

임베디드 뉴비를 위한 참고서

다음 과정은 모두 Jetson 5.1.3 환경에서 진행하였습니다.



설치환경

python = 3.8.10

opencv = 4.5.0 → 4.10.0

pytorch = 2.1.0a0+41361538.nv23.06

torch vision = 0.16.0

ultralytics = 8.2.69

1. 처음 시작

처음 시작 후 가장 먼저 할 것은 시간 설정입니다.

```
sudo timedatectl set-ntp off
```

`sudo timedatectl set-time "2024-07-29 HH:MM:SS"` : 년도 월 날짜 시간,분,초를 설정해주세요.

→ 예시 : `sudo timedatectl set-time "2024-07-29 10:30:52"`

`date` 로 시간을 확인합니다.

`sudo timedatectl set-timezone Asia/Seoul` 하면 한 방에 해결된다.

아니면 `set TZ=Asia/Seoul` 해도 된다.

2. apt 최신화 하기

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo reboot
```

를 통해서 apt들을 최신버전으로 바꿔줍니다.

이 과정은 인터넷과 다운 서버에 따라 시간이 오래 걸릴 수 있습니다.

만약 조금이라도 더 빨리 다운받고 싶다면 다운 서버를 바꾸면 됩니다.

▼ 우분투 apt-get update 속도 해결방법

- 스크립트를 다운로드합니다:

```
curl -Ss https://gist.githubusercontent.com/lesstif/8185f143ba7b8881e767900b1c8e98ad/raw/b749a8fc590ad2391fd0f8849417eeec998b33a7/change-ubuntu-mirror.sh -o change-ubuntu-mirror.sh
```

- 다운로드한 스크립트에 실행 권한을 부여합니다:

```
chmod +x change-ubuntu-mirror.sh
```

- 스크립트를 루트 권한으로 실행합니다:

```
sudo ./change-ubuntu-mirror.sh
```

카카오 사용

```
sudo bash ./change-ubuntu-mirror.sh -k
```

kaist 사용

```
sudo bash ./change-ubuntu-mirror.sh -n
```

부경대 사용

```
sudo bash ./change-ubuntu-mirror.sh -a
```

3. ros 설치하기(파이썬 3.8.1 설치 포함)

ros란?

ROS(로봇 운영 체제, Robot Operating System)는 로봇 소프트웨어 개발을 위한 오픈 소스 프레임워크입니다. ROS는 로봇 개발자에게 공통적인 기능을 제공하고, 로봇 애플리케이션의 개발, 테스트, 배포를 단순화하는 것을 목표로 합니다. ROS는 다양한 언어와 툴을 지원하며, 로봇 시스템의 구성 요소를 모듈화하여 재사용 가능하게 합니다. 주요 기능과 개념은 다음과 같습니다:

주요 기능

1. **하드웨어 추상화:** ROS는 하드웨어 추상화 레이어를 제공하여 다양한 로봇 하드웨어와의 상호작용을 단순화합니다.
2. **저수준 장치 제어:** 모터 제어, 센서 데이터 수집 등 저수준 장치 제어 기능을 지원합니다.
3. **메시지 전달:** 노드 간 메시지 전달을 통해 로봇의 다양한 구성 요소 간 통신을 가능하게 합니다.
4. **패키지 관리:** 로봇 소프트웨어를 패키지로 관리하고, 의존성 해결 및 배포를 지원합니다.
5. **시뮬레이션:** Gazebo와 같은 시뮬레이터와의 통합을 통해 로봇 동작을 시뮬레이션할 수 있습니다.

이번 프로젝트에서는 ros2에서 갈로틱 버전을 설치합니다.

▼ 설치 코드



```
sudo apt install python3-pip
sudo -H pip3 install -U jetson-stats

sudo apt update
sudo apt install curl gnupg2 lsb-release -y

sudo curl -sSL
https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc |
sudo apt-key add -

sudo sh -c 'echo "deb http://packages.ros.org/ros2/ubuntu focal
main" > /etc/apt/sources.list.d/ros2-latest.list'

mkdir -p ~/ros2_ws/src
cd ~/ros2_ws/src
git clone
https://github.com/zeta0707/monicar.git

git clone --recursive
https://github.com/zeta0707/darknet\_ros\_fp16.git

darknet_ros_fp16/darknet_ros/rm_darknet_CMakeLists.sh

cd ~/ros2_ws/src/monicar/script
./boardSelect.sh orin
./carSelect.sh pca9685Steer
./camSelect.sh usbcam

sudo apt-get update
sudo apt install ros-galactic-desktop
```

3 - 1 추가 작업

```
echo "source /opt/ros/galactic/setup.bash" >> ~/.bashrc
source ~/.bashrc
sudo apt install python3-colcon-common-extensions
```

