

지난 시간에 yolov5로 학습시켜 나온 pt파일을 활용하여 인퍼런스 코드를 짜보자.

막상 해보니 작은 모델은 결과가 좋지 않아서 가능하면 med이상의 모델로 일정 이상의 epoch를 돌려야 잘나오는 것 확인 > s모델로 10epoch만 돌린 pt로 모델 돌렸을 땐 객체를 찾지 못해 결과가 나오지 않았다.

01. Yolov5_master 폴더 안에 Test.py 추가하고 best.pt 파일 모델로 설정하고 이미지 하나씩 선택하여 bbox 시각화한 인퍼런스 코드

```
import torch
import os
import glob
import cv2

### 모델 설정
model = torch.hub.load("ultralytics/yolov5", 'custom',
path='./runs/train/exp3/weights/yolov5m_best.pt')
model.cuda()
# print(model)

### label dict
label_dict = {
    0: 'cat',
    1: 'chicken',
    2: 'cow',
    3: 'dog',
    4: 'fox',
    5: 'goat',
    6: 'horse',
    7: 'person',
    8: 'raccoon',
    9: 'skunk'
}

### model config
model.conf = 0.5 # 0.5면 50% 이상 점수 받은 애만 박스 나옴
model.iou = 0.45 # NMS에서 사용되는 IoU(Intersection over Union) 임계값을 설정
# 테스트
img = "/animal_dataset/test/images/198.jpg.rf.6704e7741f24a73b4a76c33487425d6b.jpg"
image = cv2.imread(img)
result = model(img, size=224) # 학습돌린 이미지 사이즈(리사이징 포함)과 동일하게 넣자
#print(result) # 객체 검출 결과 나오는지
#print(result.xyxy[0]) # 박스 관련 정보가 나오는지
bboxes = result.xyxy[0]

for bbox in bboxes:
    x1, y1, x2, y2, conf, cls = bbox
    x1 = int(x1.item())# 텐서로 나와서 int로 변환
```

```

y1 = int(y1.item())
x2 = int(x2.item())
y2 = int(y2.item())
conf = conf.item()
cls = int(cls.item())
cls_name = label_dict[cls]
print(x1, y1, x2, y2, conf, cls)
#print(bbox)
image = cv2.rectangle(image, (x1,y1), (x2,y2), (0,255,0),2)

cv2.imshow("test", image)
if cv2.waitKey() & 0xFF == ord('q') :
    exit()

```

02. 모든 사진 돌아가며 시각화하여 확인하는 인퍼런스 코드

```

import torch
import os
import glob
import cv2

### 모델 설정
model = torch.hub.load("ultralytics/yolov5", 'custom',
path='./runs/train/exp3/weights/yolov5m_best.pt')
model.cuda()
#print(model)

### label dict
label_dict = {
    0: 'cat',
    1: 'chicken',
    2: 'cow',
    3: 'dog',
    4: 'fox',
    5: 'goat',
    6: 'horse',
    7: 'person',
    8: 'raccoon',
    9: 'skunk'
}

img_folder_path = './animal_dataset/test'
img_path = glob.glob(os.path.join(img_folder_path, '*', '*.jpg'))

for path in img_path:
    image = cv2.imread(path)
    results = model(path, size=640)

    bboxes = results.xyxy[0]
    for bbox in bboxes:
        x1, y1, x2, y2, conf, cls = bbox

```

```
x1 = int(x1.item())
y1 = int(y1.item())
x2 = int(x2.item())
y2 = int(y2.item())
conf = conf.item()
cls = int(cls.item())
cls_name = label_dict[cls]

print(x1, y1, x2, y2, cls_name)

image = cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.imshow("test", image)
if cv2.waitKey() & 0xFF == ord('q'):
    exit()
```

