



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지 서비스
팀 명	YOLO(You Only Live Once)
문서 제목	중간보고서

Version	1.5
Date	2020-04-21

팀원	이로제 (조장)
	이명학
	박은환
	이재빈
	정진우
	주가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

지도교수	이재구
-------------	-----

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “인공지능을 적용한 실시간 적외선 이미지 객체 탐지 서비스”를 수행하는 팀 “YOLO”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “YOLO”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간보고서-인공지능을 적용한 적외선 이미지 실시간 객체 탐지 서비스.doc
원안작성자	이로제, 이명학, 박은환, 정진우, 이재빈
수정작성자	이로제,

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-04-17	이재빈, 정진우	1.0	최초 작성	웹 파트 작성
2020-04-17	이로제	1.0	최초 작성	TX2 부분 작성
2020-04-17	이명학, 박은환	1.0	최초 작성	딥러닝 모델부분 작성
2020-04-20	이로제	1.1	내용 수정	1 ~ 6 전체 1차 수정
2020-04-21	이로제	1.2	내용 추가	4.2
2020-04-22	이재빈	1.3	내용 추가	3.1, 4.2
2020-04-23	이명학	1.4	내용 추가	3.1
2020-04-23	이로제	1.5	내용 수정	1 ~ 6

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

목 차

1	프로젝트 목표	4
1.1	목표	4
1.2	기대 적용 분야	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.2	수행내용	5
2.2.1	딥러닝 모델	5
2.2.2	TX2 board	6
2.2.3	웹 서비스	9
3	수정된 연구내용 및 추진 방향	10
3.1	수정사항	10
3.1.1	딥러닝 모델	10
3.1.2	TX2 board	10
3.1.3	웹 서비스	13
4	향후 추진계획	16
4.1	계획한 일정표	16
4.2	향후 계획의 세부 내용	17
4.1.1	딥러닝 모델	17
4.1.2	TX2 board	17
4.1.3	웹 서비스	18
5	고충 및 건의사항	19
5.1	딥러닝 모델	19
5.2	TX2 board	19
5.2.1	운영체제	19
5.2.2	카메라 연동 부분	19
5.3	웹 서비스	19
6	참고문헌	20

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

1 프로젝트 목표

1.1 목표

본 프로젝트의 목표는 심층학습(deep learning)을 활용하여 적외선 카메라로 촬영되는 이미지 혹은 영상으로부터 실시간으로 객체를 탐지하여 제공하는 서비스를 개발하는 것이다.

더불어 이러한 서비스를 여러 다양한 실생활에 적용이 가능하도록 연구, 개발하는 것 또한 목표로 하고 있다.

1.2 기대 적용 분야



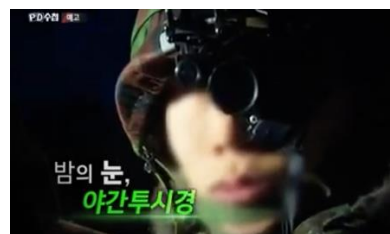
[적외선 CCTV]




[안심귀가 서비스]



[자율주행]