```
cd ~/ros2_ws
colcon build
source install/setup.bash
```

위 작업을 추가해주세요.

4. OpenCV 설치하기

OpenCV(Open Source Computer Vision Library)는 실시간 컴퓨터 비전 애플리케이션을 개발하기 위한 오픈 소스 라이브러리입니다. OpenCV는 이미지 처리, 비디오 캡처 및 분석, 객체 검출 및 추적, 머신 러닝 등의 다양한 기능을 제공합니다. 주로 C++, Python, Java 등의 언어를 지원하며, 다양한 운영 체제에서 사용할 수 있습니다.

▼ 주요 기능

1. 이미지 처리

- 이미지 필터링(블러링, 샤프닝 등)
- 히스토그램 분석 및 정규화
- 이미지 변환(회전, 크기 조정, 크롭 등)
- 색상 공간 변환(RGB, HSV, Grayscale 등)

2. 비디오 처리

- 비디오 캡처 및 저장
- 프레임 간 차이 계산
- 배경 제거 및 이동 물체 검출

3. 특징 검출 및 매칭

- 에지 검출(Canny, Sobel 등)
- 코너 검출(Harris, Shi-Tomasi 등)
- 특징점 추출(SIFT, SURF, ORB 등)
- 특징점 매칭(FLANN, BFMatcher 등)

4. 객체 검출 및 추적

- 얼굴 검출 및 인식(Haar Cascades, DNN 등)
- 사람, 자동차 등 객체 검출(YOLO, SSD 등)
- 객체 추적(MeanShift, CamShift, KCF 등)

5. 기계 학습

- 분류 및 회귀(SVM, k-NN, Decision Trees 등)
- 군집화(k-Means, MeanShift 등)
- 신경망(DNN 모듈을 통한 딥러닝 지원)

6. 3D 시각화 및 인식

- 스테레오 비전
- 3D 재구성
- 포즈 추정 및 매칭

이번 프로젝트는 4.5.0 버전을 설치하겠습니다.

설치 코드(* 주의 : 설치가 까다롭습니다.)

기존opencv 제거

```
sudo apt-get purge *libopencv*
```

관련 라이브러리 설치

```
sudo apt-get update
sudo apt-get install -y build-essential cmake git unzip pkg-c
sudo apt-get install -y libjpeg-dev libpng-dev libtiff-dev
sudo apt-get install -y libavcodec-dev libavformat-dev libsws
sudo apt-get install -y libgtk2.0-dev libcanberra-gtk*
sudo apt-get install -y python3-dev python3-numpy python3-pip
sudo apt-get install -y python3-dev python3-numpy libtbb2 lib
sudo apt-get install -y libxvidcore-dev libx264-dev libgtk-3-
sudo apt-get install -y libtbb2 libtbb-dev libdc1394-22-dev
sudo apt-get install -y libv4l-dev v4l-utils
sudo apt-get install -y libgstreamer1.0-dev libgstreamer-plug
sudo apt-get install -y libavresample-dev libvorbis-dev libxi
sudo apt-get install -y libfaac-dev libmp3lame-dev libtheora-
sudo apt-get install -y libopencore-amrnb-dev libopencore-amr
sudo apt-get install -y libopenblas-dev libatlas-base-dev lib
```

```
sudo apt-get install -y liblapack-dev libeigen3-dev gfortran
sudo apt-get install -y libhdf5-dev protobuf-compiler
sudo apt-get install -y libprotobuf-dev libgoogle-glog-dev li
```

opencv 파일 다운

```
cd ~
wget -O opencv.zip https://github.com/opencv/opencv/archive/4
wget -O opencv_contrib.zip https://github.com/opencv/opencv_c

unzip opencv.zip
unzip opencv_contrib.zip
```

빌드하기(주의!!)

```
cd opencv-4.5.0

mkdir build
cd build

cmake \
-D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib-4.5.0/modules \
-D EIGEN_INCLUDE_PATH=/usr/include/eigen3 \
-D WITH_OPENCLAMDBLAS=OFF \
-D WITH_GTK=OFF \
-D WITH_OPENCL=OFF \
-D WITH_CUDA=ON \
-D CUDA_FAST_MATH=ON \
-D OPENCV_DNN_CUDA=ON \
-D CUDA_ARCH_PTX="" \
-D WITH_CUDNN=ON \
-D WITH_CUBLAS=ON \
-D ENABLE_FAST_MATH=ON \
-D ENABLE_NEON=ON \
```

