

OpenCV

1일차 보고서

목차

1. 코드 분석
2. 실행 결과
3. 참고 문헌

1. 코드 분석

****server.py****

Import socket

- 해당 코드를 통해 Python 표준 라이브러리에 포함된 socket 모듈을 사용하여 UDP 소켓을 이용가능하도록 설정

Import numpy

- 이미지를 주고받으므로 배열과 연산을 위해 numpy를 이용 가능하도록 설정

Import cv2

- openCV를 이용하기위해 cv2 모듈 설정
- 이를 통해 이미지 읽기, 쓰기, 변환 및 다양한 영상 처리 작업 가능

UDP_IP = ~~ , UDP_PORT = ~~

- UDP 통신을 위한 IP와 PORT를 설정

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

- 소켓을 생성할 때 가장 먼저 지정해야하는 주소 패밀리 설정
- AF_INET: IPv4 주소 체계
- SOCK_DGRAM: UDP 프로토콜을 기반으로 하는 비연결형 방식

sock.bind(UDP_IP, UDP_PORT)

- 소켓에 IP와 PORT를 바인딩

data, addr = sock.recvfrom(65535)

- 버퍼 공간을 데이터를 전송할 수 있는 최대 크기인 65535byte로 설정

Photo = numpy.fromstring(data, dtype=numpy.uint8)

- 받은 바이트 데이터를 numpy배열로 저장(type은 uint8)

Photo_2 = cv2.imdecode(photo, cv2.IMREAD_COLOR)

- numpy배열로 변환된 사진을 디코딩하여 OpenCV 이미지 형식으로 변환

Cv2.imshow("띄울 창 이름", 띄울 사진)

- imshow를 통해 전송 받은 사진 화면에 출력

Cv2.waitKey(0)

- 키 입력을 받기를 기다려 사진이 출력되자마자 꺼짐 방지

Cv2.destroyAllWindows()

- 열린 모든 창을 닫아 종료

****client.py****

Import socket

- 해당 코드를 통해 Python 표준 라이브러리에 포함된 socket 모듈을 사용하여 UDP 소켓을 이용가능하도록 설정

Import numpy

- 이미지를 주고받으므로 배열과 연산을 위해 numpy를 이용 가능하도록 설정

UDP_IP = ~~ , UDP_PORT = ~~

- UDP 통신을 위한 IP와 PORT를 설정

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

- 소켓을 생성할 때 가장 먼저 지정해야하는 주소 패밀리 설정
- AF_INET: IPv4 주소 체계
- SOCK_DGRAM: UDP 프로토콜을 기반으로 하는 비연결형 방식

Photo = cv2.imread("/home/kmj/Desktop/hw2.jpg")

- 절대경로로 수신할 사진을 읽어와 변수 photo에 저장

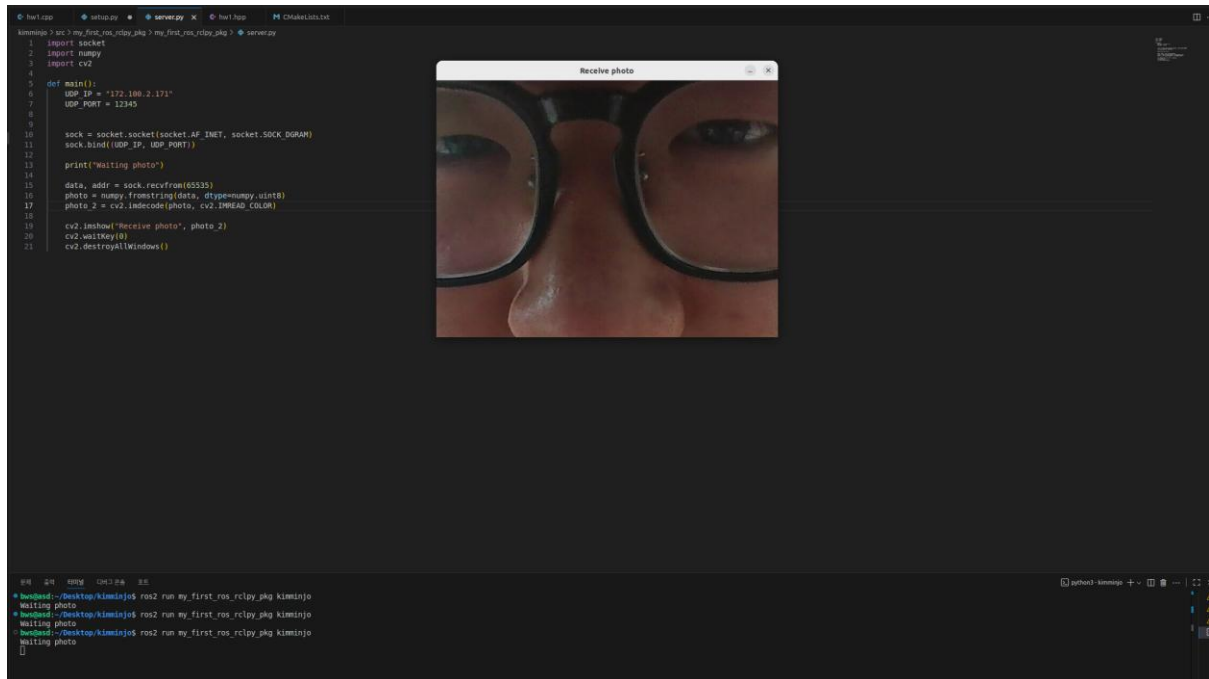
Photo_2 = cv2.imencode(".jpg", photo)[1].tobytes()

- photo_2라는 변수는 bytes타입을 가지게 됨
- 변수 photo를 jpg형식으로 인코딩

Sock.sendto(보낼데이터, (IP, PORT))

- sendto로 server에 사진 보내기

2. 실행 결과



- 상대방의 노트북에서 사진을 전송 받은 모습
- (조로 정해진 강은구 인턴이 수업으로 인해 같이 못하는 상황이라 부득이하게 현창석 인턴과 과제를 진행하였습니다.)

3. 참고 문헌

[\[Python\] OpenCV, UDP를 이용해 Webcam 영상 데이터를 서버로 보내기 :: awakening95](#)

[Python : 이미지를 저장하지 않고 다루기 \(OpenCV, Numpy\)](#)