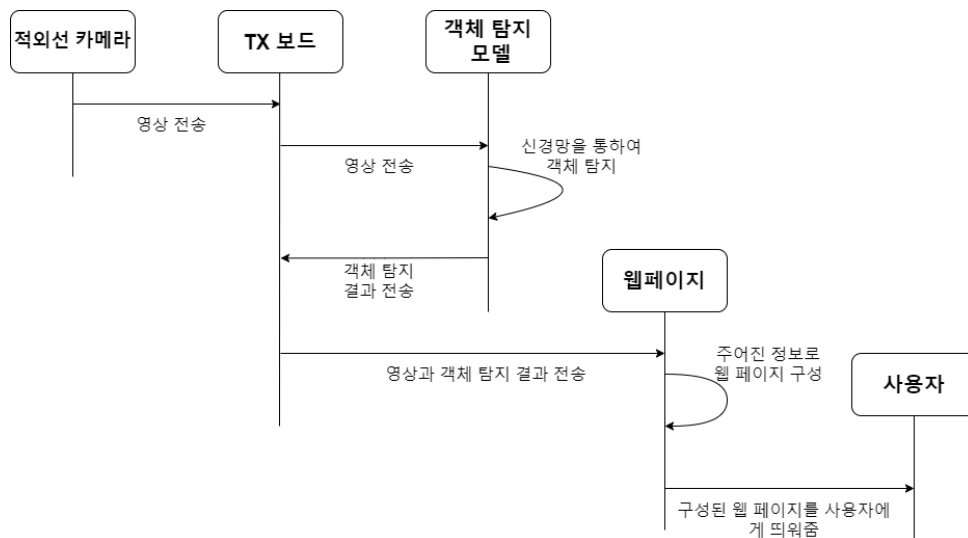
[군부대]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

적외선 카메라를 통해 실시간으로 촬영을 하고 실시간 촬영 영상을 임베디드 컴퓨팅 장치에 전송하여 심층학습을 통한 객체 탐지 결과를 웹을 통해 보여주는 총 네 단계의 흐름으로 구성할 것이다.



2.2 수행내용


2.2.1 딥러닝 모델

Faster R-CNN 모델 구현

Faster R-CNN은 YOLO와 다르게 Two Stage Detector이다. Faster R-CNN은 원본 이미지를 Feature map으로 만드는 신경망, Feature map으로부터 bounding box를 치는 신경망, 그리고 마지막으로 classification 하는 신경망으로 총 3가지가 존재한다. 이렇게 3가지 신경망을 거쳐 결과가 나오므로 YOLO와는 다르게 높은 정확도를 갖지만 속도가 느리다는 단점이 있다.

도구들 작성

신경망을 학습하고 테스트하고 그리고 그 결과를 영상으로 출력하려면 여러가지 과정을 거

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

쳐야 한다. 현재 아직 원데이터(Low data)를 그래도 사용하는 과정까지는 도달하지 못하였으므로 적외선 카메라로 찍은 영상을 프레임 단위로 객체 인식을 하여 그 결과를 띄워주는 과정이 필요하다. 여기에 필요한 여러가지 코드들을 작성해 놓았다. 이는 후에도 사용이 가능하도록 최대한 확장성에 신경을 쓰면서 만들었다.

데이터 전처리

현재 우리가 가진 데이터를 그대로 학습에 적용시키는 것은 불가능하다. 따라서 이를 적절히 바꾸어주는 과정이 필요하다. 그래서 이를 수행하는 코드를 작성하였다.

학습 진행

구현된 모델을 바탕으로 학습을 진행하였고 그 결과 FLIR 데이터셋에서는 좋은 결과를 얻었지만 SeekShot-pro 데이터셋에서는 그리 좋은 결과가 나오지는 못하였다.

2.2.2 TX2 board

(1) 카메라와의 연동을 위한 환경설정

a. numpy설치

```
$ pip install numpy
```

b. opencv 설치

```
$ sudo apt-get install cmake
$ sudo apt-get install gcc g++
$ sudo apt-get install python3-dev python3-numpy
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
$ sudo apt-get install libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev
$ sudo apt-get install libgtk-3-dev

$ sudo apt-get install git
$ git clone https://github.com/opencv/opencv.git
```



```
$ mkdir build
$ cd build
$ cmake ../
$ make
#sudo make install
```

