

Open cv day 2\_hw1

20 기 인턴 우정윤

## 목차

1. OpenCV 개념 \_\_\_\_\_

2. 코드 구성 \_\_\_\_\_

3. 코드 설명 \_\_\_\_\_

3. 통신 실행 및 결과 \_\_\_\_\_

4. 참고 문헌 \_\_\_\_\_

## 1. Opencv 개념

### 1) OpenCV 란

OpenCV 는 open source computer vision 의 약자로 영상 처리에 사용할 수 있는 오픈 소스 라이브러리를 말한다. 컴퓨터가 사람의 눈처럼 인식할 수 있게 처리해주는 역할을 하기도 하며, 우리가 많이 사용하는 카메라 어플에서도 OpenCV 가 사용된다.

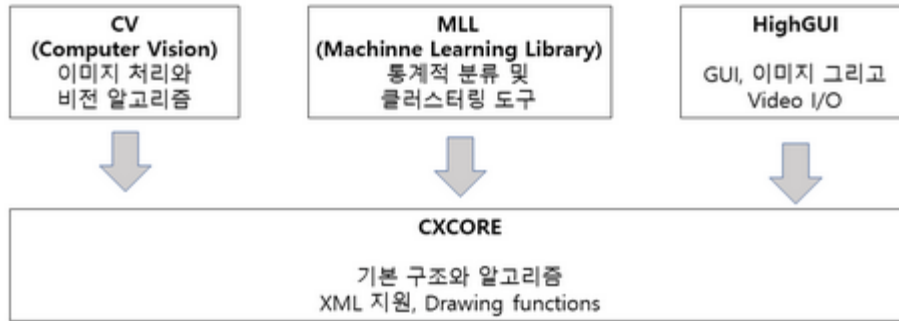
### 2) 컴퓨터 비전

컴퓨터 비전은 카메라로 얻은 일련의 영상들을 모두 컴퓨터 비전이라 한다. 컴퓨터는 숫자의 행렬로 사진 및 여러 형태를 인식하는데 빅데이터를 통해 컴퓨터에게 유의미한 정보를 입력해줌으로써 사람이 보는 시선과 최대한 비슷하게 만들어준다. 하지만 사람의 눈에도 왜곡이 있듯이 카메라에도 왜곡은 존재하기에 오차가 발생한다.

이 부분에 대한 해결책은 머신러닝 기술로 보완하고 있다. 혹은 추가적인 다른 데이터를 활용해서, 라이다 센서 혹은 거리 센서를 활용해서 왜곡정보를 올바르게 해석하기 위해 노력한다.

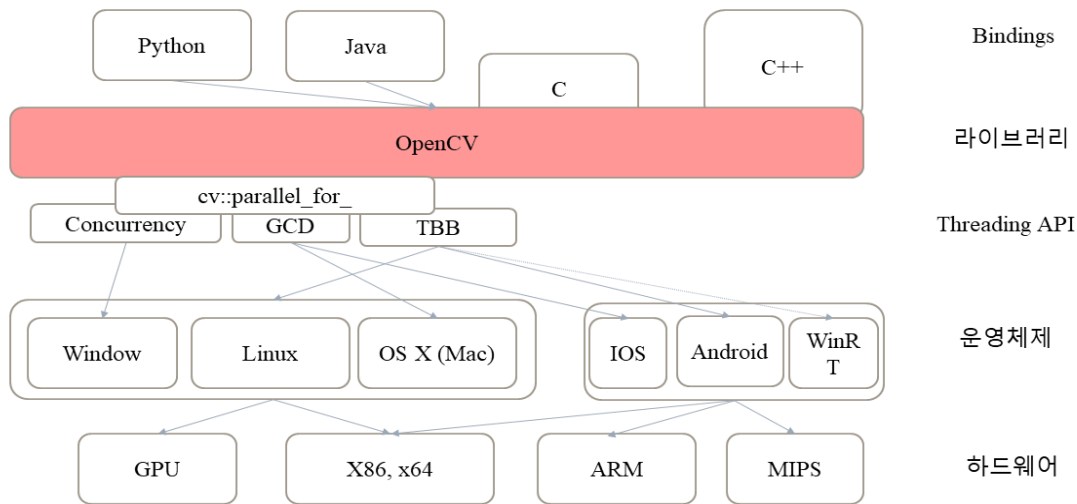
사용 가능한 언어로는 c, c++ 파이썬, 자바 등이 있으며 빅데이터와 머신 러닝에서 강점을 보이고 있는 파이썬이 많이 활용된다.

