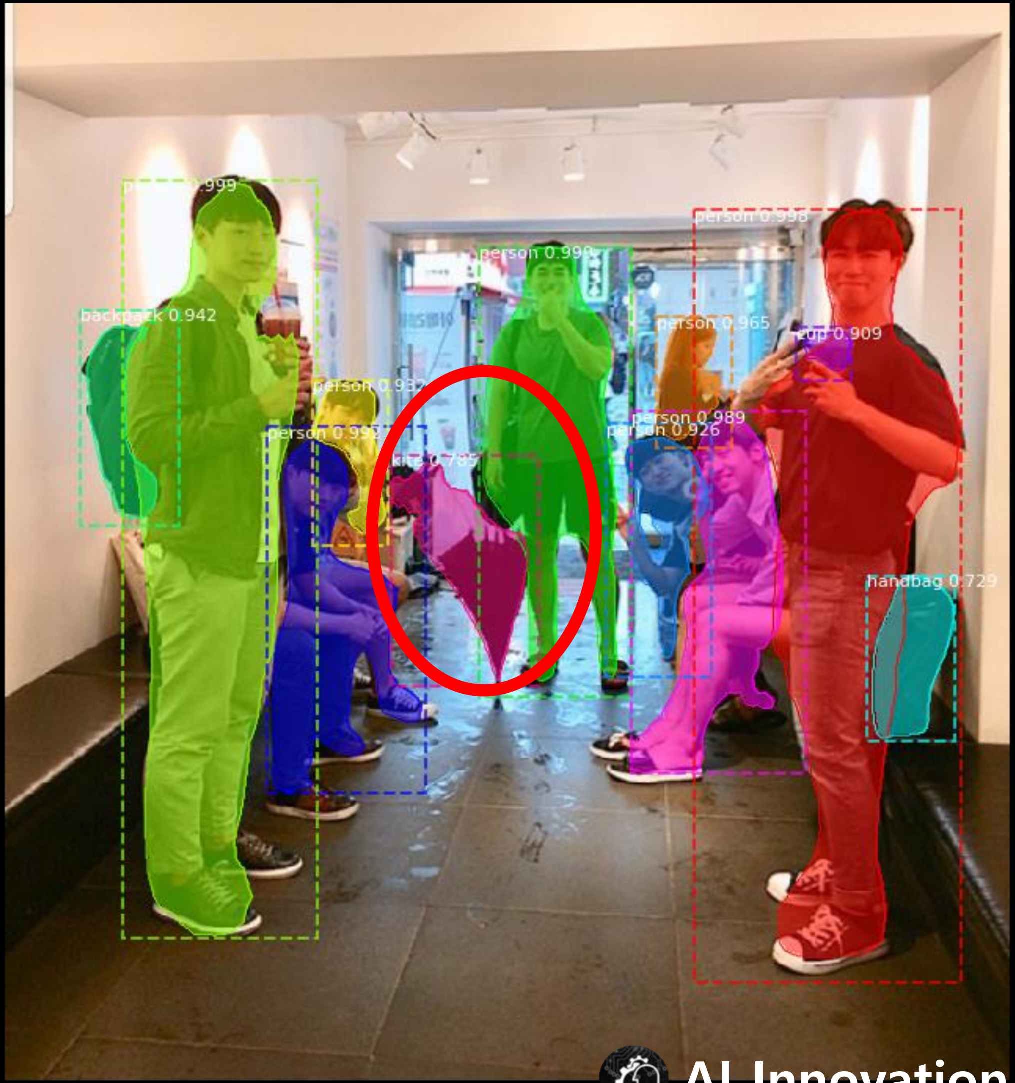
AI Innovation Square

How it works?