(2) 딥러닝 모델을 돌리기 위한 환경설정

a. pytorch 설치

```
$ wget https://nvidia.box.com/shared/static/ncgzus5o23uck9i5oth2n8n06k340l6k.whl -
O torch-1.4.0-cp36-cp36m-linux_aarch64.whl
$ sudo apt-get install python3-pip libopenblas-base
$ pip install Cython
$ pip install numpy torch-1.4.0-cp36-cp36m-linux_aarch64.whl
```


b. torchvision 설치

```
$ git clone -b
v0.5.0[ https://github.com/pytorch/vision ](https://github.com/pytorch/vision)torchvisi
on
$ cd torchvision
$ sudo python setup.py install
(or simply - *python setup.py install* - if you aren't the root user)
$ cd ../
```

c. matplotlib 설치

```
$ pip install matplotlib
```

d. jupyter notebook 설치 및 원격 설정

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

```
$ sudo apt-get install libzmq3-dev
$ pip install --upgrade --force-reinstall --no-cache-dir jupyter
```

```
# 원격 설정
$ jupyter notebook --generate-config
```

- 원격이 가능하도록 jupyter_notebook_config.py파일을 수정

```
$ sudo vi /home/<username>/jupyter/jupyter_notebook_config.py
```

원하는 원격 환경에 맞춰서 config파일을 수정

```
c = get_config()
c.NotebookApp.allow_origin='*'
c.NotebookApp.open_browser = False
....
```

(3) 카메라와의 연동을 위한 코드


TX2 board와 카메라 간의 연동을 위한 코드를 하단의 코드를 베이스로 하여 작성중이다.

```
import cv2

capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    ret, frame = capture.read()
    cv2.imshow("VideoFrame", frame)
    if cv2.waitKey(1) > 0: break

capture.release()
cv2.destroyAllWindows()
```



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

2.2.3 웹 서비스

TX2 board로부터 이미지를 받아온 이미지를 띄우기 위해 웹 서버를 구축해야 한다. 웹 서버를 구축하기 위해 python언어를 사용할 예정이고, python언어에서 웹 서버를 구축하기 위해 flask라는 프레임워크를 사용하여 구현할 것이다. 구축한 웹 서버에서 socket IO를 이용하여 이미지를 받아올 것이다. 마지막으로 웹 페이지는 HTML언어를 이용하여 제작할 예정이고, 페이지에 필요한 기능 및 디자인을 CSS3나 Javascript로 수행할 것이다.

처음에는 프로젝트 소개 및 프로젝트의 기본 배경지식을 소개하는 페이지를 기본 페이지로 설정하고, 네비게이션 바를 이용하여 녹화된 이미지나 영상을 보여주는 페이지나 팀원을 소개하는 페이지로 이동할 수 있도록 웹페이지를 구성할 예정이다.

현재는 html을 이용하여 웹페이지의 틀을 만드는 것까지 완료된 단계이고 디자인 요소 및 내용들을 추가하여 웹페이지 제작을 완료할 예정이다. 디자인 요소들은 팀원들과의 회의를 통해 결정된 상태이고, 중간 보고서 제출 전까지 웹페이지 제작을 완료할 것이다. 현재 tx2 board와 Flask 프레임워크로 구축한 웹 서버를 연결하는 것에 대해 공부를 하는 중이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

3 수정된 연구내용 및 추진 방향

3.1 수정사항

기존의 주제였던 '인공지능을 적용한 적외선 객체 탐지 안심 귀가 서비스'을 '인공지능을 적용한 적외선 영상 실시간 객체 탐지 기술'로 변경하였다. 즉, 특정 서비스를 제공하는 측면에 초점을 두어 프로젝트를 진행하는 것이 아니라 기술적인 측면에 중점을 두어 프로젝트를 진행하는 것으로 수정한 것이다.

인공지능을 적용하여 적외선 카메라로 촬영되는 이미지 혹은 영상을 실시간으로 객체 탐지를 수행하여 제공하는 기술에 초점을 맞춰 프로젝트를 진행한다.

