

인공지능 개론

L1 1.1 Vision Applications with Torchvision

국민대학교

소프트웨어융합대학원

박하명

Colab에서 구글드라이브 연결하기 (복습)

- Colab에서 구글 드라이브 mount (연결) 하기

```
from google.colab import drive  
drive.mount('/content/gdrive')
```

클릭

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6c

Enter your authorization code:

복사 붙여넣기

이 코드를 복사하여 애플리케이션으로 전환한 다음 붙여넣으세요.

4/zQGt-

HvkjWMqyMKyvZ

zwJpb3z1Qfn83EQ



필요한 모듈 불러오기

```
import torch
import torchvision
from torchvision import models # pre-trained model 사용
import torchvision.transforms as T

from PIL import Image # Python Image Library
import matplotlib.pyplot as plt
from matplotlib.path import Path # 선 그릴 때 사용
from matplotlib.patches import Rectangle, Circle, PathPatch
# 사각형, 점, 선 그릴 때 사용
```

이미지 불러오기

- PLT의 Image 모듈을 사용하여 이미지 불러오기
- 이미지 크기 조절
- matplotlib으로 그려보기



```
IMG_SIZE = 480
```

```
img = Image.open("/content/gdrive/My Drive/img1.jpg")
```

```
img = img.resize( (IMG_SIZE, int(img.height * IMG_SIZE / img.width)) )
```

```
plt.figure(figsize=(10,10)) # 표시되는 이미지 크기 조절
```

```
plt.imshow(img)
```

Object Detection 모델 불러오기

- pretrained=True 파라미터를 넣어주면 미리 학습된 모델을 불러온다..!

```
model = models.detection.keypoint_rcnn_resnet50_fpn(pretrained=True)
model.eval()
```

Object Detection

- pytorch 모델의 입력은 tensor 형태여야함 → 변환해주기
- 사용하는 모델은 이미지의 목록(list)를 입력으로 받음

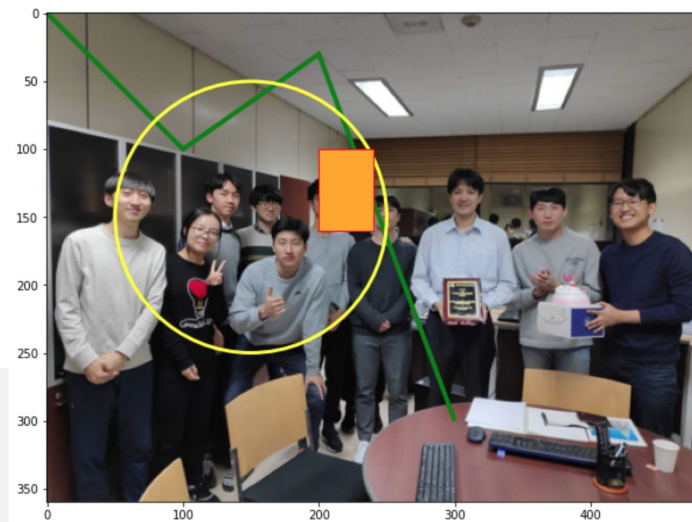
```
trans = T.ToTensor()  
img_tensor = trans(img)  
  
out = model([img_tensor])[0]  
print(out.keys())
```

처리 결과를 시각화하기

- matplotlib을 이용하여 처리 결과를 시각화하기
- 사각형, 점, 선 그리기

```
plt.figure(figsize=(10, 10))  
ax = plt.gca()  
ax.imshow(img)
```

```
line = Path([(0,0), (100, 100), (200, 30), (300, 300)])  
ax.add_patch(PathPatch(line, edgecolor='green', facecolor='none', linewidth=4))  
ax.add_patch(Circle((150, 150), radius=100, edgecolor='yellow', facecolor='none', linewidth=3))  
ax.add_patch(Rectangle((200, 100), 40, 60, edgecolor='red', facecolor='orange'))
```