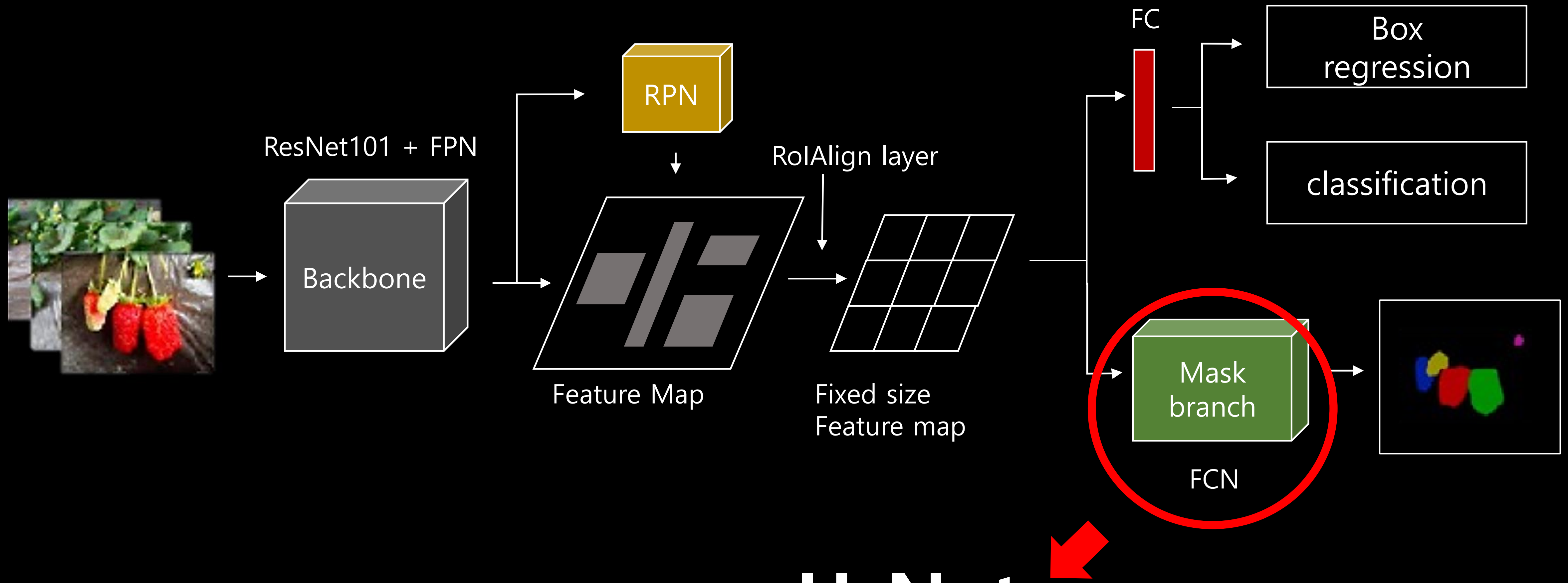
1-2. Problem


person 0.996
person 0.862
cell phone 0.942
cell phone 0.802
cell phone 0.990
cell phone 0.947
handbag 0.942
handbag 0.960


backpack 0.942
person 0.990
person 0.998
person 0.965
cup 0.909
person 0.937
person 0.992
kite 0.785
person 0.989
person 0.926
handbag 0.729

2-1. What’s the difference with original Mask R-CNN?

