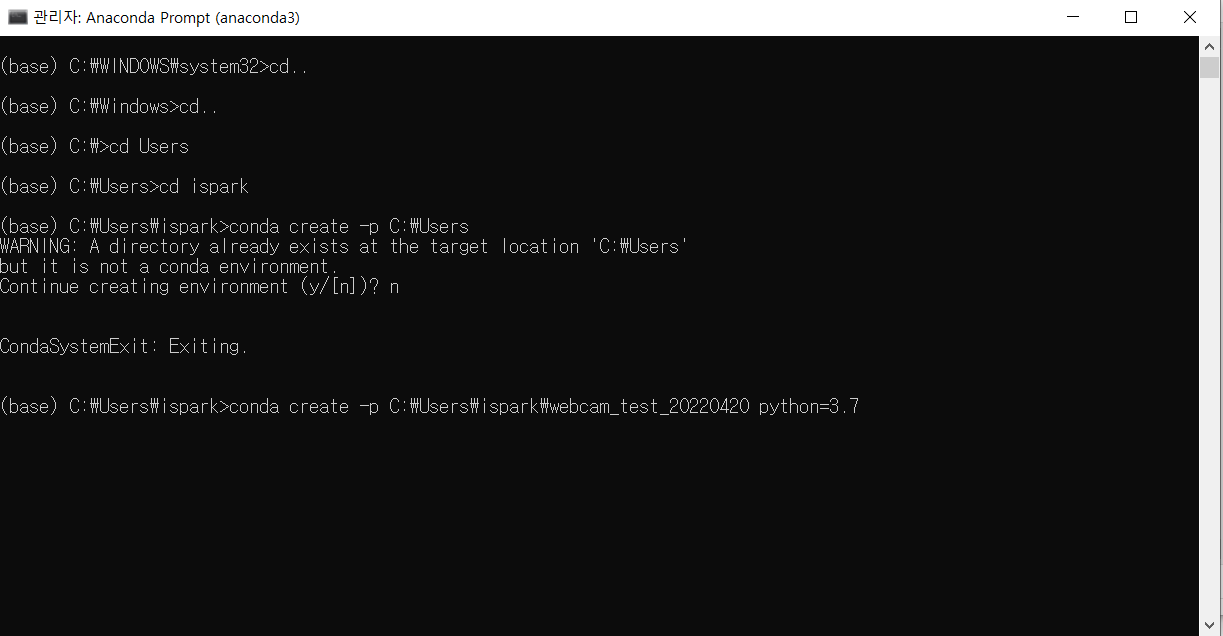
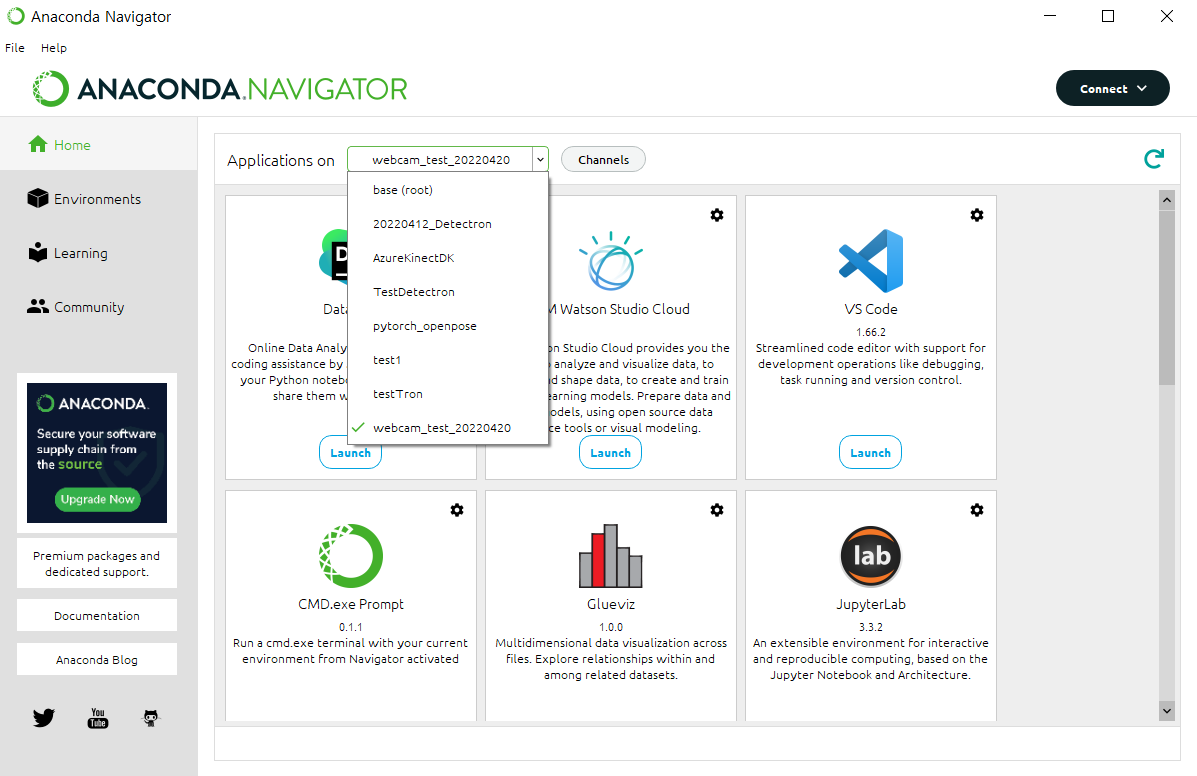
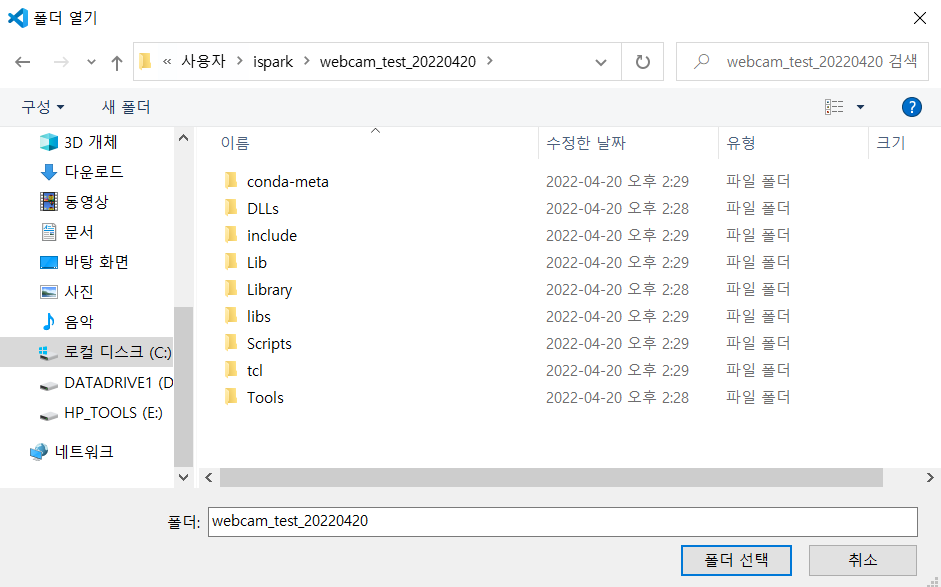
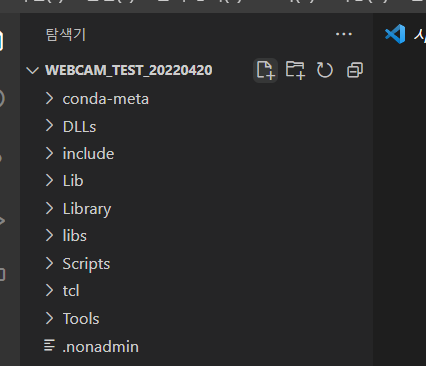
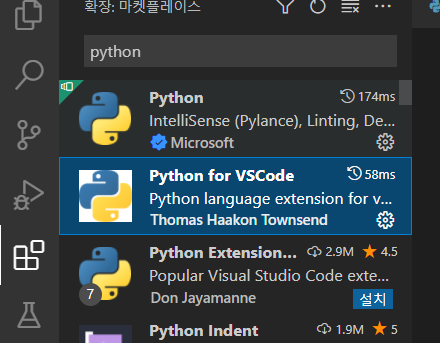
1. **Anaconda prompt로 작업환경만들기**Anaconda prompt를 관리자 권한으로 실행 후 다음과 같이 입력합니다
2. **ANACONDA.NAVIGATOR 실행**다음과 같이 생성된 환경으로 접속하고 VS Code를 실행합니다
3. **VS Code에서 폴더 열고 코드 작성  
   **왼쪽 상단의 파일 메뉴에서 폴더 열기 메뉴를 선택해 폴더 이동 후 선택합니다  
   ****   
   위의 그림과 같이 누르고 webcam\_PoseEsitimation\_to\_csv.py를 입력합니다  
   그 후 다음과 같은 코드를 복사해서 붙여넣기 합니다  
   또한 python 확장팩이 설치 안되어 있는 경우 다음과 같이 이동해 설치를 진행 해주시면 됩니다