### 3) 기본구조



Opencv 의 기본 아키텍처는 다음과 같다. 각 요소를 살펴보자

- 1) Cv: computer vision 으로 기본 구성 요소에서는 기본 이미지 처리와 고급 컴퓨터 비전 알고리즘을 포함하고 있다.
- 2) Ml: 머신 러닝이 주를 이루므로 이를 활용할 수 있는 기본적인 통계 분류기와 클러스터링 도구를 포함하고 있는 라이브러리들이 있다.
- 3) Highgui : opencv 에 기본으로 포함되어 있는 gui 로 비디오와 이미지 저장 그리고 이를 로드하기 위한 i/o 루틴을 포함하고 있다. 그렇지만 더 범용적으로 이용하기 위해서 qt 를 화용한다.
- 4) Cxcore: 이곳에는 기본 데이터 구조와 여러 content 가 저장되어 있다.



2021-05-12

다른 관점에서 보자면 이러한 아키텍처 구조를 갖는다. 다양한 라이브러리의 모듈인데 모듈 구성은 다음과 같다.

- Core - 모든 객체 타입과 기본 연산자가 들어 있는 라이브러리 섹션
- Imgproc - 이미지 처리 모듈로써 이미지 변환 등을 포함하고 있다.
- highgui - 위에서 설명했다. (사용자의 입력을 받는 유저 인터페이스 기능 포함)
- video - 비디오 라이브러리로써 비디오 스트림을 read/write 하는 함수가 포함되어 있다.
- calib3d - 캘리브레이션하는데 필요한 알고리즘이 포함된 모듈
- features2d - 중요한 feature(특징)을 탐지하고 표현, 매칭하기 위한 알고리즘이 구현된 모듈
- objdetect - 말 그대로 오브젝트 디텍트, 특정 오브젝트를 탐지하는 알고리즘이 들어있는 모듈이다. 이를 머신 러닝으로 기계 학습 시켜서 특정 물체들을 탐지할 수 있다.
- ml - 머신러닝 라이브러리

- gpu - 그래픽 카드를 이용한 연산을 사용한다. 일명 그래픽카드 연산을 위한 라이브러리이다. CUDA 연산에 최적화 되어 있다. 당연히 머신러닝을 위한 설계이다. CUDA (Computed Unified Device Architecture)로써 NVIDIA 에서 개발한 GPU 개발툴이다. 딥러닝에서 사용하기 위해서 만들어졌으며, 하드웨어에서는 얻을 수 없는 연산을 할 때 많이 사용된다. 비트코인 채굴할 때 사용하는 것이 CUDA 이다. (tensorflow 도 사용된다.)
- photo - 사진을 활용할 수 있는 다양한 도구를 포함하고 있다.
- stitching - 스티치 파이프라인을 구현하기 위한 모듈, Image stitching 이란 이미지를 포깅 지점을 찾는 과정이라 생각하면 된다. 이를 빠르게 하는 파이프라인을 구현하기 위한 모듈이라 생각하면 된다.
- nonfree - OpenCV 에서 특허나 사용 제한이 걸려있는 알고리즘의 구현이 이곳에 있다. ( 예를 들어 SIFT 알고리즘)

#### 4) 함수정리

##### 1) 영상 읽기

- cvLoadImage 함수: 영상을 메모리 상에 로드하는데 사용되는 함수
- IplImage\* cvLoadImage( char \* filename, int iscolor );
- filename : 로드되는 파일의 이름을 가리킨다.
- iscolor : 로드되는 영상의 색상을 가리킨다.

iscolor > 0 이면 3 채널 컬러 영상으로 변환하여 로드한다.

iscolor = 0 이면 흑백 영상으로 변환하여 로드한다.

iscolor < 0 이면 원래의 영상 그대로 로드한다. 즉, 원래 영상이 3 채널 컬러영상은 컬러영상으로 흑백영상은 그대로 흑백영상으로 로드한다. 예를 들어, 다음은 Lena 컬러영상을 컬러영상 그대로 메모리 상에 로드한다.

```
IpLlImage* image;  
image = cvLoadImage("../images/Lena.jpg", -1);
```

- 만약에 컬러영상을 흑백영상으로 변환하여 로드하고자 하는 경우는 다음과 같다.

```
IpLlImage* image;  
image = cvLoadImage("../images/Lena.jpg", 0);
```

파일	설명
BMP, DIB	Windows 비트맵
JPEG, JPG, JPE	JPEG 파일
PNG	Portable Network Graphics
PBM, PGM, PPM	Portable 영상 포맷
SR, RAS	Sun rasters
TIFF, TIF	TIFF 파일

다음은 opencv 에서 지원되는 파일의 종류이다.

## 2) 영상 출력

- cvNamedWindow 함수: 영상을 출력하기 위한 윈도우를 만들기 위해 사용되는 함수이다.

```
int cvNamedWindow(char* name, int flags;
```

- name : 윈도우의 이름을 나타낸다.



