YOLO_MARKER 사용법

With OpenCV 3.2

In Ubunut 18.04.5

1. OPENCV 3.2 설치

- YOLO_MARK 공식 레퍼지토리에는 opencv 2 버전과 opencv 3 버전 둘 다 호환된다고 나와있다.
- 따라서 OPENCV 3.2 를 설치한다.

- 다른 환경과 겹치지 않게 가상환경을 생성한다.

1-1 기본 패키지 설치

- Sudo apt-get update
- Sudo apt-get upgrade
- Sudo apt-get install g++
- sudo apt-get install build-essential cmake
- Sudo apt-get install pkg-config
- Sudo apt-get install libjpeg-dev libpng-dev
- Sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libxvidcore-dev libx264-dev libxine2-dev

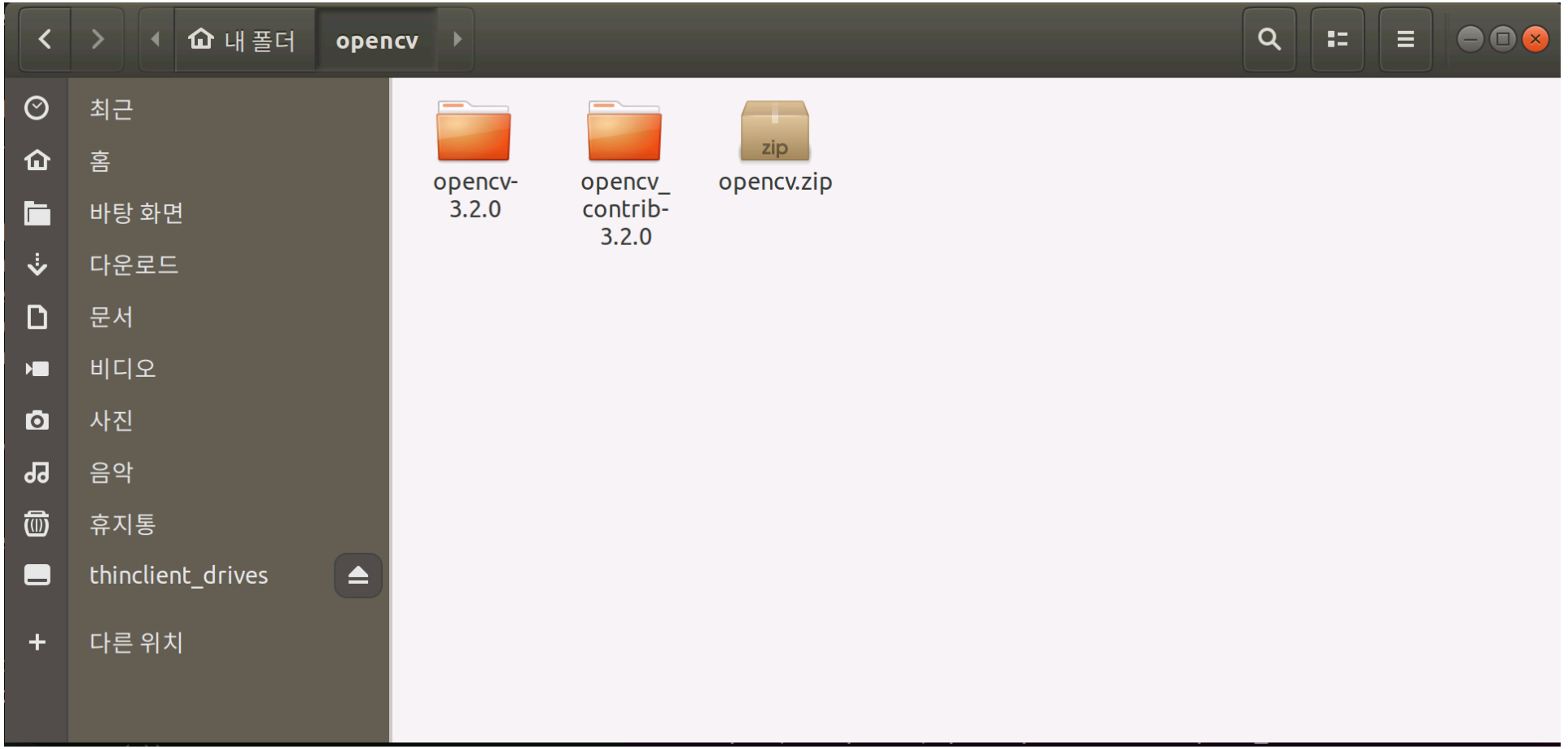
1-1 기본 패키지 설치

- sudo apt-get install lib41-dev v4l-utils
- sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
- sudo apt-get install libgtk2.0-dev
- sudo apt-get install mesa-utils libgl1-mesa-dri libgtkgl2.0-dev libgtkglext1-dev
- sudo apt-get install libatlas-base-dev gfortran libeigen3-dev
- sudo apt-get install python2.7-dev python3-dev python-numpy python3-numpy

1-2 경로 설정 및 설치

- Mkdir opencv
- Cd opencv
- Wget –O opencv.zip https://github.com/Itseez/opencv/archive/3.2.0.zip
- Unzip opencv.zip
- Wget –O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.2.0.zip
- Unzip opencv_contrib.zip

