Social Distance Analyzer

18 이민권, 20 문예훈

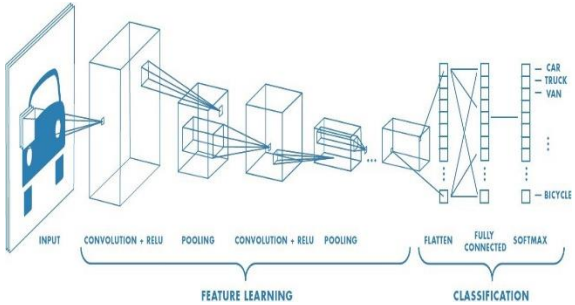
목차

- 우리들의 프로젝트
- 시연 영상
- 계기
- CNN이란?
- 필요한 모듈, 오픈소스 다운 방법
- 코드 설명
- 시연 영상 2
- 어려웠던 점, 한계
- 기대 효과

Our Project

Social Distancing Analyzer

사회적 거리두기



CNN, OpenCV, etc..

시연 영상

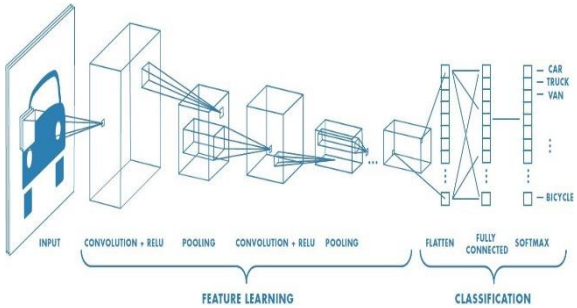

Social Distancing Analyser wrt. COVID-19
Connecting lines shows closeness among people.
YELLOW: CLOSE
RED: VERY CLOSE
Bounding box shows the level of risk to the person.
ORANGE: LOW RISK
GREEN: SAFE
TOTAL COUNT: 8
SAFE COUNT: 0
LOW RISK COUNT: 1

시작하게 된 계기

- 코로나 바이러스로 인한 사회적 거리두기 시행
  - 하지만 잘 지켜지지 않고 있는 현실
  - 심리적 불안감 조성
- Social Distance 분석 모델
  - 한동대학교 안의 사회적 거리두기 실태
  - 거리유지를 판단하는 모델 분석 및 공부

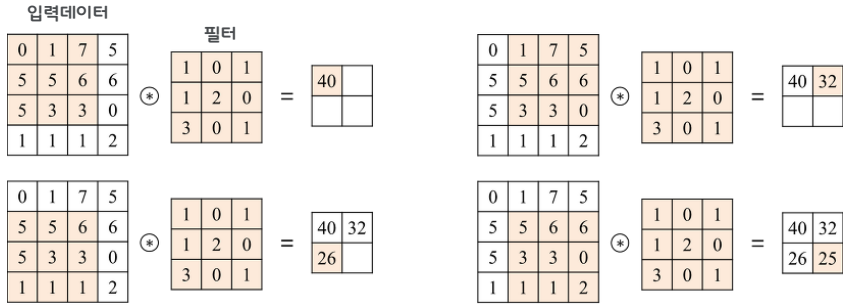
CNN (Convolutional Neuron Network)

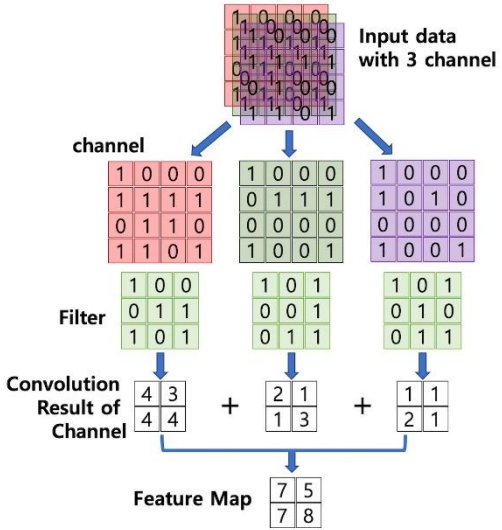
- 인간의 시신경을 모방하여 만든 딥러닝의 구조 중에 하나
- 다양한 이미지 분석 관련 작업
  - 장면 분류
  - 물체 감지 및 분할
  - ETC...
- Convolution Layer/Pooling Layer



