

# 이미지 기반 실시간 화재 감지 및 구조대상자 파악 AI 시스템 구현

김기영, 박혜민, 배병찬, 여예진

## Background

### 기존의 화재 감지 시스템

기존의 화재 감지 시스템인 열 탐지 기반 화재 감지 시스템은 온도 탐지 센서에 의존하기 때문에 실내 면적이나 천장의 높이 등 센서가 설치된 장소의 영향을 받는다.

### 프로젝트 목표

열 탐지 기반의 화재 인식에서 벗어나 CCTV와 같은 영상 감지 장치에 이미지 기반 화재 감지 시스템과 구조대상자를 파악하는 기능을 겸용하여 신속한 화재 감지와 구조를 가능하게 하는 것이다.

### YOLO(You Only Look Once)

YOLO는 하나의 CNN을 이용하여 이미지 전체를 여러 장으로 분할하지 않고 한 번만 보는 실시간 객체 탐지 모델이다. YOLO는 이미지 전체를 처리하기 때문에 클래스의 모양에 대한 정보 뿐만 아니라 주변 정보까지 학습하여 오류를 낮춘다. 이를 통해 불 객체가 없는 배경에서 노이즈가 발생하면 이를 불로 인식하는 오류를 막을 수 있기에 YOLO를 선정했다.

## Approach : Person detect and count

### 사람 감지 인공지능 모델 구현

- a) Darknet git repository를 yolov4 폴더에 clone 후 OPENCV, GPU, CUDNN, CUDNN\_HALF, LIBSO 설치

```
[ ] !git clone https://github.com/AlexeyAB/darknet
```

```
[ ] %cd darknet/
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!sed -i 's/LIBSO=0/LIBSO=1/' Makefile
```

- c) 객체 감지 모델 불러오기

```
[ ] # get bthe scaled yolov4 weights file that is pre-trained to detect 80 classes (objects) from shared google drive
!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&confirm=${wget} --quiet --save-cookies /tmp/cookies.txt --keep-session-cook
```

- c) 객체 감지 모델 불러오기

- d) 사람만 감지하도록 조정

```
# loop through detections and draw them on transparent overlay image
for label, confidence, bbox in detections:
    if label == "person":
        left, top, right, bottom = bbox2points(bbox)
        left, top, right, bottom = int(left * width_ratio), int(top * height_ratio), int(right * width_ratio), int(bottom * height_ratio)
        bbox_array = cv2.rectangle(bbox_array, (left, top), (right, bottom), class_colors[label], 2)
        bbox_array = cv2.putText(bbox_array, "{} {:.2f}".format(label, float(confidence)),
                                (left, top - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                                class_colors[label], 2)
        count.append([bbox])
cv2.putText(image1, f"COUNT : {len(count)}",
            (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 5,
            (255, 0, 0), 2)
label_html = f"COUNT : {len(count)}"
person_count=len(count)
count=[]
```

## Approach : Fire and smoke detect

### 화재와 연기 이미지 데이터셋 수집 및 라벨링

- a) Roboflow에서 'Fire and Smoke Training Image Dataset'을 다운로드 및 비율과 GPU의 성능을 고려하여 최종 1731개의 이미지를 훈련데이터로 선정
- b) 이미지 각각을 클래스와 좌표로 이루어진 텍스트 파일로 라벨링하여 데이터 잔처리 진행



