

주간 보고서-5주차 (2024.04.01 ~ 2024.04.07)

팀명 : 무^{YOLOV9} 계획

회의 참석자 : 서태원, 신재환, 최승렬, 류은환

회의 일자 및 회의 장소 :

2024.04.01(월)

- 18:30 ~ 23:00, 회의 장소 : 7호관 302호

2024.04.02(화)

- 19:00 ~ 23:00, 회의 장소 : 7호관 302호

2024.04.03(수)

- 13:00 ~ 16:30, 회의 장소 : 7호관 302호

2024.04.04(목)

- 16:30 ~ 18:00, 회의 장소 : 7호관 302호
- 19:00 ~ 22:00, 회의 장소 : 7호관 302호

2024.04.05(금)

- 16:00 ~ 18:00, 회의 장소 : 7호관 302호
- 19:00 ~ 23:00, 회의 장소 : 7호관 302호

총 활동 시간 : 22시간 30분

진행 사항 :

YOLOV9 Train Flow를 정리

- /home/pi(username)/yolov9 경로

```

$ git clone https://github.com/SkalskiP/yolov9.git
$ cd yolov9
$ pip install -r requirements.txt
$ pip install roboflow

$ wget -P /home/pi/weights -q https://github.com/WongKinYiu/y
$ wget -P /home/pi/weights -q https://github.com/WongKinYiu/y

$ ls -la /home/pi/weights

$ wget -P /home/pi/data https://media.roboflow.com/notebooks/

$ python detect.py --weights /home/pi/weights/yolov9-c.pt --c

```

- transfer learning을 하기위해 pretrained yolov9 weight를 받음.
- yolov9을 이용해 dog.jpeg에서 object detection을 이용해 바운딩 박스를 침.
- 라즈베리파이에서는 CUDA를 사용하지 못하기때문에 device 설정을 할때 CPU로 설정함
(--device cpu)

- practice1.py에서 돌린 code

```

from IPython.display import Image

Image(filename="/home/pi/yolov9/runs/detect/exp/dog.jpeg", wi

```

/home/pi/yolov9/runs/detect/exp/dog.jpeg에 detect된 사진이 저장됨.

```

$ cd /home/pi/yolov9

```

```

from roboflow import Roboflow

rf = Roboflow(api_key="____")

```

```
project = rf.workspace("mgh-jfd5j").project("car-6fyjc")
version = project.version(1)
dataset = version.download("yolov9")
```

- Robotflow에서 데이터셋에 바운딩박스과 class를 설정하는 annotation 작업을 .
- Robotflow에서 annotation된 dataset을 yolov9 model에 맞게 변형시키고 train, test, val dataset 비중을 설정한 후 API 추출.

```
$ cd /home/pi/yolov9
```

```
$ python train.py \
--batch 16 --epochs 25 --img 640 --device cpu --min-items 0 -
--data /home/pi/yolov9/car-1/data.yaml \
--weights /home/pi/weights/yolov9-c.pt \
--cfg models/detect/yolov9-c.yaml \
--hyp hyp.scratch-high.yaml
```

- batch size를 16, epoch를 25, img size를 640로 설정 후, Robotflow에서 만든 data를 이용해서 transfer learning을 시킴. 이때 pretrained 가중치는 /home/pi/weights/yolov9-c.pt에 있음.

```
from google.colab import drive
drive.mount('/content/drive')
```

- 이 부분은 google colab에서 mount 시키는 코드.
- Raspberry에서 Swap을 늘려줬지만 cpu로 train이 되지 않았기에 colab에서 train하여 model을 추출함.

```
$ ls /home/pi/yolov9/runs/train/exp/
```

- trained된 model이 있는 위치 경로.
- 이때 가장 잘 학습된 model이 best.pt로 저장됨.

```
from IPython.display import Image
```

```
Image(filename="/home/pi/yolov9/runs/train/exp/results.png",
```

```
$ cd /home/pi/yolov9
```

```
$ python val.py \
```

```
--img 640 --batch 32 --conf 0.001 --iou 0.7 --device cpu \
```

```
--data /home/pi/yolov9/car-1/data.yaml \
```

```
--weights /home/pi/yolov9/runs/train/exp2/weights/best.pt
```

- mAP를 이용해 가장 잘 학습된 model의 성능을 평가함.

```
$ python detect.py \
```

```
--img 1280 --conf 0.1 --device cpu \
```

```
--weights /home/pi/yolov9/runs/train/exp2/weights/best.pt \
```

```
--source {dataset.location}/test/images
```

- transfer learning을 이용해서 만든 best.pt model를 이용해서 object detection을 함.
- 이때 test dataset을 사용함.

