1. 사용 기술

- OpenCV

OpenCV는 실시간 이미지 프로세싱에 중점을 둔 프로그래밍 Library로 윈도우,

리눅스 등에서 사용 가능한 플랫폼이며 오픈소스 BSD 허가 하에서 무료로

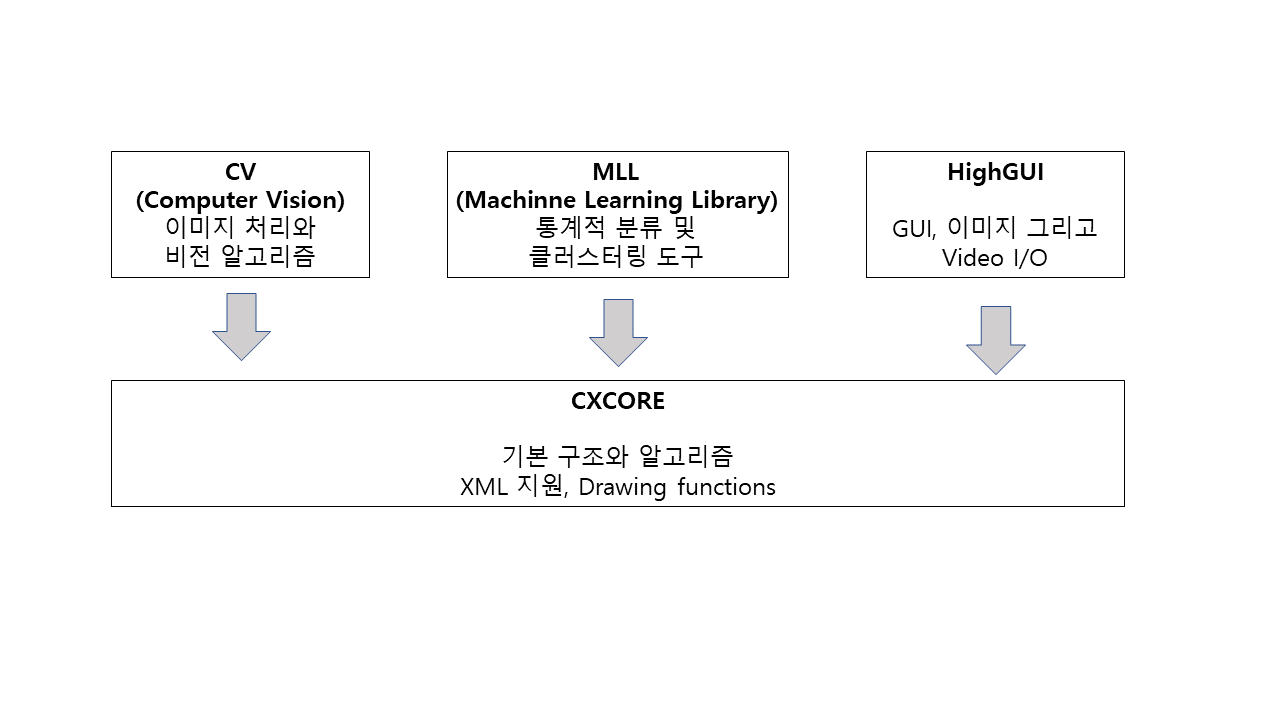
사용할 수 있다. OpenCV는 TensorFlow, Torch / PyTorch 및 Caffe의 딥러닝

프레임워크를 지원한다. 알고리즘 상으로 계산 효율성과 실시간 응용 프로그램에

중점을 두고 설계되었기 때문에 간단하게 OpenCV에서 제공되는 API를 사용하여

코딩하여도 실시간 프로세싱이 가능한 애플리케이션을 만들 수 있기 때문에

최적화나 알고리즘을 생각하지 않고도 품질 좋은 상용 프로그램을 만들 수 있다.



우리 조는 기존에 깔려있는 OpenCV를 삭제한 후 OpenCV 4.1.2를 빌드 하여

프로젝트를 진행하였다.