- flags : 윈도우의 크기조절을 나타내기 위한 플래그이다.

플래그가 CV\_WINDOW\_AUTOSIZE 이면 윈도우의 크기는 자동으로 출력할 영상의 크기에 맞춘다. CV\_WINDOW\_AUTOSIZE 대신 1 을 사용할 수 있다. 이 경우 출력된 윈도우의 크기를 사용자가 마우스를 가지고 크기 조절을 할 수 없다. 플래그가 0 인 경우 사용자가 마우스로 크기 조절을 할 수 있다.

- cvShowImage 함수: cvNamedWindow() 함수에 의해 지정된 윈도우에 영상을 출력하기 위해 사용하는 함수이다.

```
void cvShowImage(char* name, CvArr* image);
```

- name: 윈도우의 이름이며 cvNamedWindow() 함수에 의해 지정된 이름과 동일하다.

- Image: 출력할 영상을 나타내며 cvLoadImage() 함수에 의해 로드된 영상을 지정한다.

- cvWaitKey 함수 : 키보드로부터 키 입력을 기다리기 위한 함수이다.

```
int cvWaitKey(int delay);
```

- delay: 지정된 시간(단위 : milliseconds) 동안 기다린다.

cvWaitKey() 함수는 영상을 화면에 출력할 때 출력되는 시간을 지정하기 위해 사용한다. delay ≤ 0 인 경우 예를 들어, cvWaitKey(0)이면 영상이 출력된 상태에서 무한히 키보드 입력을 기다리다 키를 입력하면 입력된 키의 코드를 반환한다. cvWaitKey(1000)은 1 초 동안 기다리고 cvWaitKey(10000)이면 10 초 동안 기다린다. 만약 지정된 시간동안 키보드로부터 사용자가 키를 입력하지 않으면 -1 을 반환한다.

- cvReleaseImage 함수: cvLoadImage() 함수를 이용하여 메모리에 로드된 영상이 메모리로부터 해제하기 위해 사용하는 함수이다.

```
void cvReleaseImage( IplImage** image);
```

- image: image 는 IplImage 구조체의 이중 포인터이다. 따라서 포인터 변수 image 가 수록되어 있는 주소를 가리키므로 &image 로서 나타낸다. 다음은 Lena 영상을 메모리에 로드하고 로드된 영상을 메모리에서 해제하는 예이다.

```
IplImage* image;
```

```
image = cvLoadImage("../images/Lena.jpg", -1);
```

```
cvReleaseImage(&image);
```

- cvCreateImage 함수: 영상을 만들기 위해 사용하는 함수이다.

```
IplImage* cvCreateImage( cvSize size, int depth, int channels);
```

- size :영상의 크기를 지정한다.

영상의 크기는 cvSize() 함수와 cvGetSize() 함수를 사용하여 지정한다. 영상의 가로와 세로 길이를 직접 지정하고자 하는 경우는 cvSize() 함수를 사용하여 cvSize(width, height)처럼 지정하고 특정한 영상의 크기와 똑같은 크기의 영상을 만들고자 하는 경우는 cvGetSize() 함수를 사용하여 cvGetSize(src\_image)처럼 지정한다.

### 3) 영상 저장

- cvSaveImage 함수: 영상을 파일에 저장하기 위한 함수이다.

```
int cvSaveImage( char* filename, CvArr* image);
```

- filename: 저장할 파일이름을 나타낸다.

- image : 저장할 영상을 가리킨다.

다음은 cvSaveImage() 함수의 사용 예를 보여준다.

#### 4) 산술 연산

- cvAddS 함수: 영상에 상수값을 더하기 위해 사용하는 함수이다.

```
void cvAddS( CvArr* src, CvScalar value, CvArr* dst, CvArr* mask(NULL) );
```

- src : 원 영상을 나타낸다.

- value: 더하고자 하는 상수값을 지정한다.

상수값은 CvScalar 자료형을 갖는 CV\_RGB 와 cvScalarAll() 함수를 사용하여 지정한다. CV\_RGB 를 사용하여 지정하는 경우 예를 들어, 흑백영상에서 임의의 픽셀에 100 을 더하고자 하는 경우 상수값은 CV\_RGB(100, 100, 100)처럼 지정하고 컬러영상에서 RGB 각 채널에 상수값 100, 110, 120 을 더하고자 하는 경우 상수값은 CV\_RGB(100, 100, 120)처럼 지정하면 된다.

- dst: 목적 영상을 나타낸다.

- mask: 마스크를 지정한다. 마스크를 사용하지 않는 경우 NULL 로 지정한다.

