

# YOLOv3 Custom & OpenCV

---



# CONTENTS

---



**01 Introduction**

**02 YOLOv3 Custom**

**03 Results**

**04 References**

01



Introduction

# 01 Introduction

---

## 01 Introduction

02 PyQt5

03 OpenCV

---

# Introduction

---

YOLOv3를 Custom 하여 도로 위의 사람을 인식하도록  
학습시킨 모델을 만들어서 OpenCV와 PyQt5로  
만든 GUI에 적용시켰다.

# 02 Introduction

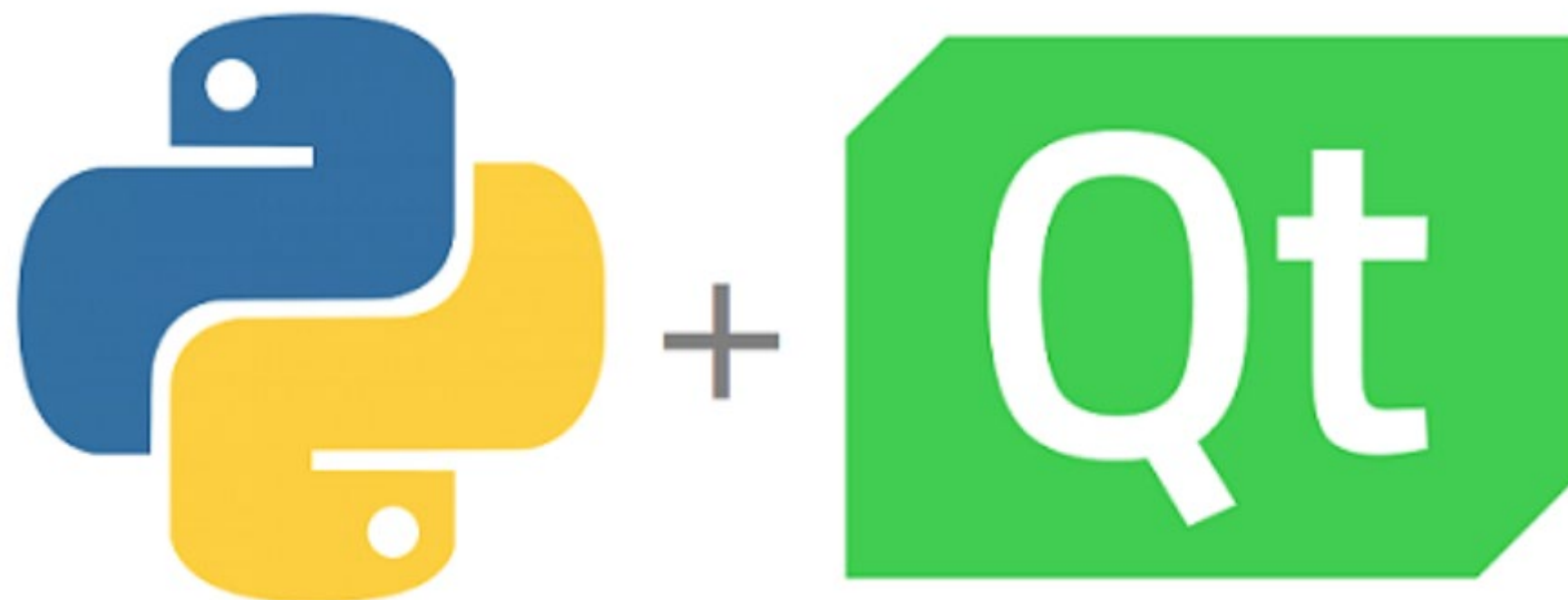
01. Introduction

**02 PyQt5**

03 OpenCV

04 YOLOv3

## PyQt5



PyQt5는 Qt5 어플리케이션 프레임워크에 대한 파이썬 버전이다.  
Qt는 플랫폼에 관계없이 다양한 기능을 포함하는 C++ 라이브러리아자 개발툴이다.

# 03 Introduction

---

01 Introduction

02 PyQt5

**03 OpenCV**

04 YOLOv3

---

## OpenCV

---



OpenCV 는 주로 실시간 컴퓨터 비전을 목표로 하는  
프로그래밍 기능 라이브러리입니다.