-TensorFlow

TensorFlow는 현재 가장 범용적으로 사용되는 딥러닝 프레임워크로 데이터

획득, 모델 학습, 예측, 미래 결과 정제와 같은 과정을 쉽게 해준다.

텐서플로우(TensorFlow)라는 이름처럼 텐서(Tensor) 다차원의 배열을 담고 있는

노드(Node)와 이를 다양한 연산으로 연결하고 있는 엣지(Edge)로 구성되어

있으며, 이러한 텐서들이 서로 연산을 통해 값을 주고받는 흐름(Flow)으로

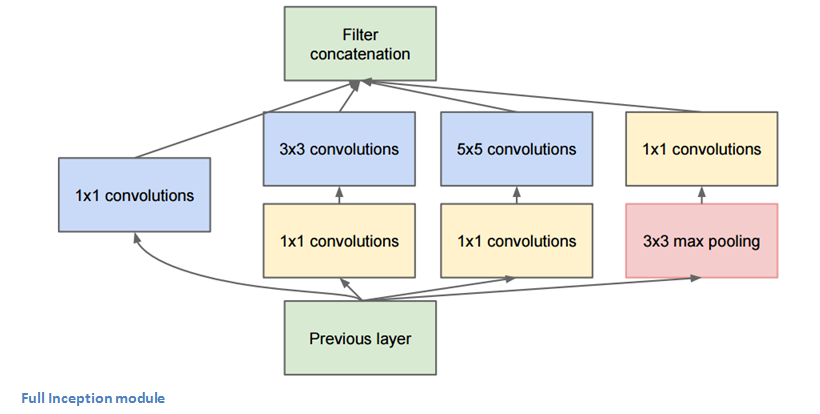
작동된다. 다수의 머신러닝과 딥 러닝 (신경망) 모델과 알고리즘을 결합해 공통

메타포를 통해 유용성을 높이기에 많이 사용된다. TensorFlow는 필기 숫자 판별,

이미지 인식, 단어 임베딩, 반복 신경망, 기계 번역을 위한 시퀀스 투 시퀀스

모델, 자연어 처리, PDE(편미분방정식) 기반 시뮬레이션 등을 위한 신경망을

학습, 실행할 수 있다.



-YOLO

YOLO는 대표적인 단일 단계 방식의 객체 탐지 알고리즘으로 이 알고리즘은

원본 이미지를 동일한 크기의 그리드로 나눈다. 각 그리드에 대해 그리드 중앙을

중심으로 미리 정의된 형태(predefined shape)으로 지정된 경계 박스의 개수를

예측하고 이를 기반으로 신뢰도를 계산한다. 이미지에 객체가 포함되어 있는지,

또는 배경만 단독으로 있는지에 대한 여부가 포함하여 높은 객체 신뢰도를 가진

위치를 선택해 객체 카테고리를 파악한다. 미리 정의-된 형태를 가진 경계 박스

수인 앵커 박스는 K-평균 알고리즘에 의한 데이터로부터 생성되며, 데이터

세트의 객체 크기와 형태에 대한 사전 정보를 확보한다. 각각의 앵커는 각기

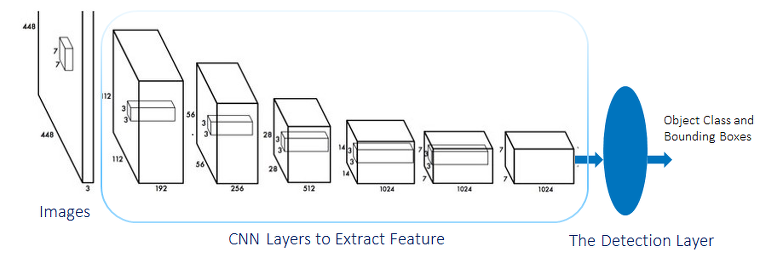
다른 크기와 형태의 객체를 탐지하도록 설계된다. 이 알고리즘은 앵커 박스와

유사한 크기의 개체를 탐지하여 최종 예측은 앵커의 위치나 크기와는 차이가

있다. 이미지의 Feature 맵에서 확보한 최적화된 오프셋이 앵커 위치나 크기에

추가된다. 탐지 레이어는 많은 회귀 및 분류 최적화 도구를 포함하고 있으며 레

이어의 개수는 앵커의 개수에 따라 결정된다.