5) 두 영상간의 산술 연산

6) 영상 반전

cvNot 함수: 반전 영상을 얻기 위해 사용하는 함수이다.

```
void cvNot( CvArr* src, CvArr* dst );
```

- src :원영상을 나타낸다.

- dst: 목적영상을 나타낸다.

cvNot() 함수는  $dst = \sim src$  를 수행한다.

7) 이진 영상

8) 영상변환

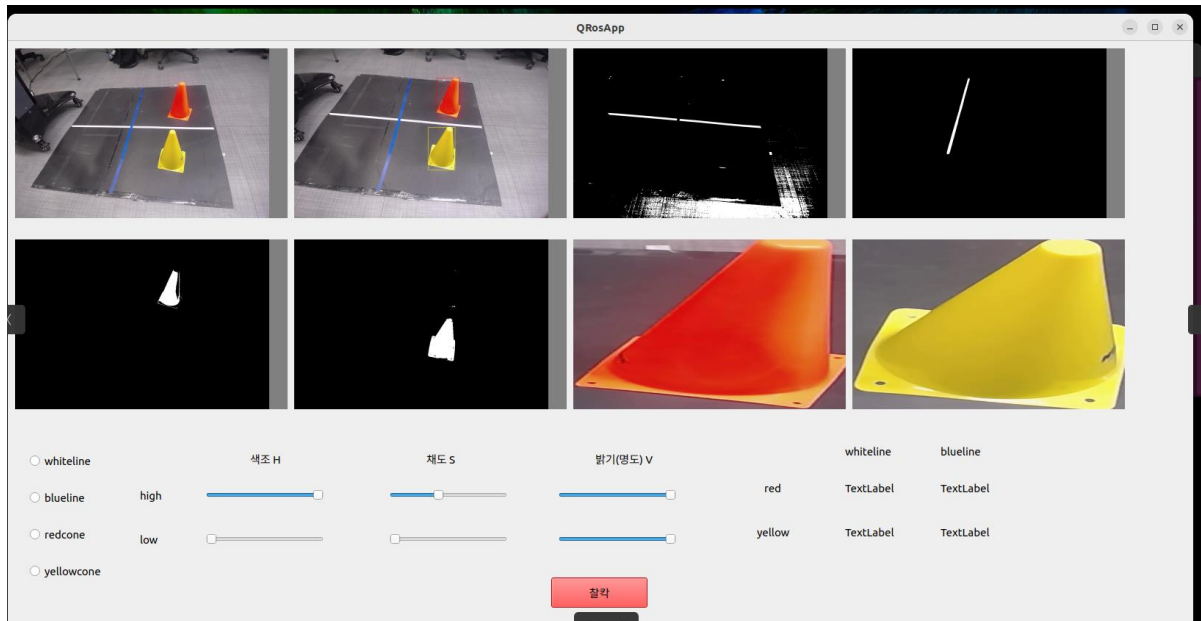
9) 영상 블러링

10) 영상 샤프닝

등에 쓰이는 다양한 함수들이 존재한다.

## 2. 코드 구성

## 1) UI



Ui 는 다음과 같이 제일 왼쪽에 `cam_raw` 그 다음 바운더리 박스 표시된 `cam_object` 와 흰색선과 푸른색 선을 각각 mask 이미지로 나타내는 `whiteline` 과 `blueline` , 빨간색 콘과 노란색 콘을 나타내는 `redcone` 과 `yellowcone` 과 각각 확대된 이미지를 표시하는 `redconezoom`, `yellowconezoom` 으로 구성하였다.

그리고 그 밑에는 radio 버튼으로 각각의 hsv 를 조정할 수 있게 모드 선택이 되도록 하였고 슬라이드 바를 각각 hsv high/low 에 할당하여 총 6 개의 슬라이더와 4 개의 라디오 버튼으로 모든 창을 제어할 수 있게 하였다. 그리고 오른쪽의 label 에는 위치를 인식하여 `right/left/on line/none` 을 표시하여 어느 사분면에 있는지 확인 할 수 있도록 하였다.

## 2) QNode

`Qobject` 와 `rclcpp::Node` 를 모두 상속 받아서 qt 와 ros 모두 사용할 수 있도록 하였다.

## 3) OpenCV

ROS2 의 기본 opencv 를 opencv\_bridge 를 사용하여 qt ros opencv 를 통합하였다.

### 3. 코드설명

#### 1) Mainwindow.cpp

### 4. 실행 및 결과

## 5. 참고문헌

- 1) <https://studium-anywhere.tistory.com/22>
- 2) <https://yonghello.tistory.com/entry/Open-CV-%ED%95%A8%EC%88%98%EC%A0%95%EB%A6%AC>
- 3) <https://diyver.tistory.com/98>