Convolutional Layer

- **합성곱:** 반전 이동한 값과 두개의 서로 다른 함수를 곱한 다음 적분해 새로운 함수를 구하는 수학 연산자
- 필터를 이용해 이미지 특징을 추출





Convolutional Layer

- 이런 식으로 하다보면 이미지의 크기가 작아지면서 가장 자리의 픽셀의 정보를 얻기 힘들어지는 문제점 발생

- But, padding을 이용해 이미지의 가장자리에 특정 값의 픽셀을 추가해 입력 이미지와 출력 이미지의 크기에 큰 차이가 없게 만들어 해결

0	0	0	0	0	0
0	0	1	7	5	0
0	5	5	6	6	0
0	5	3	3	0	0
0	1	1	1	2	0
0	0	0	0	0	0

$\circledast$

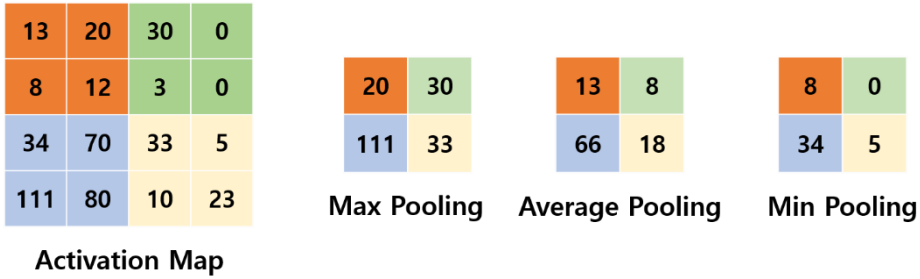
1	0	0
1	2	1
1	2	3

$=$

26	42	55	35
34	41	33	28
18	25	23	14
3	9	8	8

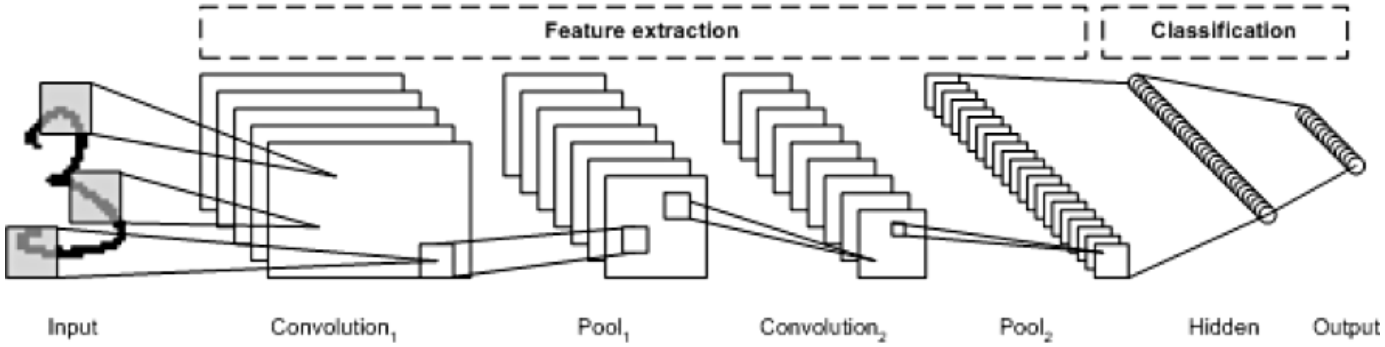
Pooling Layer

- 핵심적인 데이터 추출/강조 및 출력 데이터의 크기 축소
- Max/ Min/ Average Pooling



요약

- Convolution Layer: 이미지 특징 추출
- Pooling Layer: 이미지의 특징 강화 및 크기 축소
- 이러한 과정을 반복해 결과값을 출력