# 03 Introduction

---

01 Introduction

02 PyQt5

**03 OpenCV**

04 YOLOv3

---

## OpenCV 이란?

---

- 인텔에서 만든 강력한 영상처리 라이브러리.
- 기초 영상 처리에서 부터 고급 수준의 영상처리까지 많은 양의 알고리즘들이 함수로 구현되어 있다.
- 오픈소스로서 스펙만 맞추면 자신의 알고리즘도 라이브러리에 등록 시킬 수 있다.

# 03 Introduction

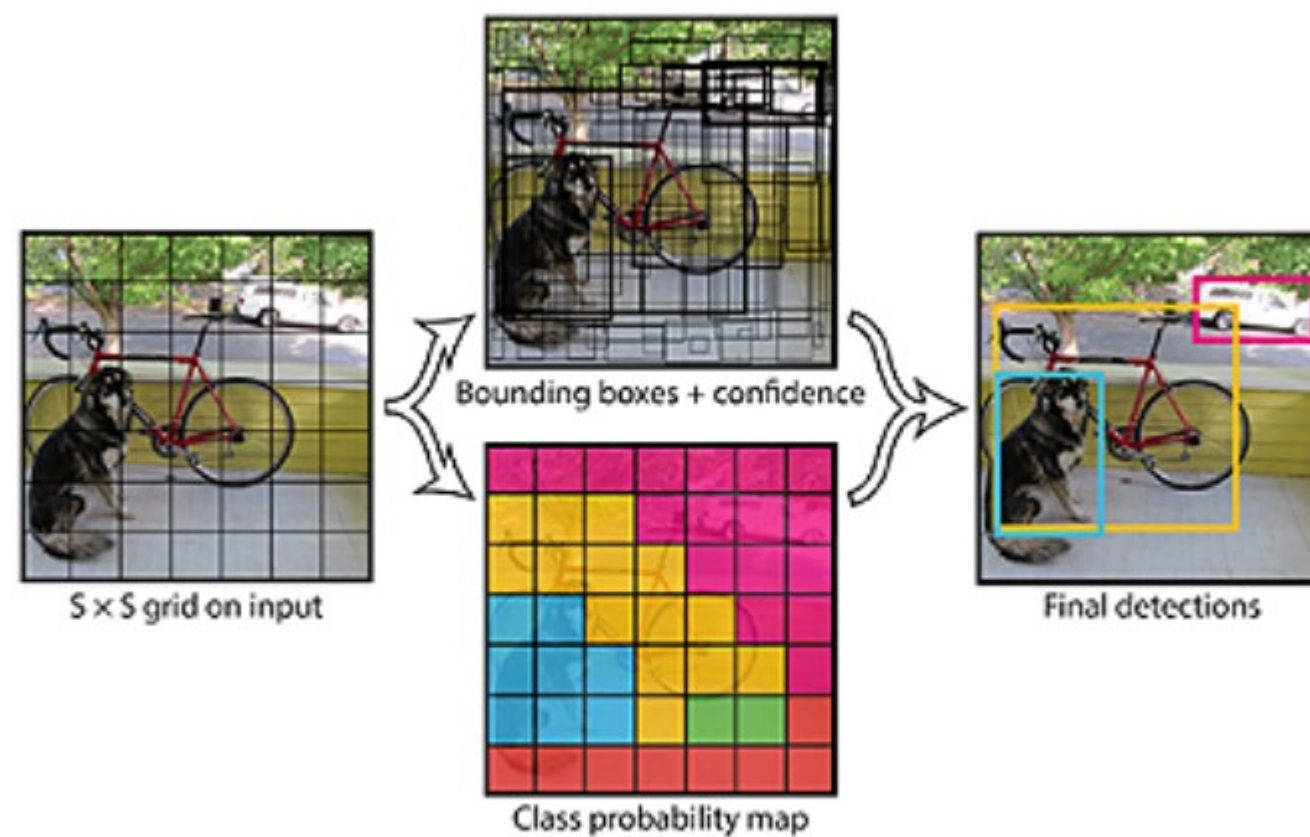
01 Introduction

02 PyQt5

03 OpenCV

**04 YOLO**

## YOLO (You Only Look Once)



<그림 1> YOLOv1 그리드

YOLO는 빠르게 이미지에서 객체를 탐지하는 모델이다.