```
-D WITH_QT=OFF \  
-D WITH_OPENMP=ON \  
-D WITH_OPENGL=ON \  
-D BUILD_JASPER=ON \  
-D BUILD_TIFF=ON \  
-D WITH_FFMPEG=ON \  
-D WITH_GSTREAMER=ON \  
-D WITH_TBB=ON \  
-D BUILD_TBB=ON \  
-D WITH_EIGEN=ON \  
-D WITH_V4L=ON \  
-D WITH_LIBV4L=ON \  
-D WITH_VTK=ON \  
-D OPENCV_ENABLE_NONFREE=ON \  
-D INSTALL_PYTHON_EXAMPLES=OFF \  
-D BUILD_NEW_PYTHON_SUPPORT=ON \  
-D OPENCV_GENERATE_PKGCONFIG=ON \  
-D BUILD_OPENCV_WORLD=ON \  
-D BUILD_PERF_TESTS=OFF \  
-D BUILD_TESTS=OFF \  
-D BUILD_OPENCV_PYTHON_TESTS=OFF \  
-D INSTALL_TESTS=OFF \  
-D INSTALL_C_EXAMPLES=OFF \  
-D BUILD_EXAMPLES=OFF ..
```

make j(오래 걸립니다)

```
nproc  
make -j4  
sudo rm -rf /usr/include/opencv4/opencv2  
sudo make install
```

설치 후 확인


```
import cv2
print("OpenCV version:", cv2.__version__)

if cv2.cuda.getCudaEnabledDeviceCount() > 0:
    print("CUDA is supported!")
    print("Number of CUDA devices:", cv2.cuda.getCudaEnabledDeviceCount())
else:
    print("CUDA is not supported.")
```

```
import cv2
cv2.__version__
cv2.cuda.getCudaEnabledDeviceCount()
cv2.cuda.getCudaEnabledDeviceCount() # → 1이 나와야 정상입니다.
```

트러블슈팅

그런데 만약에 2가지 문제가 발생한다면?

1. import cv2 가 안될시
2. cv2.cuda.getCudaEnabledDeviceCount() 가 0이 나올시

해결방법은 다음과 같습니다.



만약 import cv2 할 때, 아래와 오류가 난다면?

```
export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1
```

1. import cv2 문제의 경우

<https://gainful-fisherman-502.notion.site/GPU-jetpack-5-1-3-dc4ba954fa2d4cb590f3fb4b7aa4087f>

위 링크를 참고하는 방법도 있습니다.

```
import cv2
cv2.__version__
cv2.cuda.getCudaEnabledDeviceCount()
```

만약 `cv2.__version__` 를 통해서 버전이 달라졌다면 될 수 있습니다.

아니라면 쿠다를 지원하는 `opencv`를 설치하거나 재설치가 답입니다..

5. 파이토치 & 토치 비전 설치하기

PyTorch & torchvision : 딥러닝과 컴퓨터 비전을 위한 두 가지 중요한 라이브러리입니다.

PyTorch는 Facebook AI Research(FAIR)에서 개발한 오픈 소스 딥러닝 프레임워크입니다. 다음과 같은 주요 특징을 갖추고 있습니다:

▼ 특징

1. 동적 계산 그래프(Dynamic Computation Graphs):

- PyTorch는 동적 계산 그래프를 사용합니다. 이는 계산 그래프가 런타임에 정의되므로, 유연하고 직관적인 모델 정의 및 디버깅이 가능합니다.

2. 직관적인 인터페이스:

- Pythonic한 인터페이스를 제공하여, Python 사용자에게 익숙한 방식으로 코딩할 수 있습니다. 이는 연구 및 프로토타이핑에 매우 유리합니다.

3. 자동 미분(Automatic Differentiation):

- PyTorch의 `autograd` 패키지는 자동 미분을 지원하여, 기울기 계산을 자동화합니다. 이는 딥러닝 모델의 학습 과정에서 매우 유용합니다.

4. GPU 지원:

- PyTorch는 CUDA를 통해 GPU 가속을 지원하여, 대규모 연산을 빠르게 처리할 수 있습니다.

5. 커뮤니티와 생태계:

- PyTorch는 활발한 커뮤니티와 다양한 관련 라이브러리(예: torchvision, PyTorch Geometric) 덕분에, 많은 연구자와 개발자들이 사용하고 있습니다.

•

torchvision은 PyTorch를 위한 컴퓨터 비전 라이브러리로, 주요 기능은 다음과 같습니다:

1. 데이터셋:

- torchvision은 CIFAR-10, ImageNet, COCO 등 여러 표준 컴퓨터 비전 데이터셋을 손쉽게 로드할 수 있는 기능을 제공합니다.

2. 모델:

- torchvision에는 사전 학습된(pre-trained) 모델들이 포함되어 있습니다. 이를 통해 ResNet, VGG, AlexNet, MobileNet 등 다양한 아키텍처를 손쉽게 사용할 수 있습니다.

3. 변환(Transformations):

- 이미지 데이터를 전처리하거나 변환하는 다양한 기능(예: 자르기, 회전, 정규화 등)을 제공합니다.

4. 유틸리티 함수:

- 이미지 시각화, 데이터를 병렬로 로드하기 위한 DataLoader 등의 유틸리티 함수가 포함되어 있습니다.

▼ 설치 코드 1번