1-2 경로 설정 및 설치 결과 사진



1-2 경로 설정 및 설치

- Cd opencv-3.2.0
- Mkdir build
- Cd build

1-2 경로 설정 및 설치

- cmake ₩

-D CMAKE_BUILD_TYPE=RELEASE ₩

-D CMAKE_INSTALL_PREFIX=/usr/local ₩

-D WITH_TBB=OFF ₩

-D WITH_IPP=OFF ₩

-D WITH_1394=OFF₩

-D BUILD_WITH_DEBUG_INFO=OFF ₩

-D BUILD_DOCS=OFF ₩

-D INSTALL_C_EXAMPLES=ON ₩

-D INSTALL_PYTHON_EXAMPLES=ON ₩

-D BUILD_EXAMPLES=OFF ₩

-D BUILD_TESTS=OFF ₩

-D BUILD_PERF_TESTS=OFF ₩

-D ENABLE_NEON=ON ₩

-D WITH_QT=OFF ₩

-D WITH_OPENGL=ON ₩

-D WITH_CUDA=OFF ₩

-D WITH_CUDNN=ON ₩

-D OPENCV_DNN_CUDA=ON ₩

-D CUDA_ARCH_BIN=7.5 ₩

-D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-3.2.0/modules ₩

-D WITH_V4L=ON ₩

-D WITH_FFMPEG=ON ₩

-D WITH_XINE=ON ₩

-D BUILD_NEW_PYTHON_SUPPORT=ON ₩

-D PYTHON_INCLUDE_DIR=/usr/include/python2.7₩

-D BUILD_LIBPROTOBUF_FROM_SOURCES=ON -D PYTHON_INCLUDE_DIR2=/usr/include/x86_64-linux-gnu/python2.7 -D PYTHON_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython2.7.so ../
2>&1 ../../opencv-3.2.0/ | tee cmake_messages.txt

Make –j 할 때 에러시 OFF

Cmake error 시 OFF

1-2 경로 설정 및 설치

- cat /proc/cpuinfo | grep processor | wc –l
- Sudo make –j12
- Sudo make install