- final.py에서 실행시킨 code

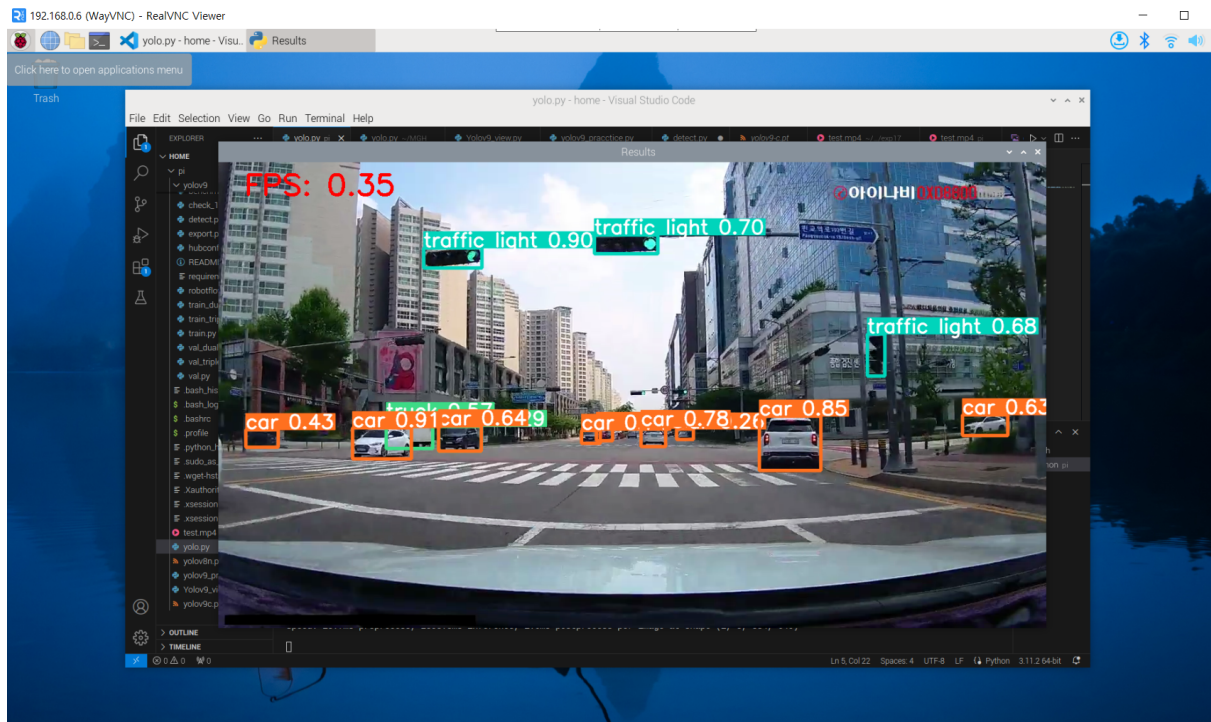
```
import glob
```

```
from IPython.display import Image, display
```

```
for image_path in glob.glob('/home/pi/yolov9/runs/detect/exp3  
    display(Image(filename=image_path, width=600))
```

- test dataset에 바운딩 박스와 레이블이 표시된 사진이
/home/pi/yolov9/runs/detect/exp3에 저장됨.