# 03 Introduction

01 Introduction

02 PyQt5

03 OpenCV

**04 YOLO**

## YOLO (You Only Look Once)

### 장점

- 간단한 처리과정으로 속도가 매우 빠르다. 또한 기존의 다른 real-time detection system들과 비교할 때, 2배 정도 높은 mAP를 보인다.
- Image 전체를 한 번에 바라보는 방식으로 class에 대한 맥락적 이해도가 높다. 이로 인해 낮은 background error(False-Positive)를 보인다.
- Object에 대한 좀 더 일반화된 특징을 학습한다. 가령 natural image로 학습하고 이를 artwork에 테스트 했을때, 다른 Detection System들에 비해 훨씬 높은 성능을 보여준다.

### 단점

- 상대적으로 낮은 정확도 (특히, 작은 object에 대해)

02

YOLOv3 Custom

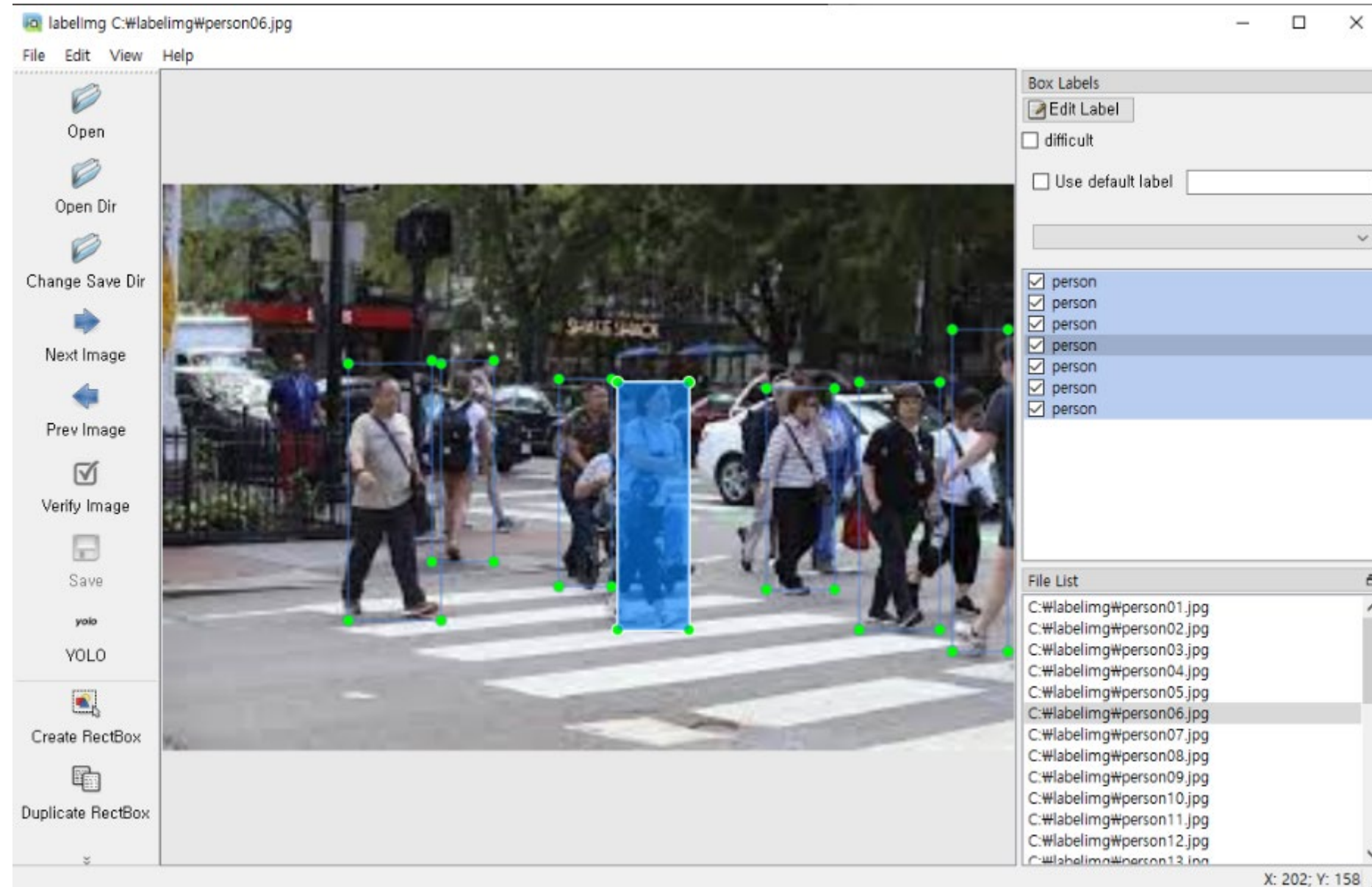
# 01 YOLOv3 Custom

## 01 Data Labeling

02 Custom Files

03 Pretrained Model

# Data Labeling



labelimg를 사용하여 약 70장의 이미지를 라벨링 하였다.