from \_\_future\_\_ import division

import argparse, time, logging, os, math, tqdm, cv2

from pyparsing import delimited\_list, delimitedList

import numpy as np

import mxnet as mx

from mxnet import gluon, image, ndarray\_doc

from mxnet import ndarray as nd

from mxnet.gluon.data.vision import transforms

import pandas as pd

#import matplotlib.pyplot as plt

import gluoncv as gcv

from gluoncv import data

from gluoncv.data import mscoco

from gluoncv.model\_zoo import get\_model

from gluoncv.data.transforms.pose import detector\_to\_simple\_pose, heatmap\_to\_coord

from gluoncv.utils.viz import cv\_plot\_image, cv\_plot\_keypoints

import matplotlib.pyplot as plt

ctx = mx.cpu()

detector\_name = "yolo3\_mobilenet1.0\_coco"

detector = get\_model(detector\_name, pretrained=True, ctx=ctx)

detector.reset\_class(classes=['person'], reuse\_weights={'person':'person'})

detector.hybridize()

estimators = get\_model('simple\_pose\_resnet18\_v1b', pretrained='ccd24037', ctx=ctx)

estimators.hybridize()

cap = cv2.VideoCapture(0)

time.sleep(1)  ### letting the camera autofocus

axes = None

num\_frames = 100

#pred\_coords\_2d=np.zeros((0,34))

pred\_coords\_2d =np.empty((1,34), float)

sample\_num = 0

captured\_num = 0

for i in range(num\_frames):

    ret, frame = cap.read()

    sample\_num = sample\_num + 1

    if not ret:

        break

    if sample\_num == 1:

        captured\_num = captured\_num + 1

        cv2.imwrite('./images/img'+str(captured\_num)+'.jpg',frame)

        sample\_num = 0

    inversed = ~frame

    frame = mx.nd.array(cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)).astype('uint8')

    x, frame = gcv.data.transforms.presets.ssd.transform\_test(frame, short=512, max\_size=350)

    x = x.as\_in\_context(ctx)

    class\_IDs, scores, bounding\_boxs = detector(x)

    pose\_input, upscale\_bbox = detector\_to\_simple\_pose(frame, class\_IDs, scores, bounding\_boxs,

                                                       output\_shape=(128, 96), ctx=ctx)

    if len(upscale\_bbox) > 0:

        predicted\_heatmap = estimators(pose\_input)

        pred\_coords, confidence = heatmap\_to\_coord(predicted\_heatmap, upscale\_bbox)

        img = cv\_plot\_keypoints(frame, pred\_coords, confidence, class\_IDs, bounding\_boxs, scores,

                                box\_thresh=0.5, keypoint\_thresh=0.2)

        print('     3150 pred\_coords type=',type(pred\_coords))

        print('     3152 pred\_coords ndim=',pred\_coords.ndim)

        print('     3154 pred\_coords shape=',pred\_coords.shape)

        #pred\_coords\_trans = nd.array([-1])

        #for \_ in range(31):

        #pred\_coords\_trans = pred\_coords.expand\_dims(axis=0)

        #pred\_coords\_trans1 = pred\_coords\_trans

        print(pred\_coords)

        print('     3170 pred\_coords\_2d\_1 shape=',type(pred\_coords))

        pred\_coords\_2d\_1 = pred\_coords.reshape(pred\_coords.shape[0], -1)

        np\_ex\_float\_array = pred\_coords\_2d\_1.asnumpy()

        print('     3170 pred\_coords\_2d\_1 shape=',np\_ex\_float\_array.shape)

        pred\_coords\_2d = np.append(pred\_coords\_2d , np\_ex\_float\_array, axis = 0)

        #print('     3190 pred\_coords\_2d\_1 shape=',pred\_coords\_2d\_1.shape)

        #print('     3192 pred\_coords\_2d shape=',pred\_coords\_2d.shape)

        #pred\_coords\_2d = np.append(pred\_coords\_2d,pred\_coords\_2d\_1, axis=0)

        #print('     3190 pred\_coords\_2d shape=',type(pred\_coords\_2d))

        #pred\_coords\_3d = pred\_coords.reshape(pred\_coords.shape[2], -1)

        #print('     3154 pred\_coords\_2d shape=',pred\_coords\_2d.shape)