필요한 모듈 및 오픈소스

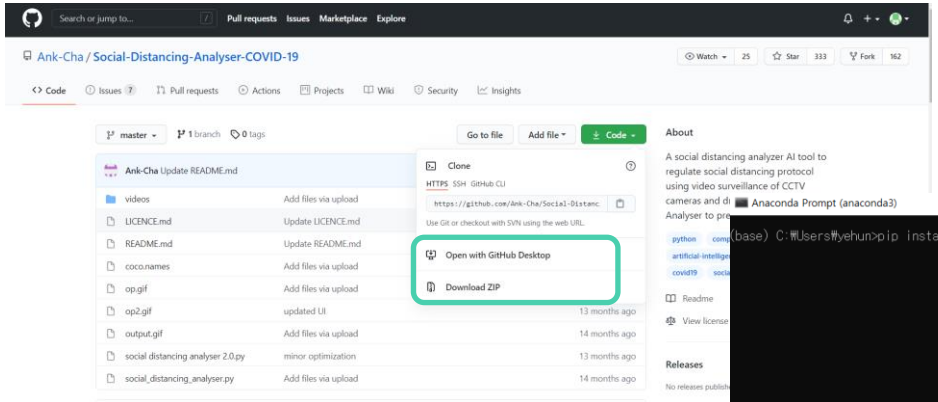
- 메인 모델
- yolov3.weight
- yolov3.cfg
- OpenCV

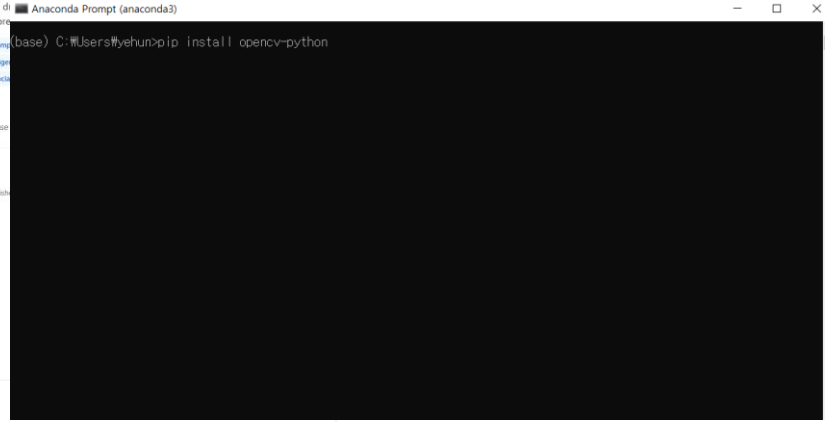
yolo는 Darknet에서 제공된 신경망 가중치 값이다

다운 링크

https://github.com/Ank-Cha/Social-Distancing-Analyser-COVID-19.git

다운 방법





Installation:

- Fork the repository and download the code.
- Download the following files and place it in the same directory
  - https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg
  - https://pjreddie.com/media/files/yolov3.weights
- For slower CPUs, use yolov3-tiny (link in the code comments)
- Install all the dependenices
- Run social_distancing_analyser.py or social_distancing_analyser 2.0.py

코드 설명

In [1]:
```
import time
import cv2
import numpy as np
```

→ 모듈 불러오기

In [3]:
```
#Loading Input Video
confid = 0.5
thresh = 0.5
vname=input("Video name in videos folder:  ")
if(vname==""):
    vname="video1.mp4"
vid_path = "./videos/"+vname
```

→ Input video를 지정 경로에서 불러오기

코드 설명

In [5]:
```python
#거리 계산
def calibrated_dist(p1, p2):
    return ((p1[0] - p2[0]) ** 2 + 550 / ((p1[1] + p2[1]) / 2) * (p1[1] - p2[1]) ** 2) ** 0.5
```

→ 거리 계산 함수

In [6]:
```python
#사람간의 거리 가까운지 판단
def isclose(p1, p2):
    c_d = calibrated_dist(p1, p2)
    calib = (p1[1] + p2[1]) / 2
    if 0 < c_d < 0.15 * calib:
        return 1
    elif 0 < c_d < 0.2 * calib:
        return 2
    else:
        return 0
```

