

HW2 보고서

로봇 20기 인턴 모시온

2025407006 로봇학부

목차

1. 개요

- 1-1. 목표
- 1-2. 기본 개념
- 1-3. 요약

2. 코드 구성

- 2-1. 위젯(hpp)
- 2-2. 위젯(cpp)
- 2-3. 송신자
- 2-4. 수신자

3. 참조 링크

1. 개요

1-1. 목표

본 과제는 UDP를 통한 이미지 송수신 실습이 목표이다.

“업무 자동화와 파이썬 프로그래밍” 강의 내용인 인코딩 및 디코딩의 개념을 이용하여 opencv 제공함수 encode 및 decode를 사용하였다. 이미지를 jpeg 인코딩을 통하여 바이트 배열로 전환하며 이를 디코딩하여 복원하도록 하였다.

Qt를 기준으로 제작하므로 임의 IP, 포트, 사진들을 지정하여 전송할 수 있도록 구성하였다.

<지시사항에 따라 모든 문체는 개조식으로 구성>

1-2. 기본 개념

Encoding(인코딩)

인코딩은 정보를 다른 형태로 변환하는 과정을 지칭함

이미지를 OpenCV Mat 객체로 JPEG인코딩을 진행한다.

<이와 관련하여, 텍스트 인코딩의 UTF, ASCII, UNIX가 있다.>

Decoding(디코딩)

디코딩은 변환된 데이터를 원래의 형태로 복원하는 과정이다.

수신자가 지정 바이트 배열을 OpenCV Mat 객체로 변환.

해당 변환 객체를 다시 복원하여 이미지 수신 과정을 완료.

<UDP 통신상 송수신 가능한 정보는 65535바이트이다.>

1-3. 요약

ROS2 & Qt 환경과 UDP사용을 위해 지정 경로 위치 이미지를 인코딩 (JPEG) 후 바이트 배열(벡터)로 전송, 수신자는 수신 내용을 디코딩하여 QImage로 바이트 배열 정보에 따라 라벨에 출력.

2. 코드 구성

(코드 의도를 간단하게 요약하여 작성)

해당 위의 자료는 제가 수강하는 "업무 자동화와 파이썬 프로그래밍"의 png파일을 html로 저장한 후 이를 jpg로 encode한 결과를 16진수로 확인하는 내용.

구성 내용이 이미지 정보를 배열로서 좌측상단부터 우측 하단 순으로 정의됨을 알게되어 GUI를 다루는 QT 또한 동일 맥락일 것이라는 추측에서 시작.

UDP 통신은 상대방의 IP주소와 각각의 송신, 수신 포트가 개별적으로 필요.
네트워크의 불안정을 포함한 여러 요인으로 통신의 불안적을 고려.
이를 보완하여 고정된 포트 채널이 아닌 임의의 입력에 따른 수정가능하게 함.

송신자는 송신 사진의 정보를 알고 있을거라는 전제하에 송신 사진을 표시하는 부가기능은 추가하지 않음.

버튼 입력과 udp수신을 신호로 connect하여 함수 동작을 구성.

2-3. 송신자

사용자가 QLineEdit에 입력한 이미지 경로(path) 불러와 문자열로 변환.
OpenCV imread()로 Mat 객체로 행렬인 이미지 읽음
(이미지 없으면 함수 종료하여 송신 방지)
Mat 행렬을 JPEG 형태로 인코딩.
(imencode(".jpg",img,buf,compression_params) 사용
<참조: C++ JPG 압축>를 통해 JPEG 압축을 입력 이미지 절반으로 지정.
(바이트 배열이므로 int형 벡터로 품질 압축을 통해 용량 축소)
이를 다시 <uchar>의 QByteArray로 변환하여 Qt UDP 송신용 형식 준비.
(uchar는 0~ 255 값을 저장하므로 RGB 이미지에 적합.)
QHostAddress, tx_port로 전송대상 설정하여writeDatagram() 전송.
QPushButton 클릭에 따라 on_send_clicked() 슬롯 호출하도록 connect 연결.

2-4. 수신자

UDP 소켓이 바인딩된 rx_port에서 데이터 수신 계속 대기.
hasPendingDatagrams()로 수신 대기 중인 데이터그램 확인.
정보 수신이 확인될 시 이미지 출력을 위한 동작을 while의 판별문으로 설계.
이미지 출력을 위한 데이터그램 크기만큼 QByteArray(datagram) 버퍼 확보.
readDatagram()으로 수신된 데이터 Datagram을 버퍼에 읽어 저장.
QByteArray를 다시 <uchar>로 변환하여 OpenCV 디코딩을 준비.
imdecode(buf, IMREAD_COLOR)로 JPEG 바이트 배열을 Mat으로 복원.
복원된 Mat 이미지가 비어있지 않은지 확인하여 오류를 미연에 방지.
해당 이미지를 그대로 출력할 시 색감에 오류가 발생, 이를 보완하기 위하여
BGR -> RGB 변환(cvtColor)을 추가 수행하여 Qt에서 색상 정상 표시 완료.
Qimage (img.data, img.cols, img.rows, img.step, QImage::Format_RGB888).
위의 QImage로 8비트(0~255)의 값을 각각 지니는 픽셀들을 순서대로 저장.
(RGB888 포맷은 0~255의 값을 지니는 3채널 포맷을 뜻하여 이를 활용.)
해상도 및 크기는 제각각이므로 setPixmap을 통해 수신 이미지 조정.

(각자의 코드 제작과 일정에 따라 사전에 주어진 조의 구성으로 촬영 영상을 준비하지 못하였습니다.)

3. 참조 링크

.png 이미지 .jpg 인코딩 UDP 통신

<https://rootiel.tistory.com/entry/OPENCV-UDP-STREINGCPP>

<https://www.youtube.com/watch?v=lcp5t20INyQ>

UDP 통신에 대해 전송이 제한되는 이미지 크기

<https://stackoverflow.com/questions/1098897/what-is-the-largest-safe-udp-packet-size-on-the-internet>

QT와 UDP 복습

<https://doc.qt.io/qt-6/qudpsocket.html>

C++ JPG 압축

<https://products.aspose.com/words/ko/cpp/compress/jpg/>