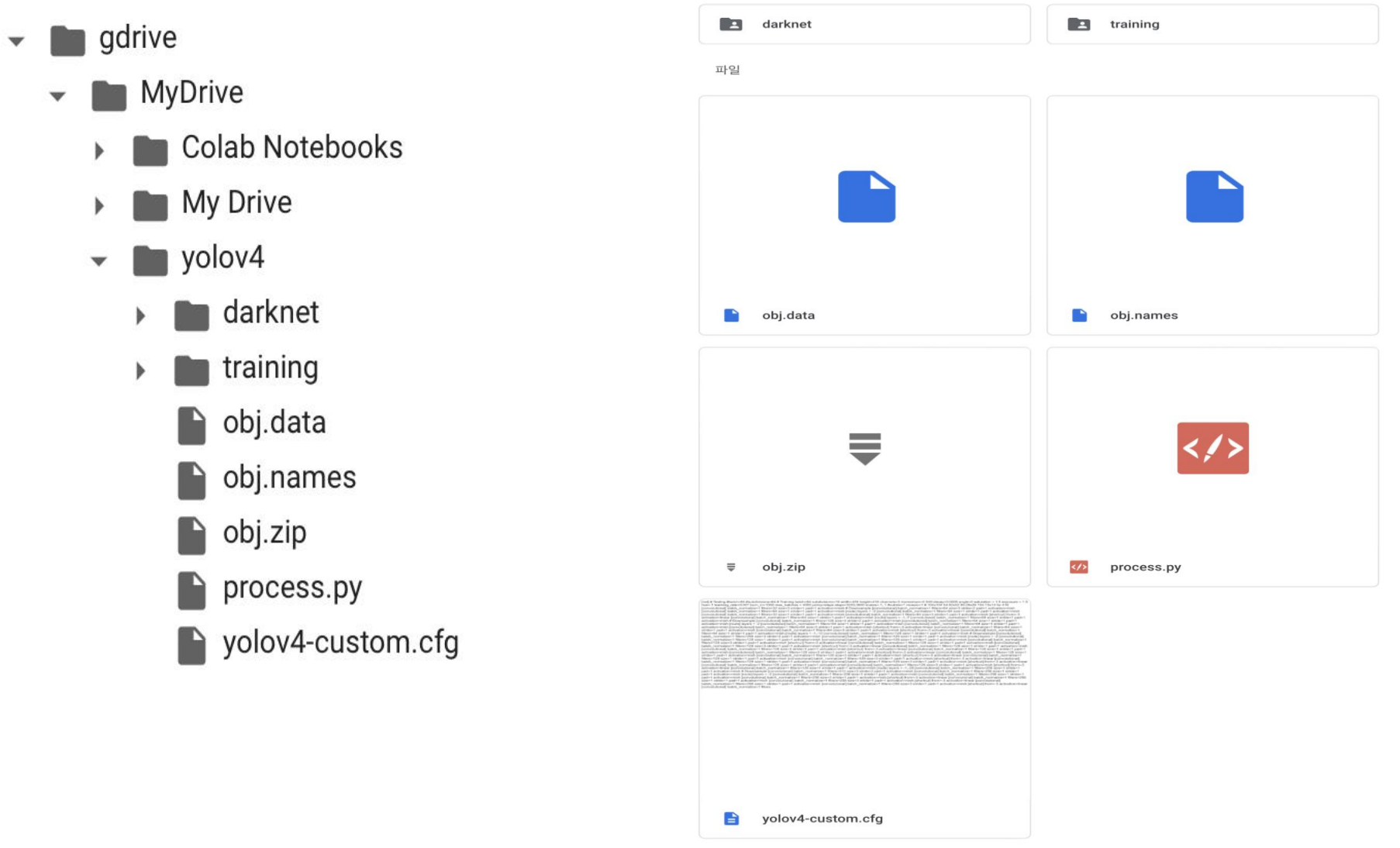
이미지파일 (.jpg파일)

```
1 0.32211538461538464
0.3918269230769231
0.6370192307692307
0.33413461538461536
0 0.6105769230769231
0.5480769230769231
0.19110576923076922 0.203125
```

텍스트파일 (.txt파일)

### 화재 감지 인공지능 모델 구현

- a) obj.data, obj.names, obj.zip, process.py, yolov4-custom.cfg파일을 구현할 모델에 맞게 수정 후 colab에 마운트



- b) Darknet git repository를 yolov4 폴더에 clone 후 OPENCV, GPU, CUDNN, CUDNN\_HALF, LIBSO 설치

```
[ ] !git clone https://github.com/AlexeyAB/darknet
```

```
[ ] %cd darknet/
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!sed -i 's/LIBSO=0/LIBSO=1/' Makefile
```

- c) 전이 학습을 위한 사전 훈련된 yolov4 weight파일 다운로드

```
!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4_conv.137
```

- d) 라벨링 된 데이터셋, cfg파일, 사전 훈련 가중치 파일로 학습 진행

```
[14] !./darknet detector train data/obj.data cfg/yolov4-custom.cfg yolov4_conv.137 -dont_show -map
```

## Results

### 두 모델 결합 및 결과

사람 인원 수 파악 -> 화재 및 연기 감지의 순서로 진행되도록 병합 및 모델 마지막에 결과를 출력하여 직관적으로 파악할 수 있게 구현함



### References

김치용, 이현수, 이광엽. (2022). 데이터 증강 학습 이용한 딥러닝 기반 실시간 화재경보 시스템 구현. 전기전자학회논문지, 26(3), 136-142.