라즈베리파이5에서 yolov9-c를 구동시켜보고, fps가 0.3~0.5정도로 나온 것을 확인함.



coco dataset을 yolo형식에 맞게 txt파일로 변환해 주어야 하는 사실을 확인함. 그래서 이러한 dataset들을 YOLO저자가 만든 darknet 프레임워크가 사용하는 label format으로 변경해주는 프로젝트인 **convert2Yolo**를 이용하기로 결정한 후 해당 프로젝트가 어떤 형식으로 구성되어있고 어떻게 사용하는지 숙지하고 이를 적용함

개발환경 세팅 → Ubuntu 22.04 LTS에 CUDA를 설치

- 설치가능한 NVIDIA 그래픽 카드 드라이버 목록 중 추천 드라이버를 다운함(nvidia-driver-535)

```
(test2) mgh@ese-System-Product-Name:~$ sudo ubuntu-drivers devices
== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==
modalias : pci:v000010DEd00002684sv000010DEsd0000167Cbc03sc00i00
vendor    : NVIDIA Corporation
driver    : nvidia-driver-535 - distro non-free recommended
driver    : nvidia-driver-550-server - distro non-free
driver    : nvidia-driver-545-open - distro non-free
driver    : nvidia-driver-550-server-open - distro non-free
driver    : nvidia-driver-535-server - distro non-free
driver    : nvidia-driver-530 - third-party non-free
driver    : nvidia-driver-545 - distro non-free
driver    : nvidia-driver-535-open - distro non-free
driver    : nvidia-driver-535-server-open - distro non-free
driver    : xserver-xorg-video-nouveau - distro free builtin
```

```
$ sudo ubuntu-drivers devices
$ sudo apt install nvidia-driver-535
$ sudo reboot
$ sudo lshw -c display
```

- configure 항목에서 NVIDIA 그래픽 카드 드라이버가 설치되어 nvidia로 바뀐 것을 확인

```
(test2) mgh@ese-System-Product-Name:~$ sudo lshw -c display
[sudo] mgh 암호:
*-display
  description: VGA compatible controller
  product: NVIDIA Corporation
  vendor: NVIDIA Corporation
  physical id: 0
  bus info: pci@0000:01:00.0
  version: a1
  width: 64 bits
  clock: 33MHz
  capabilities: pm msi pciexpress vga_controller bus_master cap_list rom
  configuration: driver=nvidia latency=0
  resources: iomemory:400-3ff iomemory:400-3ff irq:188 memory:84000000-84ffffff memory:4000000000-400ffffff memory:4010000000-4011ffffff ioport:6000(size=128) memory:85000000-8507ffff
*-graphics
  product: EFI VGA
  physical id: 2
  logical name: /dev/fb0
  capabilities: fb
  configuration: depth=32 resolution=1920,1080
```

- RTX 4090의 capability가 8.9 이기 때문에 GPUs supported를 참고하여 toolkit 12.2 version을 설치함.

GeForce Notebook Products

GPU	Compute Capability
<u>GeForce RTX 4090</u>	8.9
<u>GeForce RTX 4080</u>	8.9

GPUs supported [\[edit \]](#)

Supported CUDA Compute Capability versions for CUDA SDK version and Microarchitecture (by code name):

Compute Capability (CUDA SDK support vs. Microarchitecture)												
CUDA SDK Version(s)	Tesla	Fermi	Kepler (Early)	Kepler (Late)	Maxwell	Pascal	Volta	Turing	Ampere	Ada Lovelace	Hopper	Blackwell
1.0 ^[36]	1.0 – 1.1											
1.1	1.0 – 1.1+x											
2.0	1.0 – 1.1+x											
2.1 – 2.3.1 ^{[37][38][39][40]}	1.0 – 1.3											
3.0 – 3.1 ^{[41][42]}	1.0	2.0										
3.2 ^[43]	1.0	2.1										
4.0 – 4.2	1.0	2.1										
5.0 – 5.5	1.0			3.5								
6.0	1.0		3.2	3.5								
6.5	1.1			3.7	5.x							
7.0 – 7.5		2.0			5.x							
8.0		2.0				6.x						
9.0 – 9.2			3.0				7.0 – 7.2					
10.0 – 10.2			3.0					7.5				
11.0 ^[44]				3.5					8.0			
11.1 – 11.4 ^[45]				3.5					8.6			
11.5 – 11.7.1 ^[46]				3.5					8.7			
11.8 ^[47]				3.5						8.9	9.0	
12.0 – 12.4					5.0						9.0	