```
sudo apt-get -y install python3-pip libopenblas-dev;
export TORCH_INSTALL=https://developer.download.nvidia.cn/c
python3 -m pip install --upgrade pip; python3 -m pip instal

sudo apt-get install libjpeg-dev zlib1g-dev libpython3-dev
git clone --branch v0.16.0 https://github.com/pytorch/vision
cd torchvision
export BUILD_VERSION=0.16.0 # where 0.x.0 is the torchvisi

python3 setup.py install --user
```

▼ 설치 코드 2번 → 이걸로 해결함

torch, torchvision 설치하기

```
# Python 패키지 관리자(Pip)를 설치
sudo apt install python3-pip

# Python에서 날짜와 시간의 파싱, 조작, 연산 등을 쉽게 할 수 있도록 도
```

```

pip3 install -U python-dateutil

# 현재 설치된 버전 삭제
pip uninstall torch torchvision

# pi 토치에 필요한 모든 ubuntu 패키지를 설치한다.
sudo apt-get -y install python3-pip libopenblas-dev;

# 다음의 단계에 따라 pytorch를 설치한다.
export TORCH_INSTALL=https://developer.download.nvidia.cn/c

# Install PyTorch.
python3 -m pip install --upgrade pip; python3 -m pip instal

# torchvision 설치
sudo apt-get install libjpeg-dev zlib1g-dev libpython3-dev
git clone --branch v0.16.0 https://github.com/pytorch/vision
cd torchvision
export BUILD_VERSION=0.16.0 # where 0.x.0 is the torchvisi
python3 setup.py install --user
# cd ../ # attempting to load torchvision from build dir w
# pip install 'pillow<7' # always needed for Python 2.7, nc

#setup.py install --user 명령어는 매우 오래 걸린다..!
#(한 30분 정도 걸리니 기다리자!)

```

설치 확인

```

import torch
import torchvision

print("PyTorch version:", torch.__version__)
print("TorchVision version:", torchvision.__version__)

```

```
cuda_available = torch.cuda.is_available()
print("CUDA availbale: ", cuda_available)
```

6. yolov8 설치하기

공식 사이트

1. 패키지 목록 업데이트, pip 설치 및 최신 버전으로 업그레이드

```
sudo apt update
sudo apt install python3-pip -y
pip install -U pip
```

2. 설치 `ultralytics` 선택적 종속성이 있는 pip 패키지

```
pip install ultralytics[export]
```

3. 디바이스 재부팅

```
sudo reboot
```

6-1. import cv2 에러가 난다면?

```
pip uninstall opencv-python
pip uninstall opencv-python-headless
```

4.10.0 버전 설치로 오류가 날 수 있으니 삭제해 보세요.

- 전 현재 `cv2.cuda.getCudaEnabledDeviceCount()` 하면 '0' 이 나오지만 CUDA 잘 됩니다.
- `torch.cuda.is_available()` 이 True라면 잘 될 수 있으니 확인해보세요.

7. yolo pt 파일 실행하기

```

import cv2
import torch

from ultralytics import YOLO

# yolo 모델 로드
model = YOLO('best.pt') # 파일 이름 넣으시면 됩니다.

# CUDA 사용 가능한지 확인
device = 'cuda' if torch.cuda.is_available() else 'cpu'
model.to(device)

# 카메라 설정(예: /dev/video0)
cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Error: Could not open camera.")
    exit()

# 2프레임마다 한 번씩 처리
frame_skip = 2
frame_count = 0

while True:
    # 프레임 읽기
    ret, frame = cap.read()
    if not ret:
        print("Error: Could not read frame.")
        break

    frame_count += 1
    if frame_count % frame_skip != 0:
        continue

    # 프레임 크기 조정
    resized_frame = cv2.resize(frame, (640, 640 ))

```

```

# 프레임을 BCHW 형식으로 변환
frame_tensor = torch.from_numpy(resized_frame).permute(2,

# YOLOv8 객체 감시 수행
results = model(frame_tensor)

# 감지된 객체를 프레임에 표시
for result in results:
    for box in result.bboxes:
        x1, y1, x2, y2 = box.xyxy[0].tolist()
        confidence = box.conf[0].item()
        class_id = box.cls[0].item()
        label = f'{model.names[int(class_id)]}: {confidence}'
        cv2.rectangle(resized_frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)
        cv2.putText(resized_frame, label, (int(x1), int(y1) + 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0))

# 결과 프레임 표시
cv2.imshow('YOLOv8 Detection', resized_frame)

# 'q' 키를 눌러 종료
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# 리소스 해제
cap.release()
cv2.destroyAllWindows()

```