        #print(pred\_coords\_2d[:1,:1])

        #print(pred\_coords\_2d)

        #print(pred\_coords\_3d)

        #df = pd.DataFrame(list(pred\_coords\_2d[:1,:1]))

        #df.to\_csv("test2.csv",header='col',index=None)

        #np.savetxt('TopCoatMain\_bb.csv', list(pred\_coords\_3d), delimiter=',')

        df = pd.DataFrame(pred\_coords\_2d)

        df.columns = ['nose\_x', 'nose\_y','left\_eye\_x','left\_eye\_y','right\_eye\_x', 'right\_eye\_y','left\_ear\_x', 'left\_ear\_y','right\_ear\_x', 'right\_ear\_y','left\_shoulder\_x', 'left\_shoulder\_y','right\_shoulder\_x', 'right\_shoulder\_y','left\_elbow\_x', 'left\_elbow\_y','right\_elbow\_x', 'right\_elbow\_y','left\_wrist\_x', 'left\_wrist\_y','right\_wrist\_x', 'right\_wrist\_y','left\_hip\_x', 'left\_hip\_y','right\_hip\_x', 'right\_hip\_y','left\_knee\_x', 'left\_knee\_y','right\_knee\_x', 'right\_knee\_y','left\_ankle\_x', 'left\_ankle\_y','right\_ankle\_x', 'right\_ankle\_y']

        df.to\_excel('data.xlsx')

        csv\_data = pd.read\_excel('data.xlsx')

        #fourcc = cv2.VideoWriter\_fourcc('m','p','4','v')

