

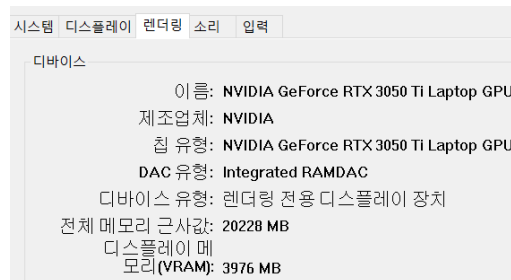
ROS

▼ YOLO

1. 윈도우 그래픽카드 관련 세팅

tensorflow에 그래픽카드 연산을 사용할 수 있게 하기 위해 세팅을 해준다.

- Nvidia 그래픽 드라이버 설치
 - - GPU를 지원하는 Tensorflow를 설치하기 위해 가장 먼저 NVidia 그래픽 드라이버를 설치
 - 그래픽 드라이버는 윈도우 dxdiag를 통해 확인할 수 있음
 - 확인한 GPU 유형에 맞는 그래픽 드라이버 다운로드(<https://www.nvidia.co.kr/Download/index.aspx?lang=kr>)



NVIDIA 드라이버 다운로드

아래의 드롭다운 목록에서 자신의 NVIDIA 제품에 알맞은 드라이버를 선택하세요. 1 [상세 가이드 보기](#)

제품 유형:	GeForce
제품 시리즈:	GeForce RTX 30 Series (Notebooks)
제품 계열:	GeForce RTX 3050 Ti Laptop GPU
운영 체제:	Windows 10 64-bit
다운로드 타입:	Game Ready 드라이버(GRD) ?
언어:	Korean

검색

- CUDA toolkit 설치
 - <https://developer.nvidia.com/cuda-toolkit-archive>
 - CUDA 11.2 버전 설치 : exe (local)로 설치
CUDA Toolkit 11.2.2(March 2021), Versioned Online Documentation
- cuDNN 설치(nvidia 로그인을 해야 받을 수 있음)
 - **Download cuDNN v8.1.0 (January 26th, 2021), for CUDA 11.0,11.1 and 11.2**
 - 압축해제 후 CUDA 폴더에 덮어쓰기를 해야함.

cuda nn 압축해제한 폴더이다.

bin	2023-03-13 오후 4:40	파일 폴더	
include	2023-03-13 오후 4:40	파일 폴더	
lib	2023-03-13 오후 4:40	파일 폴더	
NVIDIA_SLA_cuDNN_Support	2021-01-22 오전 3:47	텍스트 문서	21KB

CUDA 경로에 위의 파일들을 덮어쓴다.

내 PC > Windows10 (C:) > Program Files > NVIDIA GPU Computing Toolkit > CUDA > v11.2				
	이름	수정된 날짜	유형	크기
un	bin	2023-03-13 오후 4:40	파일 폴더	
	compute-sanitizer	2023-03-13 오후 4:32	파일 폴더	
	extras	2023-03-13 오후 4:32	파일 폴더	
	include	2023-03-13 오후 4:40	파일 폴더	
	lib	2023-03-13 오후 4:32	파일 폴더	
	libnvvp	2023-03-13 오후 4:32	파일 폴더	
	nvml	2023-03-13 오후 4:32	파일 폴더	
	nvvm	2023-03-13 오후 4:32	파일 폴더	
	nvvm-prev	2023-03-13 오후 4:32	파일 폴더	
	src	2023-03-13 오후 4:32	파일 폴더	
val	tools	2023-03-13 오후 4:32	파일 폴더	
	CUDA_Toolkit_Release_Notes	2021-02-24 오후 10:40	텍스트 문서	26KB
	DOCS	2021-02-24 오후 10:40	파일	1KB
	EULA	2021-02-24 오후 10:40	텍스트 문서	62KB
	NVIDIA_SLATCuDNN_Support	2021-01-22 오전 3:47	텍스트 문서	21KB
	README	2021-02-24 오후 10:40	파일	1KB
	version	2021-02-27 오전 5:38	JSON 원본 파일	3KB

