

On_Device AI

https://github.com/planxlabs/toheaven/blob/main/SerBot_%EC%98%A8%EB%94%94%EB%B0%94%EC%9D%B4%EC%8A%A4%20AI.pdf

라벨링 1,0 0,1 -> softmax

전처리 - 흑백처리, 사이즈 조절(resize) <- 코딩시간 줄임

reshape(50,50,1) 50x50 : 사이즈 , 1 : 채널 수

output_size=2 True or False

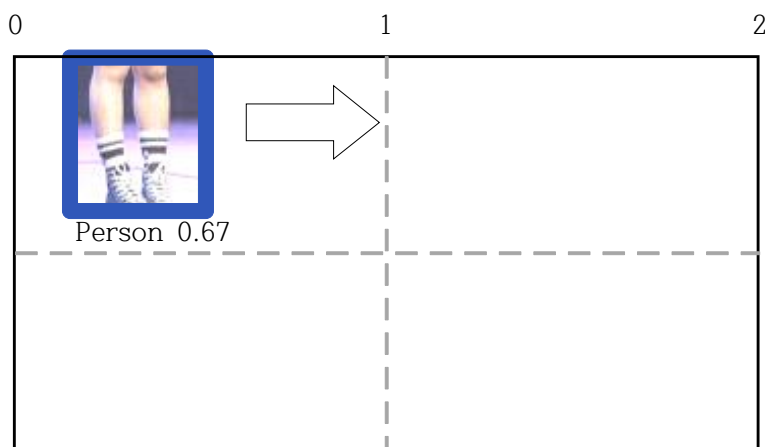
CNN.run([CNN.X_data[20]]) 반환값은 print써서 출력, 리스트형태로 나옴 (예측검증)

<https://www.kaggle.com/>

https://github.com/hanback-docs/Spoken_Digit_Dataset

https://github.com/hanback-docs/ssd_mobilenet_v2_coco_engine

.pth PyTorch파일



대상 위치가 1보다 작으면 오른쪽으로,
크면 왼쪽으로 이동하여
화면중앙(가로기준)에 대상이 위치하도록
이동한다

인식물의 크기를 통해 가까이 다가오는지 간접적으로 확인가능. 라이다를 같이 쓰면 좋음 (센서퓨전)

```
sudo ln -s /usr/local/lib/python3.6/dist-packages/pop/Pilot.py Pilot.py
```

#

1. 카메라에서 사람이 벗어나면 찾기
2. 라이다와 연동시키기

pop.Pilot 들어가기(ctrl+클릭)

Object_Follow 클래스 전체 복사해서 class Object_Follow2() 만듦

상속 지움 (주피터노트북에서 사용하던 것)

사용하지않는 멤버변수 삭제

```
cd/u/l/l/p/dist-/p (tab) -> pop폴더
```

```
cd model
```

```
ls -> yolov4-tiny, COCO_trt(텐서rt)
```

```
cd yolov4-tiny -> coco데이터셋(레이블셋) / yolov4-tiny.weights :모델파일
```

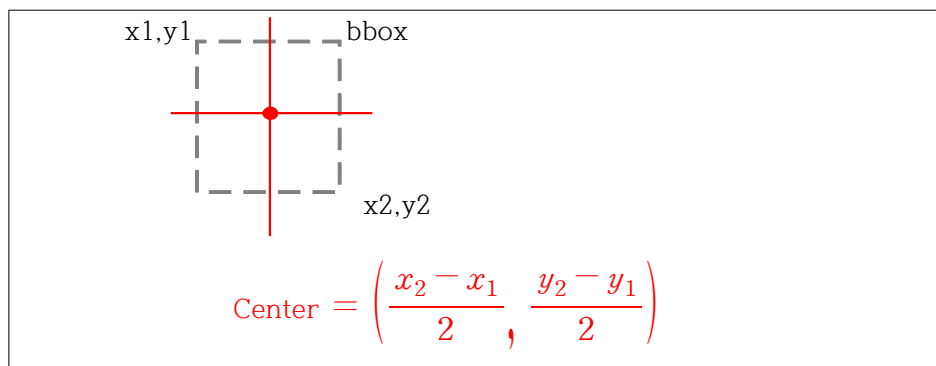
```
cat coco.names
```

_cat(카탈로그)...x

weights를 cuda에 배치하고, 경로에서 가져와 셋팅함 (메모리에 적재 후 옴로모델에 셋팅) - 시간 많이걸림

detect 중요함 (예측. 학습은 x)

소멸자 만들기 self.camera.release()



코랄, 모비디우스//

```
ssh soda@192.168.55.1
```

```
soda
```

```
sudo vi /etc/X11/xorg.conf
```

수정(아래 페이지에 있음)

```
sudo reboot
```

NoMachine -> 마우스를 화면 오른쪽 위로 올려둠 -> 접히는 모양 -> 클릭

setting

Quality -> Best quality

Resolution -> 1280x720



```

1 # Copyright (c) 2017, NVIDIA CORPORATION. All Rights Reserved.
2 #
3 # This is the minimal configuration necessary to use the Tegra driver.
4 # Please refer to the xorg.conf man page for more configuration
5 # options provided by the X server, including display-related options
6 # provided by RandR 1.2 and higher.
7
8 # Disable extensions not useful on Tegra.
9 Section "Module"
10 Disable      "dri"
11 SubSection   "extmod"
12 |Option      "omit xfree86-dga"
13 EndSubSection
14 EndSection
15
16 Section "Device"
17 Identifier   "Tegra0"
18 Driver       "nvidia"
19 Option       "AllowEmptyInitialConfiguration" "true"
20 EndSection
21
22 Section "ServerLayout"
23 Identifier   "ServerLayout0"
24 Option       "StandbyTime" "0"
25 Option       "SuspendTime" "0"
26 Option       "OffTime"     "0"
27 Option       "BlankTime"   "0"
28 EndSection
29
30 Section "Screen"
31 Identifier   "Screen0"
32 Monitor      "Monitor0"
33 SubSection   "Display"
34 |Viewport    0 0
35 |Depth       24
36 |Virtual     1280 720
37 |Modes       "1280x720"
38 EndSubSection
39 EndSection

```

줄 삭제 - 'dd'
수정 - 'I'
저장 - ':x'

[배경 바꾸는 법]

```

cd
ls -l .fehbg           프레임버퍼에 이미지를 출력하는 공간
cat .fehbg
//samba로 홈디렉토리로 사진 이동 후, vi편집기로 fehbg파일 수정

```

[부팅시 자동실행]

```

cd .config
ls
cd openbox
ls
cat autostart
// chhn 이름(자동변경) / tint 작업표시줄 / ap.sh ap모드 / can can연결
// 아래에 (/usr/bin/python3 /home/soda/Project/python/example.py) 추가하면 자동 실행함.
// openbox, 절대경로.
// ex. (/usr/bin/python3 /home/soda/Project/python/ai/my.py) &
// & : 백그라운드에서 실행하라는 의미

```