→ 거리 판정 함수

코드 설명

In [7]:

```
#사물을 인식했을 때 붙이는 이름 파일 불러오기
labelsPath = "./coco.names"
LABELS = open(labelsPath).read().strip().split("₩n")

np.random.seed(42)

weightsPath = "./yolov3.weights"
configPath = "./yolov3.cfg"

# CNN learning --> making boxes
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

#video 경로에서 video를 캡처해서 불러옴
vs = cv2.VideoCapture(vid_path)
writer = None
(W, H) = (None, None)
```

→ 객체들의 리스트 불러오기

→ ‘Darknet’의 가중치와 신경망 구조 불러오기

→ Video를 학습할 opencv 신경망 생성

→ 불러온 video를 opencv에서 사용하기 위한 method

```
fl = 0
q = 0
while True:

    (grabbed, frame) = vs.read()     #불러온 video를 frame으로 나눔

    if not grabbed:
        break

    if W is None or H is None:
        (H, W) = frame.shape[:2]
        q = W

    frame = frame[0:H, 200:q]
    (H, W) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),       #이미지 frame 속 객체를 뽑아냄
                                 swapRB=True, crop=False)
    net.setInput(blob)                                               #객체를 신경망 input으로 넣어줌
    start = time.time()
    layerOutputs = net.forward(ln)                                   #순방향 / ouput = layer
    end = time.time()

    boxes = []
    confidences = []
    classIDs = []

    for output in layerOutputs:

        for detection in output:

            scores = detection[5:]
            classID = np.argmax(scores)              #layer 중 가장 큰 값의 index
            confidence = scores[classID]
            if LABELS[classID] == "person":          #인식한 사람이 '사람'일 경우

                if confidence > confid:                                      #'사람' 객체에 씌울 박스 생성
                    box = detection[0:4] * np.array([W, H, W, H])
                    (centerX, centerY, width, height) = box.astype("int")

                    x = int(centerX - (width / 2))
                    y = int(centerY - (height / 2))

                    boxes.append([x, y, int(width), int(height)])
                    confidences.append(float(confidence))
                    classIDs.append(classID)

    idxs = cv2.dnn.NMSBoxes(boxes, confidences, confid, thresh)     #박스들 값 저장(좌상단 우하단)
                                                                    #output은 진짜 필요한 Bounding box의 indexes를 반환한다
```

→ Video frame 별로 나눠줌

→ 포착한 사물(객체) 뽑아냄

→ 객체들로 신경망 학습

→ 객체가 '사람 ' 일 때, 사람 위치에 씌울 '박스' 좌표 생성

```
if len(idxs) > 0:

    status = list()
    idf = idxs.flatten()
    close_pair = list()
    s_close_pair = list()
    center = list()
    dist = list()
    for i in idf:
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])
        center.append([int(x + w / 2), int(y + h / 2)])

        status.append(0)
    for i in range(len(center)):
        for j in range(len(center)):
            g = isclose(center[i], center[j])

            if g == 1:

                close_pair.append([center[i], center[j]])
                status[i] = 1
                status[j] = 1
            elif g == 2:
                s_close_pair.append([center[i], center[j]])
                if status[i] != 1:
                    status[i] = 2
                if status[j] != 1:
                    status[j] = 2

    total_p = len(center)
    low_risk_p = status.count(2)
    high_risk_p = status.count(1)
    safe_p = status.count(0)
    kk = 0
```