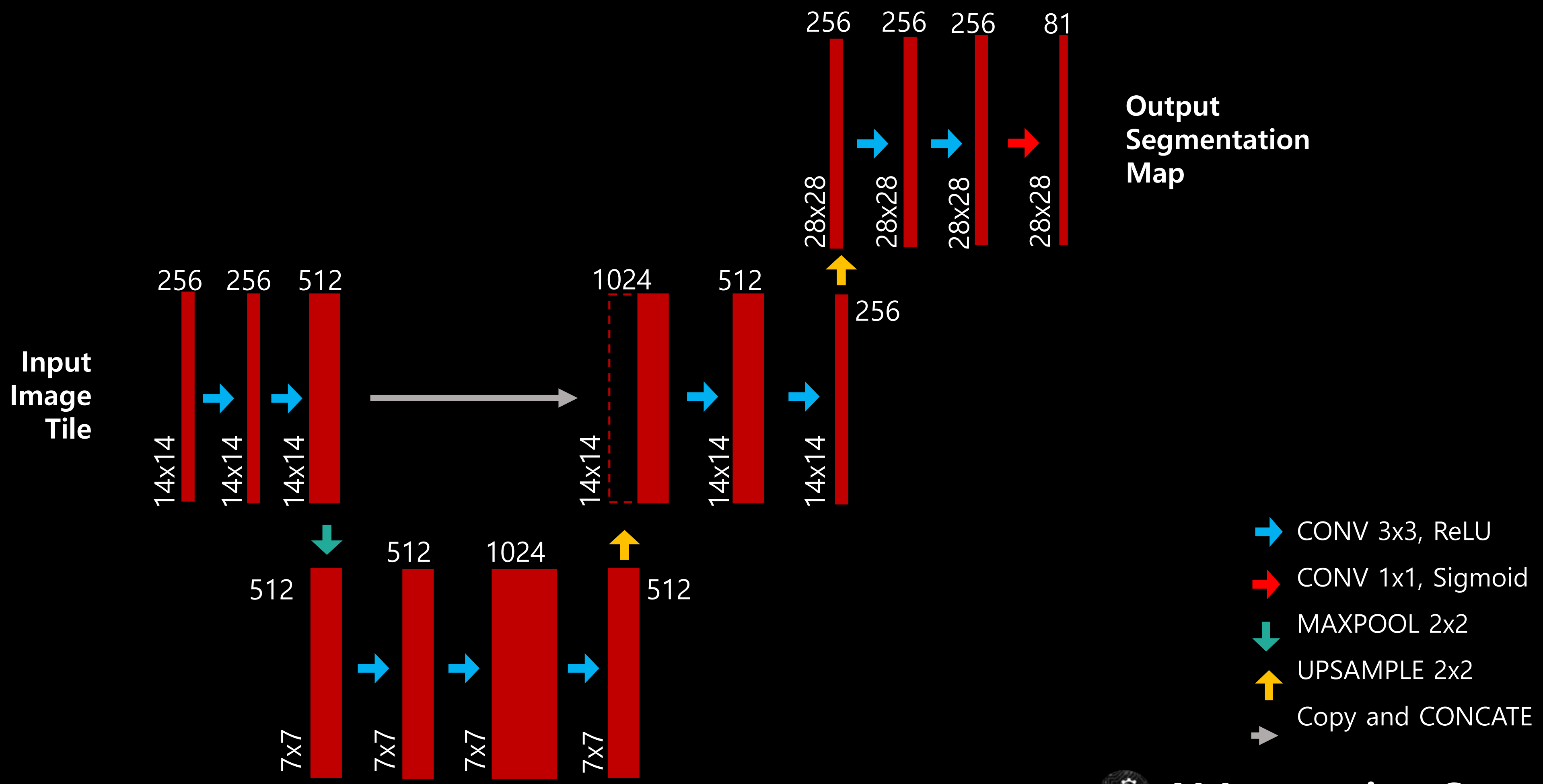
Head Architecture



U-Net