2. Pytorch, Yolo 설치 과정

Versions

- python - 3.7.5
- cuda - 11.2
- cudnn - 8.1.0.77
- pytorch - torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio==0.8.1

Pytorch 설치하기

- 우리는 cuda 11.2 버전을 사용하고 있지만, pytorch에서 해당 버전이 존재하지 않는다. 해결 방법으로는 pytorch cuda 11.1 버전이 cuda 11.2 버전에서도 정상동작하는 것을 확인해서 해당 pytorch 버전을 설치해주면된다....

```
# CUDA 11.1
pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio==0.8.1 -f https://download.pytorch.org/whl/torch_stable.htm
```

- 설치 확인(해당 파이썬 파일을 실행시켜줬을때 에러가 나지 않고 결과값이 잘 출력되어야한다.)

```
import torch

dtype = torch.float
#device = torch.device("cpu") # cpu
device = torch.device("cuda:0") # gpu

# N은 배치 크기이며, D_in은 입력의 차원입니다;
# H는 은닉층의 차원이며, D_out은 출력 차원입니다.
N, D_in, H, D_out = 64, 1000, 100, 10

# 무작위의 입력과 출력 데이터를 생성합니다.
x = torch.randn(N, D_in, device=device, dtype=dtype)
y = torch.randn(N, D_out, device=device, dtype=dtype)

# 무작위로 가중치를 초기화합니다.
w1 = torch.randn(D_in, H, device=device, dtype=dtype)
w2 = torch.randn(H, D_out, device=device, dtype=dtype)

learning_rate = 1e-6
for t in range(500):
    # 순전파 단계: 예측값 y를 계산합니다.
    h = x.mm(w1)
```

```

h_relu = h.clamp(min=0)
y_pred = h_relu.mm(w2)

# 손실(loss)을 계산하고 출력합니다.
loss = (y_pred - y).pow(2).sum().item()
if t % 100 == 99:
    print(t, loss)

# 손실에 따른 w1, w2의 변화도를 계산하고 역전파합니다.
grad_y_pred = 2.0 * (y_pred - y)
grad_w2 = h_relu.t().mm(grad_y_pred)
grad_h_relu = grad_y_pred.mm(w2.t())
grad_h = grad_h_relu.clone()
grad_h[h < 0] = 0
grad_w1 = x.t().mm(grad_h)

# 경사하강법(gradient descent)를 사용하여 가중치를 갱신합니다.
w1 -= learning_rate * grad_w1
w2 -= learning_rate * grad_w2

```

Yolo v5 설치

- pytorch hub를 이용하여 yolo_v5를 실행 시킬 것이다.
- 바탕화면에서 git bash로 yolo v5 깃을 클론해온다.

```
git clone https://github.com/ultralytics/yolov5 # clone
```

- yolov5 폴더로 접근한다.

```
cd yolov5
```

- yolo v5에서 요구하는 패키지들을 설치해준다.

```
pip install -r requirements.txt
```

- 설치 확인(에러가 발생하지 않고 해당 이미지에서 객체 탐지가 잘 되어야 한다.)

```

import torch

# Model
model = torch.hub.load("ultralytics/yolov5", "yolov5s") # or yolov5n - yolov5x6, custom

# Images
img = "https://ultralytics.com/images/zidane.jpg" # or file, Path, PIL, OpenCV, numpy, list

# Inference
results = model(img)

# Results
results.print() # or .show(), .save(), .crop(), .pandas(), etc.

```

▼ ROS

첨부한 pdf 파일 참고

▼ Socket, TTS

Python 내부 패키지 클릭

- gtts

```
pip install gtts
```

- tts 사운드 파일 실행

```
pip install playsound
```

- socket

```
pip uninstall python-socketio  
pip install python-socketio
```

→ 이전에 requirements.txt 파일로 다운로드 된 파일이 버전이 낮아 소켓통신에서 400에러가 뜨기때문에 꼭 설치 해제 후 다시 설치
해야함