# 02 YOLOv3 Custom

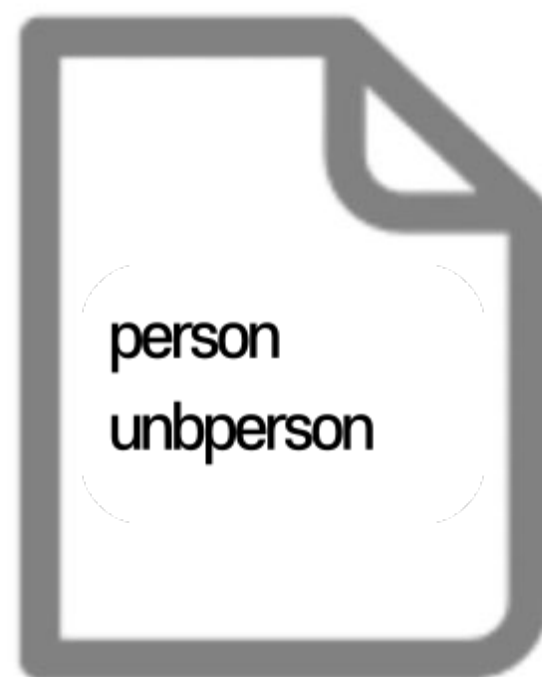
01 Data Labeling

## 02 Custom Files


03 Pretrained Model

## Custom Files

Classes.names



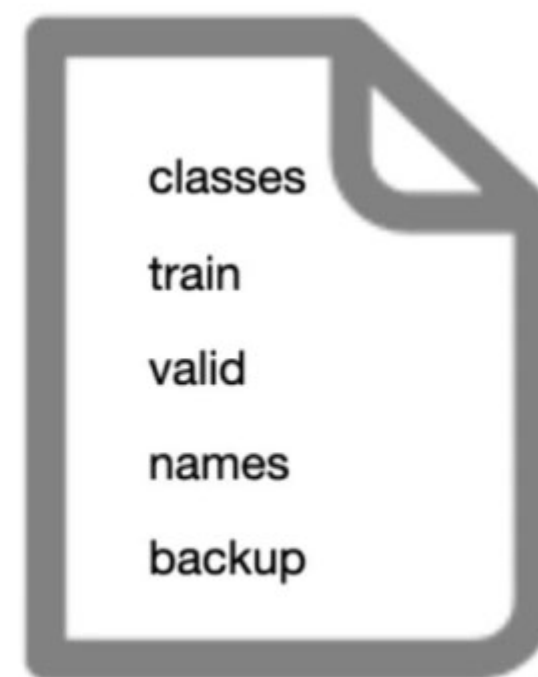
classes.names

 classes.names - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
person
unbperson
```

custom\_data.data



custom\_data.data

 custom\_data.data - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
classes = 2
train = /content/drive/MyDrive/darknet/custom/train.txt
valid = /content/drive/MyDrive/darknet/custom/test.txt
names = /content/drive/MyDrive/darknet/custom/classes.names
backup = backup
```