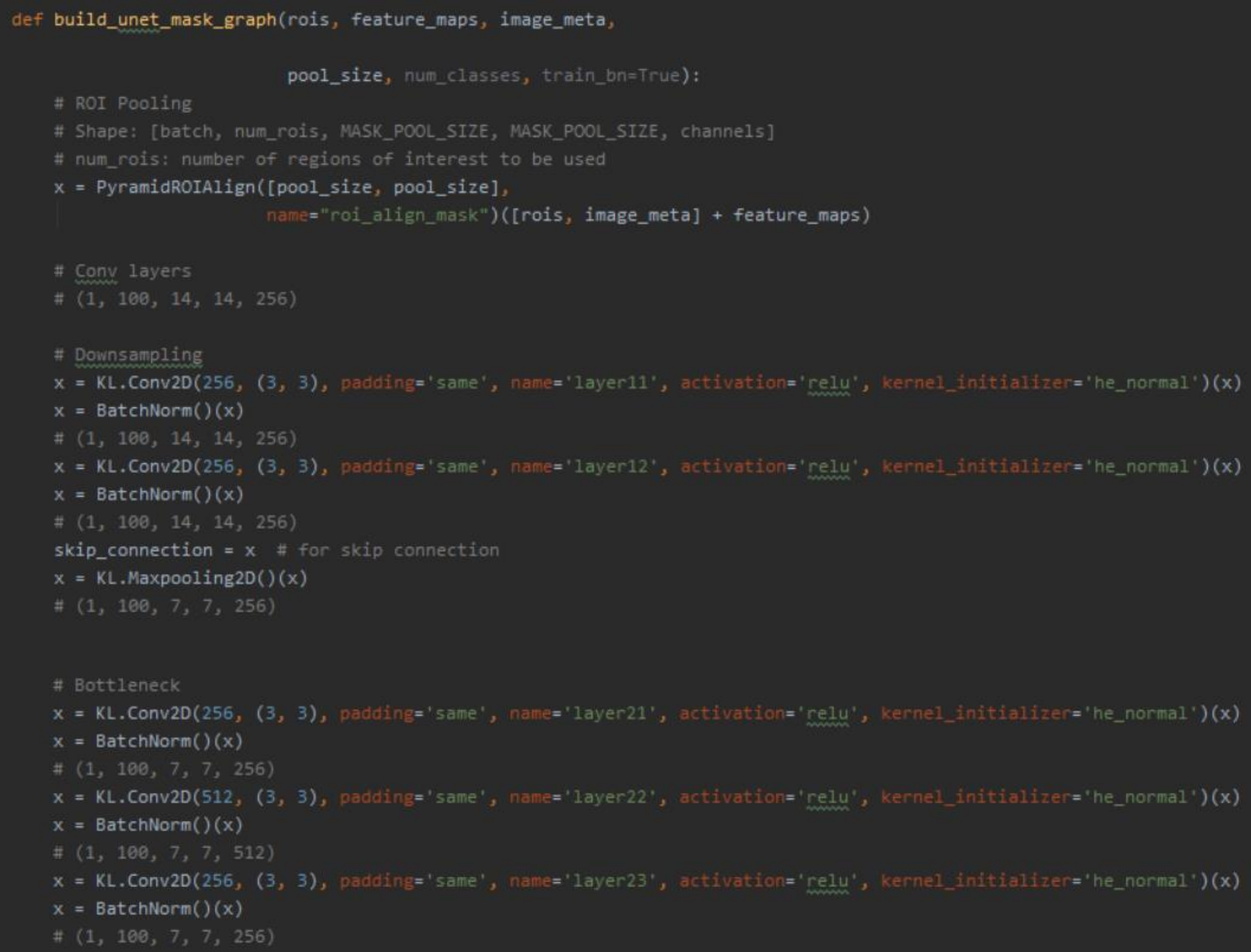
2-1. What's the difference with original Mask R-CNN?