숫자 개수 조절

위 명령어의 출력값으로 사용 가능한 cpu 코어 개수 확인

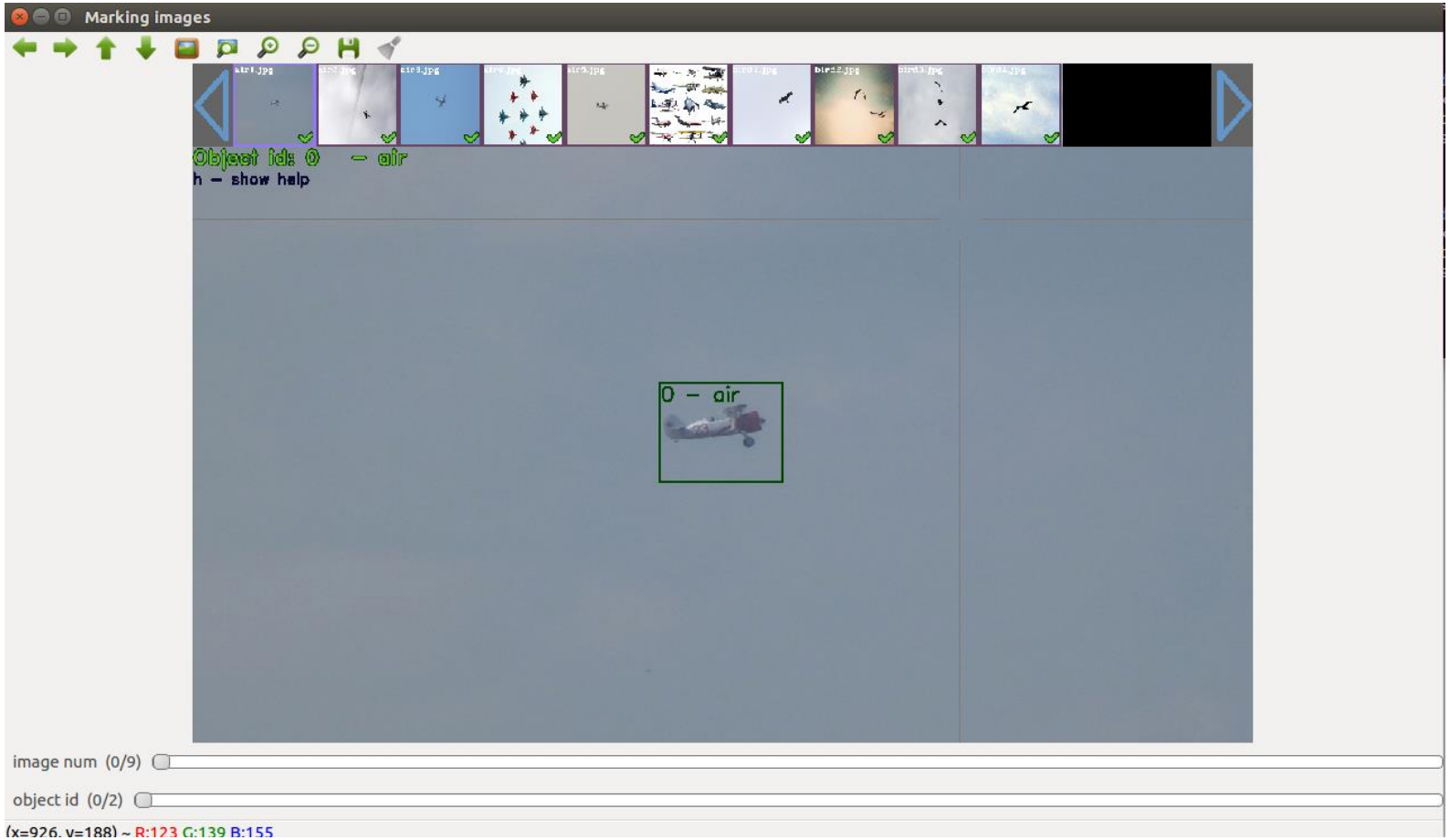
1-3 opencv 설치 확인

- Pkg-config --modversion opencv (버전 확인)
- Pkg-config –libs –cflags opencv

2. YOLO_MARK 사용

- git clone https://github.com/AlexeyAB/Yolo_mark
- Cd Yolo_mark
- Cmake .
- Chmod u+x ./linux_mark.sh
- ./linux_mark.sh