3.1.1 딥러닝 모델

원래 YOLO를 이용하여 객체 인식을 하려고 하였으나 YOLO를 알려면 그 이전 모델인 R-CNN을 알아야 한다는 것을 깨닫고 우선 R-CNN에 대한 공부와 이에 관한 코드를 돌려보는 작업을 해야 하였다. 따라서 우선 R-CNN의 대표적인 모델인 Faster R-CNN에 대하여 공부를 진행하고 이에 관한 코드를 돌려보았다.


3.1.2 TX2 board

(1) 환경설정 부분

TX2 board의 스토리지는 32GB로 256GB의 SSD를 추가로 마운트하였다.

기존에는 docker를 사용하여 각 환경을 설정하려 하였으나, TX2 board에 설치된 운영체제에서는 docker hub에서 제공되고 있는 다양한 docker 이미지들을 사용하기에는 제약이 존재하여 우선은 docker의 사용을 유보하기로 하였다.

(2) 카메라와의 연동을 위한 코드 부분

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

- 기존에 사용하려고 한 코드

```

import core
import usb.util
import Tkinter
from PIL import Image, ImageTk
import numpy
from scipy.misc import toimage
import sys, os, time

# find our Seek Thermal device 289d:0010
dev = usb.core.find(idVendor=0x289d, idProduct=0x0010)
if not dev: raise ValueError('Device not found')

def send_msg(bmRequestType, bRequest, wValue=0, wIndex=0,
data_or_wLength=None, timeout=None):
    assert (dev.ctrl_transfer(bmRequestType, bRequest, wValue, wIndex,
data_or_wLength, timeout) == len(data_or_wLength))

# alias method to make code easier to read
receive_msg = dev.ctrl_transfer

def deinit():
    '''Deinit the device'''
    msg = '\x00\x00'
    for i in range(3):
        send_msg(0x41, 0x3C, 0, 0, msg)

# set the active configuration. With no arguments, the first configuration will be the
active one
dev.set_configuration()

# get an endpoint instance
cfg = dev.get_active_configuration()
intf = cfg[(0,0)]

```




```
custom_match = lambda e: usb.util.endpoint_direction(e.bEndpointAddress) ==  
usb.util.ENDPOINT_OUT  
ep = usb.util.find_descriptor(intf, custom_match=custom_match) # match the first  
OUT endpoint  
assert ep is not None  
  
# Setup device  
try:  
    msg = 'Wx01'  
    send_msg(0x41, 0x54, 0, 0, msg)  
except Exception as e:  
    deinit()  
    msg = 'Wx01'  
    send_msg(0x41, 0x54, 0, 0, msg)
```

기존의 코드로 실행하여 적용해 보았으나 그대로 사용하여 적용하기에는 수정해야 할 오류가 너무 많았다. 따라서 기존의 코드를 참고하되 이번 취지에 적합한 코드를 새로 작성할 예정이다.

- 수정하여 사용하려는 코드

```
import cv2  
  
capture = cv2.VideoCapture(0)  
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)  
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)  
  
while True:  
    ret, frame = capture.read()  
    cv2.imshow("VideoFrame", frame)  
    if cv2.waitKey(1) > 0: break
```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

```
capture.release()
cv2.destroyAllWindows()
```

위의 코드를 기반으로 하여 opencv의 VideoCapture을 사용해 연결된 카메라로부터 입력을 받아 알맞은 frame으로 끊어서 이미지로 저장을 하는 코드를 작성할 예정이다.

3.1.3 웹 서비스

지난 계획 제안서에서 위험 인물을 인공지능이 객체 탐지하고 수상한 객체를 판단해서 기록하는 것과 합리적인 판단을 인공지능에게 요구하는 것에 있어서 불가능함을 느꼈기 때문에 서비스를 제공하는 목적의 웹페이지를 만드는 것이 아니라 적외선 카메라에서의 객체 탐지 기술 자체를 보여주는 웹페이지를 만드는 것으로 수정했다. 카메라에서 적외선 이미지를 받아오면 심층학습 모델을 통해 객체 탐지를 수행하고, 수행된 결과물이 이미지 형태나 동영상 형태로 저장될 것이다. 저장된 이미지나 동영상을 우리가 구축한 웹 서버 밑에 넣어서 웹페이지에서 보여주는 형식으로 웹페이지를 구축할 예정이다.