Modified U-Net



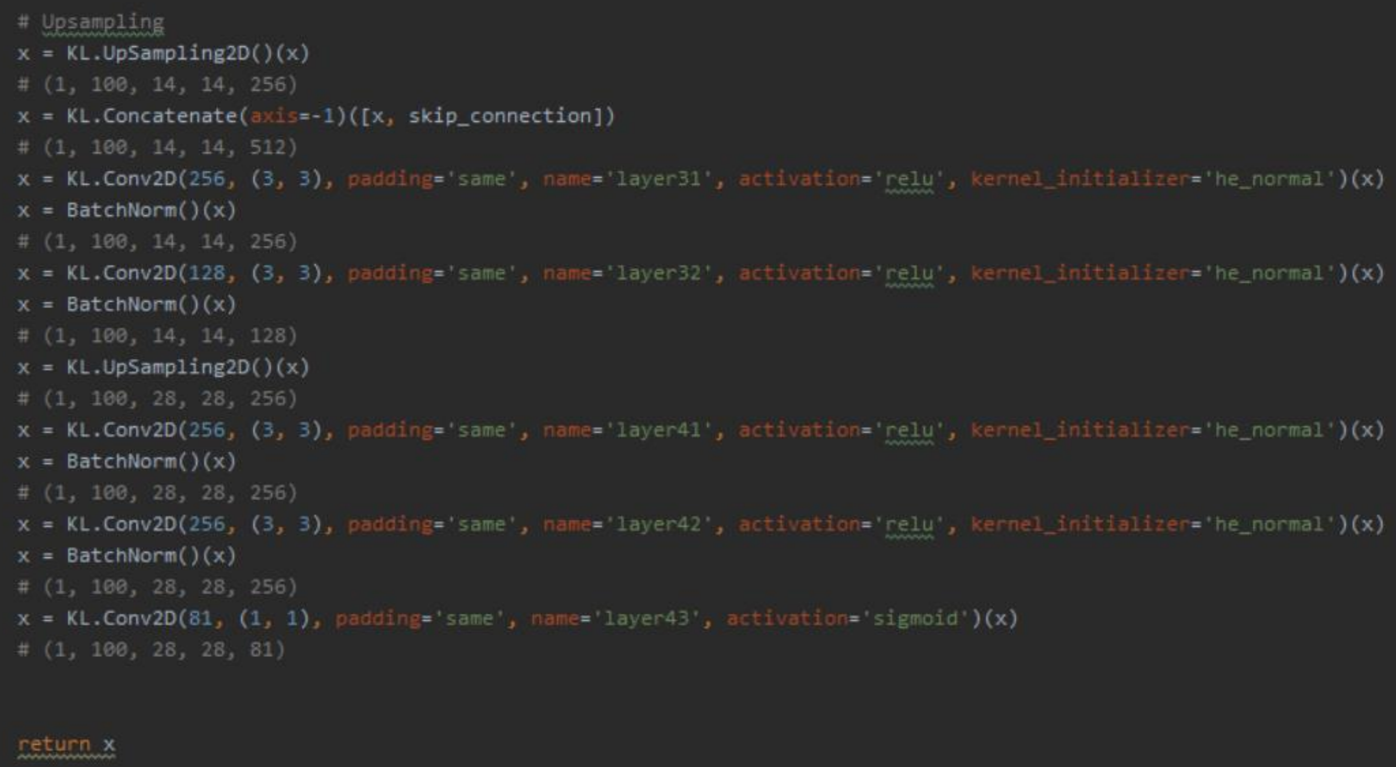
AI Innovation Square

2-1. What’s the difference with original Mask R-CNN?



```
def build_unet_mask_graph(rois, feature_maps, image_meta,

                          pool_size, num_classes, train_bn=True):
    # ROI Pooling
    # Shape: [batch, num_rois, MASK_POOL_SIZE, MASK_POOL_SIZE, channels]
    # num_rois: number of regions of interest to be used
    x = PyramidROIAlign([pool_size, pool_size],
                        name="roi_align_mask")([rois, image_meta] + feature_maps)

    # Conv layers
    # (1, 100, 14, 14, 256)

    # Downsampling
    x = KL.Conv2D(256, (3, 3), padding='same', name='layer11', activation='relu', kernel_initializer='he_normal')(x)
    x = BatchNorm()(x)
    # (1, 100, 14, 14, 256)
    x = KL.Conv2D(256, (3, 3), padding='same', name='layer12', activation='relu', kernel_initializer='he_normal')(x)
    x = BatchNorm()(x)
    # (1, 100, 14, 14, 256)
    skip_connection = x  # for skip connection
    x = KL.Maxpooling2D()(x)
    # (1, 100, 7, 7, 256)


    # Bottleneck
    x = KL.Conv2D(256, (3, 3), padding='same', name='layer21', activation='relu', kernel_initializer='he_normal')(x)
    x = BatchNorm()(x)
    # (1, 100, 7, 7, 256)
    x = KL.Conv2D(512, (3, 3), padding='same', name='layer22', activation='relu', kernel_initializer='he_normal')(x)
    x = BatchNorm()(x)
    # (1, 100, 7, 7, 512)
    x = KL.Conv2D(256, (3, 3), padding='same', name='layer23', activation='relu', kernel_initializer='he_normal')(x)
    x = BatchNorm()(x)
    # (1, 100, 7, 7, 256)
```