처리 결과를 시각화하기

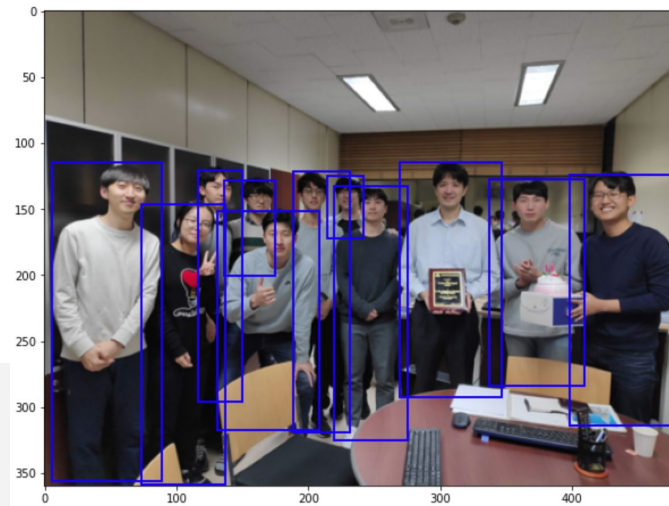
- 찾은 개체에 사각형 그리기

```
plt.figure(figsize=(10, 10))
ax = plt.gca()
ax.imshow(img)
```

```
zipped = zip(out['scores'].tolist(), out['boxes'].tolist(), out['keypoints'][:, :, :2].tolist())
```

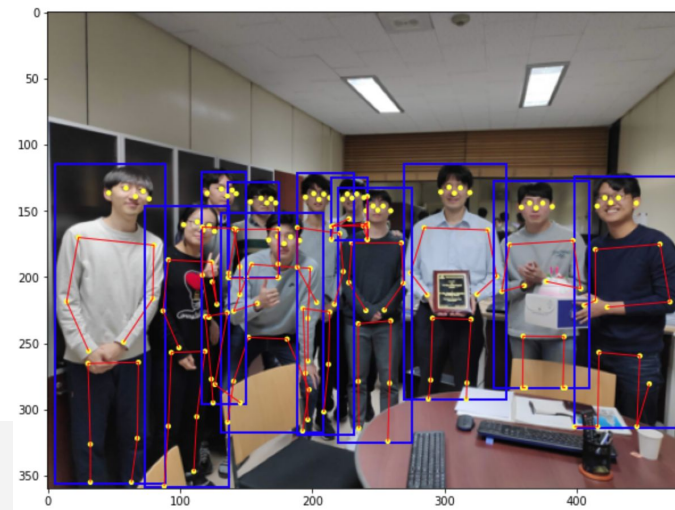
```
for score, box, keys in zipped:
    if score < 0.85:
        continue
```

```
rect = Rectangle((box[0], box[1]), # position
                  box[2]-box[0], # width
                  box[3]-box[1], # height
                  linewidth=2, facecolor='none', edgecolor='blue')
ax.add_patch(rect)
```



처리 결과를 시각화하기

- 키포인트에 점찍고 선으로 잇기



...

```
for k in keys:
```

```
    ax.add_patch(Circle((k[0], k[1]), radius=2, facecolor='yellow', edgecolor='none'))
```

```
line = Path([keys[10], keys[8], keys[6], keys[5], keys[7], keys[9]])
```

```
ax.add_patch(PathPatch(line, linewidth=1, edgecolor='red', facecolor='none'))
```

```
line = Path([keys[16], keys[14], keys[12], keys[11], keys[13], keys[15]])
```

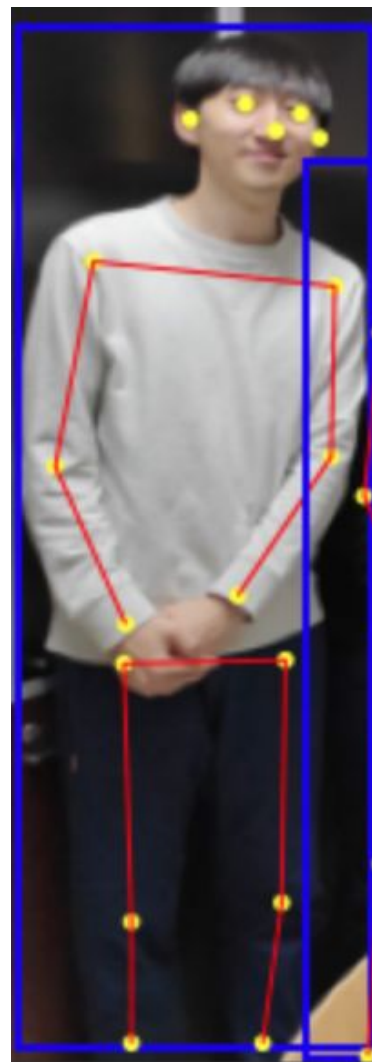
```
ax.add_patch(PathPatch(line, linewidth=1, edgecolor='red', facecolor='none'))
```