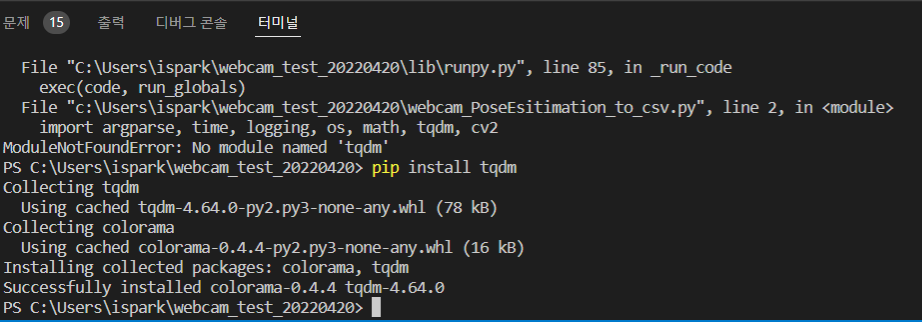
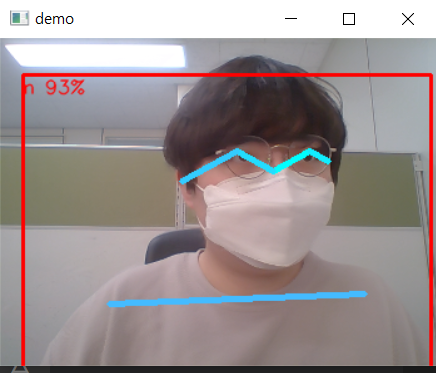
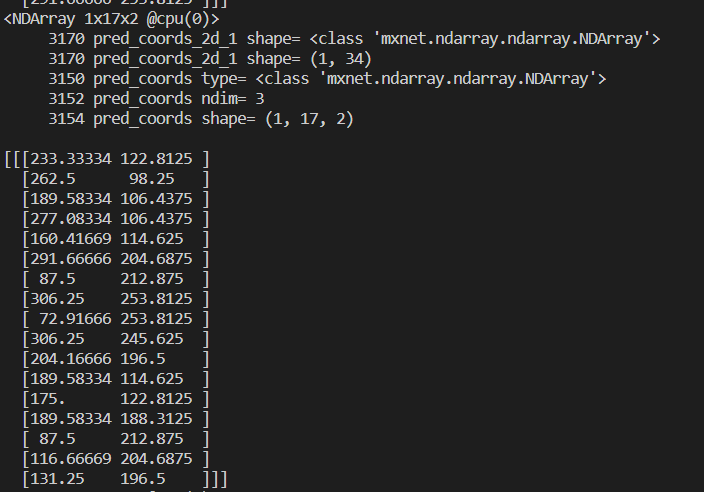
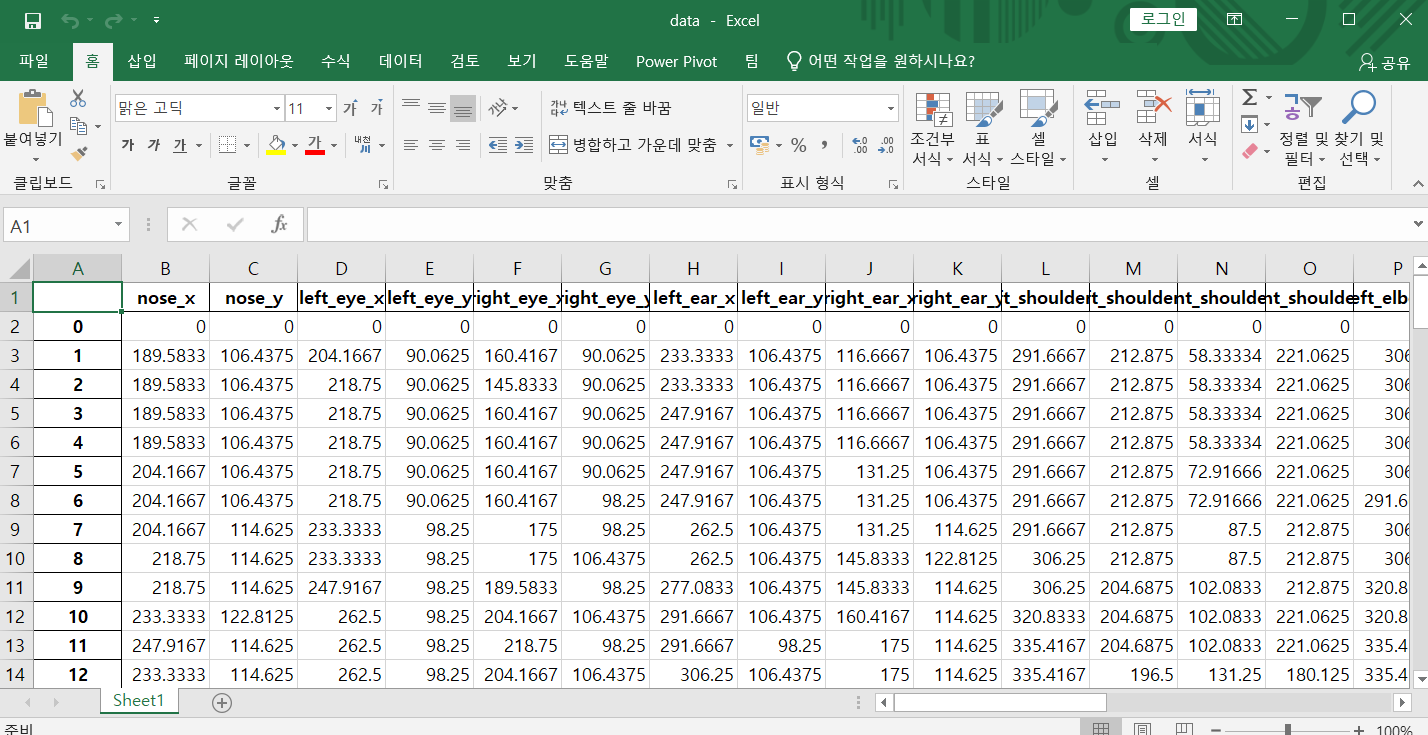
        #delay = round(1000/fps)

        #out = cv2.VideoWriter('SaveVideo.mp4',fourcc,fps,(640, 480))

    cv\_plot\_image(img)

    cv2.waitKey(1)

cap.release()

1. **모듈설치**해당 코드를 F5로 실행할 경우 모듈들이 존재하지 않음으로 다음과 같이 터미널에 입력합니다  
   pip install tqdm   
   pip install opencv-python  
   pip install pyparsing  
   pip install mxnet  
   pip install pandas  
   pip install gluoncv  
   pip install –upgrade pip  
   pip install openpyxl
2. **실행  
   **실행에 성공하면 다음과 같이 webcam이 보이고 ****  
   좌표 값 들이 터미널에 찍히게 됩니다  
   해당 작업들은 100frame을 채운 후 자동 종료 되며 data.xlsx 파일이 폴더에 생성됩니다  
   ****  
   폴더 내에 images 폴더를 만들어 놨다면 각 프레임 별 이미지가 생성되어 있습니다  
   **실행이 되지 않는 경우 webcam에 사람을 인식 못할 경우 프로그램이 종료 될 수 있습니다**[**https://cv.gluon.ai/build/examples\_pose/cam\_demo.html**](https://cv.gluon.ai/build/examples_pose/cam_demo.html) **원본코드**