처음에는 로그인 창을 만들어서 아이디와 비밀번호를 입력하여 들어가면 카메라를 선택하거나 추가할 수 있는 목록창과 동영상을 보여주는 2개의 창으로 구성하여 디바이스를 추가한 후에 동영상을 보여주는 페이지로 구성할 예정이다. 동영상 옆에 on/off 버튼을 두어서 on버튼으로 전환 시 객체 탐지가 된 동영상을 보여주고, off버튼으로 전환 시 객체 탐지가 되지 않은 원본 동영상을 보여줄 예정이다. 또한 설정 버튼을 두어서 누를 경우에 모달 창 형식으로 객체 탐지가 된 동영상과 객체 탐지된 것들에 대한 정보를 보여주는 창을 보여줄 것이다.

- 새로 추가할 기능
 1. 로그인 창 : 특정 사용자만 사용할 수 있도록 로그인 창을 만든다.

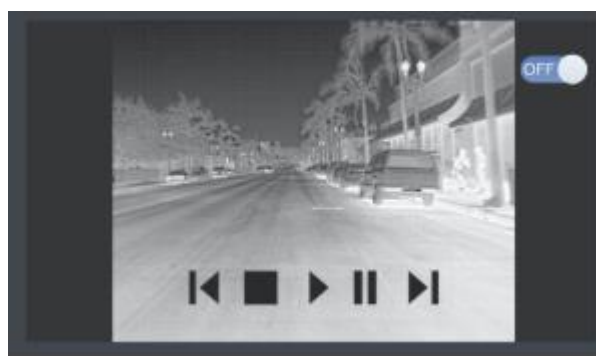
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23




2. 카메라 디바이스 목록 창 : 여러 디바이스들을 보여주는 창으로 처음에는 디바이스가 없고 사용자가 디바이스를 추가할 수 있도록 한다.




3. 동영상 플레이 창 : 객체 탐지가 안 된 동영상을 보여준다. on 버튼으로 전환 시 객체 탐지가 된 동영상을 보여준다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

4. 모달 창 : 동영상 옆에 있는 설정 버튼을 누를 시 동영상과 텍스트 창을 중심으로 다른 콘텐츠들이 희미해진다. 객체 탐지가 된 동영상에 재생되고, 동영상에서 객체의 종류 및 개수를 파악할 수 있다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23


4 향후 추진계획

4.1 계획한 일정표

세부 일정	1월	2월	3월	4월	5월	6월
프로젝트 주제 선정						
시나리오 구성						
수행 계획서 작성						
계획서 발표 준비						
개발 환경 구축						
객체 탐지 신경망 연구						
신경망 구현						
웹 페이지 설계						
웹 페이지 및 네트워크 연결 구현						
Tx 2 board와 카메라 연결						
촬영 영상 Tx 2 board에 전송						
시스템 테스트						
중간 자문 평가 준비						
전시용 자료 제작						
온라인 평가 자료 제작						
최종결과보고서 작성						

[계획한 일정표]

- 노란색(): 문서작업과 관련된 일정
- 파란색(): 주제의 구체적 구현과 관련된 일정

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

4.2 향후 계획의 세부 내용

4.1.1 딥러닝 모델

현재 객체 인식 모델을 돌린 결과 FLIR 데이터 셋에서는 훌륭한 성능을 보였지만 우리가 가진 카메라의 경우 성능이 그리 좋지 못하게 나왔다. 가장 주요한 이유로는 학습 데이터와 테스트 데이터의 격차가 커서라고 생각된다.