네트워크의 최종 출력단에서 bounding box의 위치 찾기와 클래스 분류가

동시에 이뤄진다. 하나의 네트워크로 한 번에 특징 추출, 경계 박스 생성,

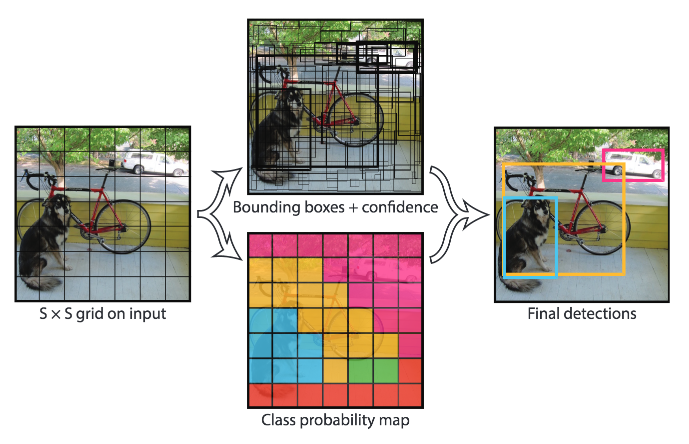
클래스를 분류하여 간단하고 빠르다. 이미지를 입력받아 S x S 크기의 그리드로

이미지를 나누어 각 그리드에서 예측을 한 후 이를 종합하여 bounding box를

구성한다. 이미지 내에서 알고리즘을 훈련시킬 때 사용된 클래스 라벨에 속하는

모든 물체를 검출하고, 그 위치들도 바운딩 박스로 알려준다. 만약 훈련된

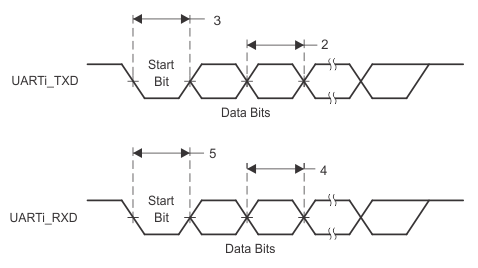
클래스 라벨에 속하는 물체가 이미지 내에 없으면 아무것도 검출해내지 않는다.



-UART

UART(범용 비동기화 송수신기: Universal asynchronous receiver/transmitter)는 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송하는 컴퓨터 하드웨어의 일종이다. UART는 일반적으로 EIA RS-232, RS-422, RS-485와 같은 통신 표준과 함께 사용한다. UART의 U는 범용을 가리키는데 이는 자료 형태나 전송 속도를 직접 구성할 수 있고 실제 전기 신호 수준과 방식(이를테면 차분 신호)이 일반적으로 UART 바깥의 특정한 드라이버 회로를 통해 관리를 받는다는 뜻이다. 통신 데이터는 메모리 또는 레지스터에 들어 있어 이것을 차례대로 읽어 직렬화 하여 통신한다. 최대 8비트가 기본 단위이다.

UART는 일반적으로 컴퓨터나 주변 기기의 일종으로 병렬 데이터를 직렬화 하여 통신하는 개별 집적 회로이다. 비동기 통신이므로 동기 신호가 전달되지 않는다. 따라서 수신 쪽에서 동기신호를 찾아내어 데이터의 시작과 끝을 시간적으로 알아 처리할 수 있도록 약속되어 있다. 디지털 회로는 자체의 클럭 신호를 사용하여 정해진 속도로 수신 데이터로 부터 비트 구간을 구분하고 그 비트의 논리 상태를 결정하여 데이터 통신을 한다.