CUDA Toolkit 12.2 Downloads

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System

LinuxWindows

Architecture

x86_64ppc64learm64-sbsaaarch64-jetson

Distribution

CentOSDebianFedoraKylinOSOpenSUSERHELRockySLESUbuntu

Distribution

WSL-Ubuntu

Version

20.0422.04

Installer Type

deb (local)deb (network)runfile (local)

```
$ wget https://developer.download.nvidia.com/compute/cuda/12.
$ sudo sh cuda_12.2.0_535.54.03_linux.run
```


그 후 cuDNN을 설치

cuDNN Archive

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

Download cuDNN v8.9.7 (December 5th, 2023), for CUDA 12.x

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

[Local Installer for Windows \(Zip\)](#)

[Local Installer for Linux x86_64 \(Tar\)](#)

[Local Installer for Linux PPC \(Tar\)](#)

[Local Installer for Linux SBSA \(Tar\)](#)

[Local Installer for Debian 11 \(Deb\)](#)

[Local Installer for Ubuntu20.04 x86_64 \(Deb\)](#)

[Local Installer for Ubuntu22.04 x86_64 \(Deb\)](#)

[Local Installer for Ubuntu20.04 aarch64sbsa \(Deb\)](#)

[Local Installer for Ubuntu22.04 aarch64sbsa \(Deb\)](#)

[Local Installer for Ubuntu20.04 cross-sbsa \(Deb\)](#)

[Local Installer for Ubuntu22.04 cross-sbsa \(Deb\)](#)

cuDNN v8.9.7중 Local install for Linux x86_64(Tar) 다운로드

```
cd ~/다운로드
```

```
tar xvf cudnn-linux-x86_64-8.6.0.163_cuda11-archive.tar.xz
```

```
sudo cp cudnn-*-archive/include/cudnn*.h /usr/local/cuda/include
```

```
sudo cp -P cudnn-*-archive/lib/libcudnn* /usr/local/cuda/lib64
```

```
sudo chmod a+r /usr/local/cuda/include/cudnn*.h /usr/local/cu
```

다운로드 받은 위치로 이동 후 압축을 풀어줌. 그 후 필요 파일들을 복사함

```
sudo nano ~/.bashrc
```

```
export PATH="/usr/local/cuda-12.2/bin:$PATH"
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.2/
```

.bashrc 파일을 열어서 cuda Toolkit 관련 설정을 환경변수에 추가함.

nvidia-smi와 nvcc -V 명령어로 그래픽 카드 드라이버 버전과 설치 가능한 최대 CUDA 버전을 확인하고 cuda toolkit이 제대로 설치된 것을 확인

```
(test2) mgh@ese-System-Product-Name:~$ nvidia-smi
Tue Apr  9 04:51:42 2024

+-----+
| NVIDIA-SMI 535.161.07                Driver Version: 535.161.07   CUDA Version: 12.2     |
+-----+-----+-----+-----+-----+-----+
| GPU    Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan    Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|                               |                      |              MIG M. |
+=====+=====+=====+=====+=====+=====+
|   0   NVIDIA GeForce RTX 4090                Off | 00000000:01:00.0  On |          Off |
|  0%    25C    P8               16W / 450W |  419MiB / 24564MiB |      0%      Default |
|                               |                      |              N/A |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                                |
| GPU   GI    CI          PID    Type    Process name                  GPU Memory |
|          ID    ID                                   |          Usage   |
+=====+=====+=====+=====+=====+=====+
|   0   N/A   N/A         1140     G   /usr/lib/xorg/Xorg              35MiB |
|   0   N/A   N/A         1744     G   /usr/libexec/gnome-initial-setup  6MiB |
|   0   N/A   N/A         2617     G   /usr/lib/xorg/Xorg              64MiB |
|   0   N/A   N/A         2760     G   /usr/bin/gnome-shell            125MiB |
+-----+-----+-----+-----+-----+-----+

```

```
(test2) mgh@ese-System-Product-Name:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Jun_13_19:16:58_PDT_2023
Cuda compilation tools, release 12.2, V12.2.91
Build cuda_12.2.r12.2/compiler.32965470_0

```

추후 계획

coco dataset을 yolo에 맞게 변환 해 주고, pretrained를 사용하지 않고, 처음부터 끝까지 weight값들을 학습시킬 계획