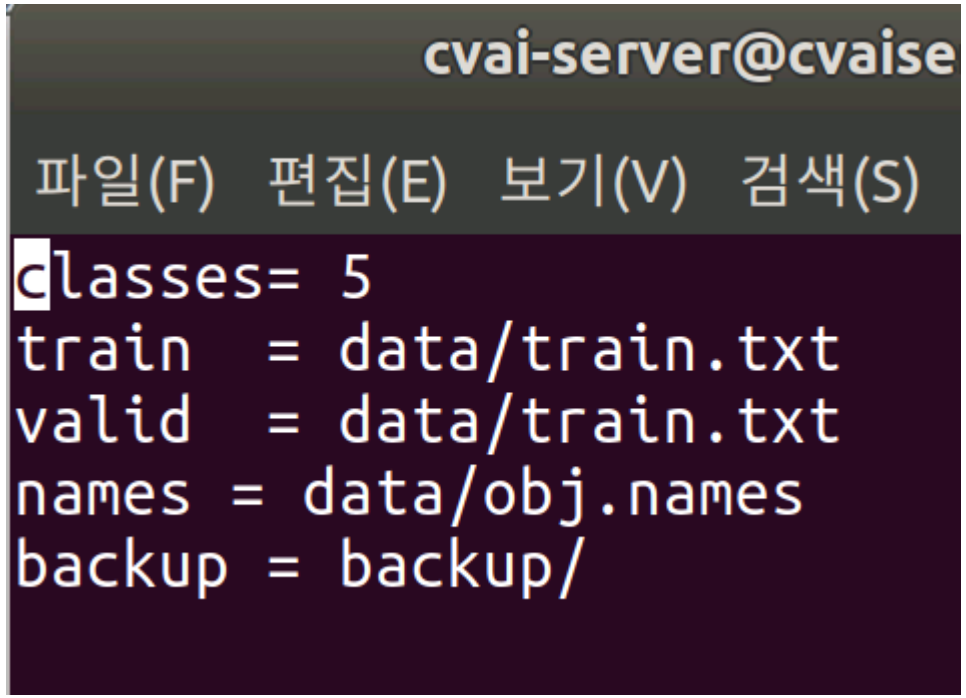
2-1 YOLO_MARK 샘플 사용 결과 화면



2-2 YOLO_MARK 설정

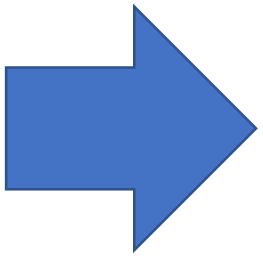
- 이미지가 뜨는 것을 확인 한 후 ESC 를 누르고 샘플 데이터를 종료한다.
- 그 이후 Yolo_mark/x64/Release/data/img 안의 샘플 폴더를 모두 지운다.
- 그리고 labeling할 이미지를 모두 그 폴더로 넣는다
- Yolo_mark/x64/Release/data 경로로 들어간 후 vi obj.data 입력

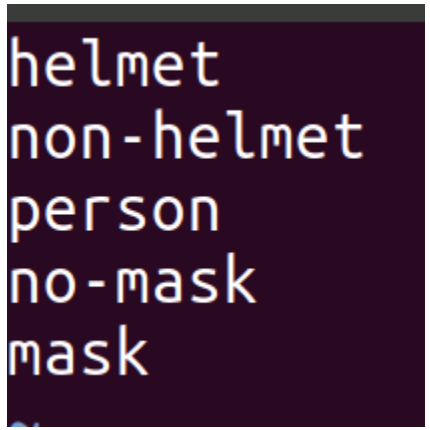
2-2 YOLO_MARK 설정



Obj.data

Classes 의 개수를 편집 후 :wq (저장하고 나가기) 입력





Obj.names

Obj.data 수정이 끝난 후 vi obj.names로 접속 한 다음 클래스의 이름을 설정하고 :wq

2-3 YOLO_MARK 실행

- Cd ~/Yolo_mark
- ./linux_mark.sh

실행 화면

2-3-1 YOLO_MARK 조작법

- 마우스 드래그로 bounding box 그림
- 숫자 패드로 클래스의 번호 생성
- 잘못 그렸을 경우 생성한 bounding box 를 클릭하고 R
- 자세한 도움말 H

2-4 YOLO_MARK 마무리

- 입력이 끝난 후 ctrl c
- 라벨링한 좌표들은 Yolo_mark/x64/Release/data/img 에 저장 및 텍스트 파일에 모두 저장되어있음
- 차후 yolo 모델 작업에 따라 변경 필요