[Seek Shot Pro data]



[FLIR data]

4.1.2 TX2 board

세부 일정	1월	2월	3월	4월	5월	6월
Tx 2 board와 카메라 연결						
촬영 영상 Tx 2 board에 전송						

[Tx2 board 관련 일정표]

(1) TX2 board와 카메라 연결

현재 카메라와의 연동을 위한 코드를 구현 중이다. 사용하는 언어는 python이며 opencv를 통해 TX2 board가 외부 카메라를 인식하고 frame에 맞춰 입력을 읽어내도록 구현할 예정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

(2) 촬영 영상 TX2 board에 실시간 전송

(1)번에서 카메라와의 연동을 하여 성공적으로 입력신호를 받아오게 하고 입력 받은 값들을 이미지로 저장하여 딥러닝 모델에 전송할 수 있도록 코드를 구현할 예정이다.

4.1.3 웹 서비스

웹 서버에 녹화가 된 영상이나, 촬영된 이미지를 넣어서 녹화가 된 이미지 혹은 영상을 웹페이지에 띄울 것이고, 만약에 카메라 팀에서 실시간 스트리밍하는 방법을 찾아서 알려준다면, 그것을 토대로 실시간으로 적외선 이미지(혹은 영상)와 객체 탐지가 된 이미지(혹은 영상)를 웹 페이지에 띄울 것이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

5 고충 및 건의사항

5.1 딥러닝 모델

우리가 training에 사용한 데이터 셋은 FLIR에서 제공하는 데이터셋을 사용하였다. 그런데 그 데이터셋은 화질이 굉장히 좋은 반면 우리가 가진 데이터셋은 화질이 낮아 객체 탐지가 제대로 되지 않는 문제가 발생하였다.

5.2 TX2 board

5.2.1 운영체제


TX2 board는 L4T(linux for tegra)를 기반으로 Linux kernel과 bootloader, NVIDIA drivers 등이 작동이 되는데 이와 같은 환경이 기존 Ubuntu환경에 적용하고 설치하는 방법과는 달리 해야 하는 제약이 생기게 되어 이러한 부분이 프로젝트를 진행하는 데 있어 어려움이 있다.

5.2.2 카메라 연동 부분

위와 같은 운영체제의 제약으로 카메라 연동부분에 있어서도 어려움이 있다.

5.3 웹 서비스

카메라와 tx2 board를 연결하는 팀원의 조언을 들었을 때, 실시간으로 카메라에서 찍은 영상을 웹으로 전송하는 것을 구현하는 것은 현실적으로 불가능할 것 같아 보인다. 그래서 녹화된 영상 및 객체 탐지 영상을 웹에 올리는 방향으로 구현을 하는 것을 목표로 할 예정이고, 만약에 실시간으로 스트리밍이 된다면, 웹 서버에 실시간으로 적외선 이미지(혹은 영상)와 객체 탐지가 된 이미지(혹은 영상)를 띄울 예정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	인공지능을 적용한 실시간 적외선 이미지 객체 탐지	
	팀 명	YOLO(You Only Live Once)	
	Confidential Restricted	Version 1.1	2020-APR-23

6 참고문헌

번호	종류	출처	발행년도	저자	기타
1	웹 페이지	http://mediahub.seoul.go.kr/archives/1236923	2015	Joseph-Redmon	
2	웹 페이지	https://www.docker.com/			
3	웹 페이지	https://hub.docker.com/			
4	웹 페이지	https://www.nvidia.com/ko-kr/autonomous-machines/embedded-systems/jetson-tx2/			
5	웹 페이지	https://docs.nvidia.com/jetson/archives/l4t-archived/l4t-3231/index.html			
6	웹 페이지	http://www.irobotnews.com/news/articleView.html?idxno=11706			
7	웹 페이지	http://www.goodmorningcc.com/news/articleView.html?idxno=207023			
8	논문	Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks			
9	논문	Joseph Redmon , Santosh Divvala, Ross Girshick , Ali Farhadi, "You Only Look Once : Unified, Real-Time Object Detection"			