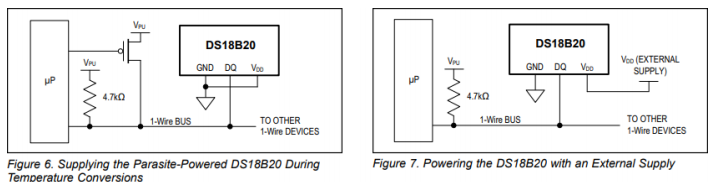
-FLASK

flask는 python의 마이크로 웹 프레임워크이다. 특별한 도구나 라이브러리가 필요 없기 때문에 마이크로 프레임워크라 부른다. 애플리케이션 기능을 추가할 수 있는 확장 기능을 지원하고 다양한 웹 엔진과 붙여서 가볍게 쓸 수 있어서 많이 사용된다. 확장 기능은 객체 관계매퍼, 양식 유효성 확인, 업로드 관리, 다양한 개방형 인증 기술, 여러 공통 프레임워크 관련 도구들을 위해 존재한다. 확장 기능들은 코어 플라스크 프로그램에 비해 훨씬 더 정기적으로 업데이트 된다. flask 설치는 pip install Flask를 통해 이뤄지고 Flask에서는 URL을 방문할 때 준비된 함수가 트리거 되도록 바인딩 하기 위해 route() 데코레이터를 사용한다. 이를 라우팅이라 한다.

2. 사용한 통신 방식

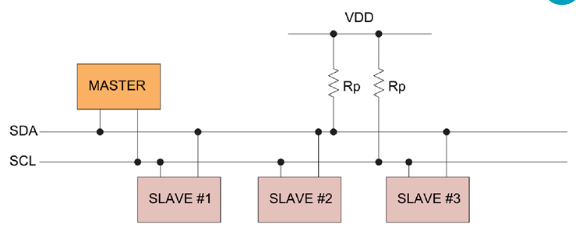
- one-wire(DS19B20)

Dallas Semiconductor 사에서 개발한 통신 방식으로 단지 하나의 선을 이용하는 직렬 통신이다. 하나의 통신선으로 전원 공급 및 다중 모듈 통신이 모두 가능하다. DB18B20 센서 역시 one-wire 통신 방식으로 데이터 전송속도는 느리지만 장거리 통신이 가능하다는 장점이 있다. DB18B20는 보통 온도계나 기상 관측장비 같은 저가의 장비들에 많이 이용된다.



-I2C(LCD)

I2C는 두 개의 신호선(SDA, SCL)으로 다수의 I2C 통신을 지원하는 디바이스와 데이터를 송/수신할 수 있는 통신방식이다. 하나의 마스터와 다수의 슬레이브로 연결이 구성되며, 마스터에서 기준클럭(SCL)을 생성하고, 이 클럭에 맞춰 데이터(SDA)를 전송 및 수신한다. 각 송신과 수신은 구분(송신과 수신이 동시에 이루어지지 않음)되어 있는 반이중(Half-Duplex) 방식이다. 각 슬레이브는 개별 주소(어드레스)를 가지고 있으며, 이 주소를 통해 식별이 가능하다. 즉, 기준클럭과 데이터는 I2C 네트워크의 모든 디바이스에게 전달되고, 해당 주소를 가진 디바이스만 응답하는 방식으로 서로 데이터를 주고 받는다.



3. 구현

-사람 객체 인식

|  |
| --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| - 1 라인  : OpenCV 패키지 import  - 2~3 라인  : cvlib 라이브러리에 있는 detect\_common\_objects 함수가 실질적으로 물체 검출 |

|  |
| --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| - UART 사용을 위해 serial 패키지 import 후 연결 |

|  |
| --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| * 7 라인   : 첫번째(0) 카메라를 VideoCapture 타입의 객체로 가져옴   * 9 라인   : 캠이 열려있는 경우 while문 반복   * 12 라인   : 물체 검출   * 15 라인   : 검출된 물체 가장자리에 바운딩 박스 그림   * 17 라인   : 카메라 display 출력   * 19 라인   : Q 눌렀을 경우 종료 |

|  |
| --- |
|  |
| - 22 라인  : labeling된 person 객체의 개수   * 24 라인   : UART serial 통신으로 person 수 송신 |

|  |
| --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| - 웹캠 resoure release |

|  |
| --- |
| camera\_person\_detect.py 결과 |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| bbox : detect한 부분, label : 물체를 detect한 라벨, conf : label로 분류된 확률 |