→ 조건문을 통해 사람들의 거리를 계산하고 판정

→ 사람 위치에 씌워질 박스 중간값 간의 연산

→ High risk / Low risk / Safe 로 나누어 판정

```
for i in idf:                                                    #박스간의 선을 이어주는 작업

    sub_img = frame[10:170, 10:W - 10]
    black_rect = np.ones(sub_img.shape, dtype=np.uint8) * 0

    res = cv2.addWeighted(sub_img, 0.77, black_rect, 0.23, 1.0)

    frame[10:170, 10:W - 10] = res

    cv2.putText(frame, "Social Distancing Analyser wrt. COVID-19", (210, 45),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
    cv2.rectangle(frame, (20, 60), (510, 160), (170, 170, 170), 2)
    cv2.putText(frame, "Connecting lines shows closeness among people. ", (30, 80),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 0), 1)
    cv2.putText(frame, "-- YELLOW: CLOSE", (50, 110),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 255), 1)
    cv2.putText(frame, "--    RED: VERY CLOSE", (50, 130),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1)

    cv2.rectangle(frame, (535, 60), (W - 20, 160), (170, 170, 170), 2)
    cv2.putText(frame, "Bounding box shows the level of risk to the person.", (545, 80),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 0), 1)
    cv2.putText(frame, "-- DARK RED: HIGH RISK", (565, 110),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 150), 1)
    cv2.putText(frame, "--   ORANGE: LOW RISK", (565, 130),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 120, 255), 1)

    cv2.putText(frame, "--    GREEN: SAFE", (565, 150),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)

    tot_str = "TOTAL COUNT: " + str(total_p)                     #risk count
    high_str = "HIGH RISK COUNT: " + str(high_risk_p)
    low_str = "LOW RISK COUNT: " + str(low_risk_p)
    safe_str = "SAFE COUNT: " + str(safe_p)

    sub_img = frame[H - 120:H, 0:210]
    black_rect = np.ones(sub_img.shape, dtype=np.uint8) * 0

    res = cv2.addWeighted(sub_img, 0.8, black_rect, 0.2, 1.0)

    frame[H - 120:H, 0:210] = res
```

```
        cv2.putText(frame, tot_str, (10, H - 90),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 1)
        cv2.putText(frame, safe_str, (10, H - 65),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 1)
        cv2.putText(frame, low_str, (10, H - 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 120, 255), 1)
        cv2.putText(frame, high_str, (10, H - 15),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 150), 1)

        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])
        if status[kk] == 1:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 150), 2)

        elif status[kk] == 0:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        else:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 120, 255), 2)

        kk += 1
    for h in close_pair:
        cv2.line(frame, tuple(h[0]), tuple(h[1]), (0, 0, 255), 2)
    for b in s_close_pair:
        cv2.line(frame, tuple(b[0]), tuple(b[1]), (0, 255, 255), 2)

    cv2.imshow('Social distancing analyser', frame)
    cv2.waitKey(1)
```

→박스와 선 생성 / risk 텍스트로 표시

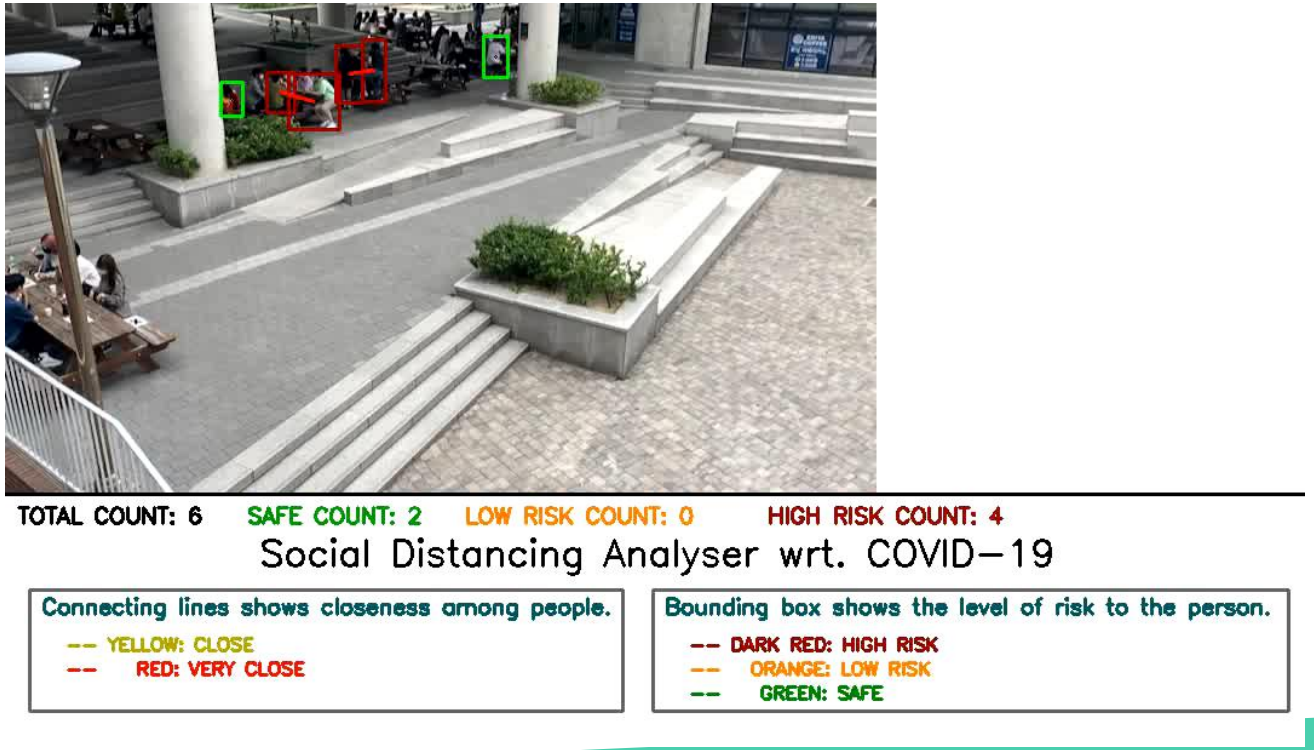
코드 설명

```
    writer.write(frame)
print("Processing finished: open output.mp4")
writer.release()
vs.release()
```

```
Processing finished: open output.mp4
```