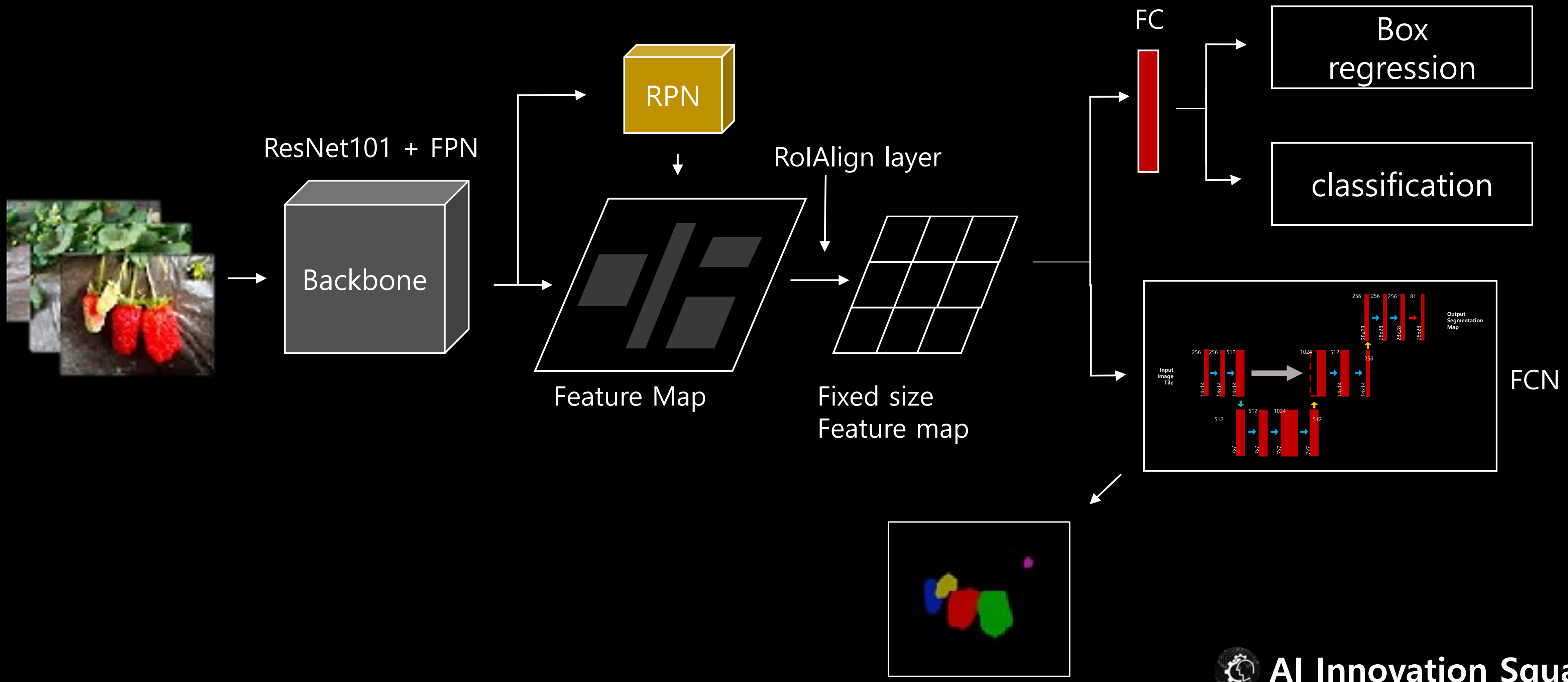
2-1. What’s the difference with original Mask R-CNN?



```
# Upsampling
x = KL.UpSampling2D()(x)
# (1, 100, 14, 14, 256)
x = KL.Concatenate(axis=-1)([x, skip_connection])
# (1, 100, 14, 14, 512)
x = KL.Conv2D(256, (3, 3), padding='same', name='layer31', activation='relu', kernel_initializer='he_normal')(x)
x = BatchNorm()(x)
# (1, 100, 14, 14, 256)
x = KL.Conv2D(128, (3, 3), padding='same', name='layer32', activation='relu', kernel_initializer='he_normal')(x)
x = BatchNorm()(x)
# (1, 100, 14, 14, 128)
x = KL.UpSampling2D()(x)
# (1, 100, 28, 28, 256)
x = KL.Conv2D(256, (3, 3), padding='same', name='layer41', activation='relu', kernel_initializer='he_normal')(x)
x = BatchNorm()(x)
# (1, 100, 28, 28, 256)
x = KL.Conv2D(256, (3, 3), padding='same', name='layer42', activation='relu', kernel_initializer='he_normal')(x)
x = BatchNorm()(x)
# (1, 100, 28, 28, 256)
x = KL.Conv2D(81, (1, 1), padding='same', name='layer43', activation='sigmoid')(x)
# (1, 100, 28, 28, 81)


return x
```

2-2. Specified Architecture

Modified Head Architecture



ResNet101 + FPN

Backbone

RPN

Feature Map

RoIAlign layer

Fixed size
Feature map

FC

Box
regression

classification

FCN

AI Innovation Square

3. Conclusion

Paper Review

↓

Build an Architecture

↓

Train CoCo dataset

↓

Transfer learning
with iMaterialist dataset

Dataset : iMaterialist (Fashion) 2019 at FGVC6 (19GB)

GPU : GeForce RTX 2080 Ti
CUDA Toolkit 9.0
Anaconda (python3.7)

---

numpy
scipy
Pillow
cython
matplotlib
scikit-image
tensorflow>=1.3.0
keras>=2.0.8
opencv-python
h5py
imgaug
IPython[all]

Q & A