-온도 체크

|  |
| --- |
| readfla.py |
| 텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트이(가) 표시된 사진  자동 생성된 설명 |
| - 다양한 함수 호출을 통해 값들을 받아와 웹 화면에 온도와 인원수를 보여주는 부분이다.  - one-wire 통신을 통해 온도 센서 값을 받아오고 i2c 통신을 통해 LCD에 온도 값을 보여준다.  - uart 통신을 통해 다른 라즈베리에서 인원수 값을 받아와 웹에 보여준다.  - 온도가 30도 이상이거나 5인 이상인 경우에는 led 및 버저 음을 이용해 상황을 알리도록 했다. |

4. 구현 결과

|  |
| --- |
| **인원파악 라즈베리파이** |
|  |
| * 웹캠을 활용하여 현재 인원을 파악한다. * 파악한 인원 정보를 UART를 활용하여 서버 라즈베리파이에 송신한다. |
| **서버 및 센서 라즈베리파이** |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| * 웹캠의 인원파악 정보를 시리얼 통신으로 수신받고, 그 외 발열 체크 정보를 측정하여, 취합한 정보를 웹사이트를 통해 제공시켜준다. * 체온의 경우 서버 및 센서 라즈베리파이에서 측정을 한다. |

|  |
| --- |
| **서버 및 센서 라즈베리파이 기능 설명** |
| 텍스트, 테이블이(가) 표시된 사진  자동 생성된 설명 |
| * 체온이 일정 기준을 넘지 않거나, 인원파악 결과 5명을 넘지 않을 때   ▶ 녹색 LED 점등, 현재 인원 및 체온 정보 제공   * 체온이 기준을 넘거나. 인원파악 결과 5명이 넘었을 경우   ▶ 적색 LED 점등, 부저를 통한 알림, 현재 인원 및 체온 정보 제공 |

|  |  |
| --- | --- |
| **인원파악 라즈베리파이의 인원이 5인 이하일 경우** | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 실내, 모니터, 화면이(가) 표시된 사진  자동 생성된 설명 |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| * 5인 이하의 경우 인원을 준수한 상황이므로 녹색 LED 점등을 하여 양호한 상황임을 알려준다. * 취합된 온도 및 인원의 정보를 웹사이트를 통해 제공한다. | |
| **인원파악 라즈베리파이의 인원이 5인 이상일 경우** | |
| 텍스트, 실내, 디스플레이이(가) 표시된 사진  자동 생성된 설명 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 텍스트, 전자기기이(가) 표시된 사진  자동 생성된 설명 |
| * 5인 이상의 경우 적색 LED 점등과 동시에 부저에서 알림음이 울리게 된다. * 취합된 온도 및 인원의 정보를 웹사이트를 통해 제공한다. | |

|  |  |
| --- | --- |
| **체온측정 결과 정상 범위일 경우** | |
|  | 텍스트이(가) 표시된 사진  자동 생성된 설명 |
| 전자기기, 회로이(가) 표시된 사진  자동 생성된 설명 | |
| * 정상범위의 체온일 경우, 녹색 LED 점등을 하여 양호한 상황임을 알려준다. * 취합된 온도 및 인원의 정보를 웹사이트를 통해 제공한다. | |

|  |  |
| --- | --- |
| **체온측정 결과 정상 범위가 아닐 경우** | |
| 사람, 실내이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 디스플레이이(가) 표시된 사진  자동 생성된 설명 |
| 텍스트, 전자기기, 회로이(가) 표시된 사진  자동 생성된 설명 | |
| * 정상범위의 체온이 아닐 경우, 적색 LED 점등과 부저에서 알림음이 울리게 된다. * 취합된 온도 및 인원의 정보를 웹사이트를 통해 제공한다. | |