→ 출력 영상(mp4 or avi)

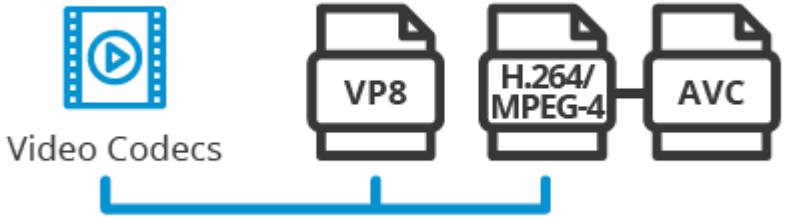
시연 영상2


TOTAL COUNT: 6
SAFE COUNT: 2
LOW RISK COUNT: 0
HIGH RISK COUNT: 4
Social Distancing Analyser wrt. COVID-19
Connecting lines shows closeness among people.
-- YELLOW: CLOSE
-- RED: VERY CLOSE
Bounding box shows the level of risk to the person.
-- DARK RED: HIGH RISK
-- ORANGE: LOW RISK
-- GREEN: SAFE

한계와 어려웠던 점

- 사람들이 인식되는 기준?
  - ➢ 혼동 될 만한 요소가 있을 시 인식 어려움

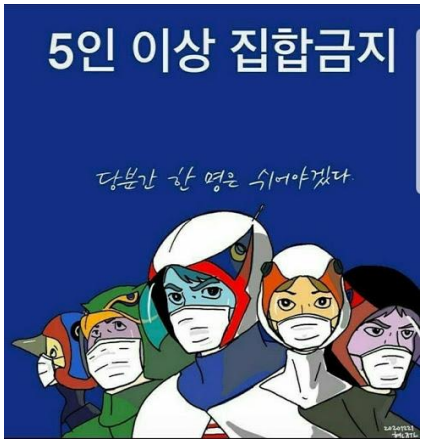
- 출력 영상의 format 과 codec 문제



기대효과

- 5인 이상 모임에게 경고 및 집합 방지

- 실시간 거리 판정 ← CCTV



Reference

- https://github.com/Ank-Cha/Social-Distancing-Analyser-COVID-19
- https://junha1125.github.io/blog/artificial-intelligence/2020-08-19-YOLO_OpenCV_DNN/
- https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg
- https://ko.wikipedia.org/wiki/%ED%95%A9%EC%84%B1%EA%B3%B1
- https://sungwookkang.com/1408

감사합니다 :3