참고: 키포인트 순서

```
COCO_PERSON_KEYPOINT_NAMES = [
```

```
    'nose', # 0
    'left_eye', # 1
    'right_eye', # 2
    'left_ear', # 3
    'right_ear', # 4
    'left_shoulder', # 5
    'right_shoulder', # 6
    'left_elbow', # 7
    'right_elbow', # 8
    'left_wrist', # 9
    'right_wrist', # 10
    'left_hip', # 11
    'right_hip', # 12
    'left_knee', # 13
    'right_knee', # 14
    'left_ankle', # 15
    'right_ankle' # 16
```

```
]
```



새로운 이미지 불러오기

```
IMG_SIZE = 480

img = Image.open("/content/gdrive/My Drive/img2.jpg")
img = img.resize( (IMG_SIZE, int(img.height * width / img.width)) )

plt.imshow(img)
```

Semantic Segmentation 모델 불러오기

- 구글도 사용한다는 DeepLabV3 사용해보기
- pre-trained model은 20개의 class를 가짐 (배경까지 21개)

```
model = models.segmentation.deeplabv3_resnet101(pretrained=True).eval()
```

```
['__background__', 'aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus',  
'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse', 'motorbike',  
'person', 'pottedplant', 'sheep', 'sofa', 'train', 'tvmonitor']
```

Semantic Segmentation

- 입력 형태 변환
- 모델에 이미지 넣어 semantic segmentation 결과 얻기

```
trans = T.Compose([
    T.ToTensor(),
    T.Normalize([ 0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

img_tensor = trans(img)

out = model(img_tensor.unsqueeze( 0))['out']
```

처리결과 시각화하기

- 영역마다 랜덤한 색상 채우기

```
import numpy as np
import random as rnd

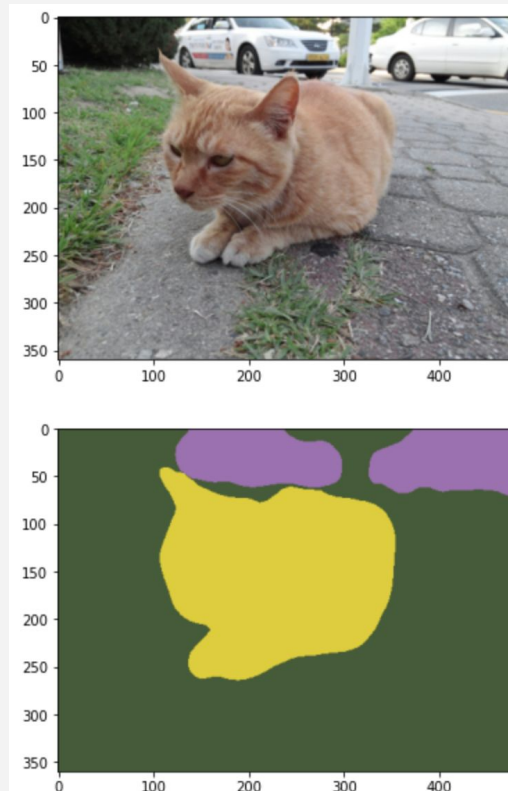
colors = [ (rnd.random(), rnd.random(), rnd.random()) for i in range(21)]

out_pred = out.argmax(1).squeeze()

rgb = np.zeros( (out_pred.shape[0], out_pred.shape[1], 3))

for i in range(out_pred.shape[0]):
    for j in range(out_pred.shape[1]):
        rgb[i][j] = colors[out_pred[i][j]]

fig, ax = plt.subplots(1,2, figsize=(10,10))
ax[0].imshow(img)
ax[1].imshow(rgb)
fig.show()
```



응용: 우주의 고양이

```
img_bg = Image.open("/content/gdrive/My Drive/bg.jpg")  
img_bg = img_bg.resize( (img.width, img.height) )  
img_bg = np.array(img_bg)
```

```
img_np = np.array(img)
```

```
for i in range(out_pred.shape[0]):  
    for j in range(out_pred.shape[1]):  
        if out_pred[i][j] != 8:  
            img_np[i][j] = img_bg[i][j]
```

```
plt.figure(figsize=(12,12))  
plt.imshow(img_np)
```



Question?