# 02 YOLOv3 Custom

01 Data Labeling

## 02 Custom Files

03 Pretrained Model

## Custom Files


### train.txt

 train.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
/content/drive/MyDrive/darknet/custom/person01.jpg
/content/drive/MyDrive/darknet/custom/person02.jpg
/content/drive/MyDrive/darknet/custom/person03.jpg
/content/drive/MyDrive/darknet/custom/person04.jpg
/content/drive/MyDrive/darknet/custom/person05.jpg
/content/drive/MyDrive/darknet/custom/person06.jpg
/content/drive/MyDrive/darknet/custom/person07.jpg
/content/drive/MyDrive/darknet/custom/person08.jpg
/content/drive/MyDrive/darknet/custom/person09.jpg
/content/drive/MyDrive/darknet/custom/person10.jpg
/content/drive/MyDrive/darknet/custom/person11.jpg
/content/drive/MyDrive/darknet/custom/person12.jpg
/content/drive/MyDrive/darknet/custom/person13.jpg
```

### test.txt

 test.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
/content/drive/MyDrive/darknet/custom/person25.jpg
/content/drive/MyDrive/darknet/custom/person26.jpg
/content/drive/MyDrive/darknet/custom/person27.jpg
/content/drive/MyDrive/darknet/custom/person28.jpg
/content/drive/MyDrive/darknet/custom/person29.jpg
/content/drive/MyDrive/darknet/custom/person30.jpg
/content/drive/MyDrive/darknet/custom/person31.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking20.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking21.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking22.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking23.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking24.jpg
/content/drive/MyDrive/darknet/custom/Jaywalking25.jpg
```

Train, Test 데이터셋은 8:2로 나눈다.



# 03 YOLOv3 Custom






01 Data Labeling

02 Custom Files

## 03 Pretrained Model

# Pretrained Model

PC > 바탕 화면 > pythonProject > FinalProj > weights

이름	수정한 날짜	유형	크기
 custom-train-yolo_1000.weights	2021-07-28 오전 9:19	WEIGHTS 파일	240,554KB
 custom-train-yolo_2000.weights	2021-07-28 오전 9:20	WEIGHTS 파일	240,554KB
 custom-train-yolo_3000.weights	2021-07-28 오전 9:19	WEIGHTS 파일	240,554KB
 custom-train-yolo_4000.weights	2021-07-28 오전 9:21	WEIGHTS 파일	240,554KB
 custom-train-yolo_final.weights	2021-07-28 오전 9:21	WEIGHTS 파일	240,554KB

1000, 2000, 3000, 4000 번 weights를 비교한 결과  
4000번 학습한 Weights의 결과가 가장 좋게 나왔다.

03 |

Results

# Results

```
1 # https://for-sign.tistory.com/40
2 from PyQt5 import QtCore, QtGui
3 from PyQt5 import QtWidgets, uic
4 from PyQt5.QtGui import QPixmap, QImage
5
6 import sys
7 import cv2
8 import numpy as np
9
10 from time import sleep
11 #비디오 재생을 위해 스레드 생성
12 import threading
13
14 ### output file ###
15 fourcc = cv2.VideoWriter_fourcc(*"MJPG")
16 writer = cv2.VideoWriter('output.avi', fourcc, 30, (800,600), True)
17 #####
18
19 # Yolo 로드
20 net = cv2.dnn.readNet("weights/custom-train-yolo_final.weights", "custom-train-yolo.cfg")
21 classes = []
22 with open("classes.names", "r") as f:
23     classes = [line.strip() for line in f.readlines()]
24 layer_names = net.getLayerNames()
25 output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
26 colors = np.random.uniform(0, 255, size=(len(classes), 3))
```

```
28 # 1. qt를 사용하여 GUI 프로그램 환경 구축
29 class Ui(QtWidgets.QDialog):
30     def __init__(self):
31         super(Ui, self).__init__()
32         uic.loadUi('video_load.ui', self)
33
34         self.loadBtn = self.findChild(QtWidgets.QPushButton, 'loadBtn')
35         self.loadBtn.clicked.connect(self.loadBtnClicked)
36         self.procRun = self.findChild(QtWidgets.QPushButton, 'procRun')
37         self.procRun.clicked.connect(self.procRunClicked)
38         # self.photo = self.findChild(QtWidgets.QLabel, 'photo')
39         # self.photo.setScaledContents(True)
40         self.result = self.findChild(QtWidgets.QLabel, 'result')
41         self.fnameEdit = self.findChild(QtWidgets.QLineEdit, 'fnameEdit')
42         self.fnameEdit.clear()
43         self.out_check = False
44
45         self.show()
46
47
48 def processingImage(self, img_gray, img_src):
49     # 여기에 이미지 프로세싱을 진행하고 output으로 리턴하면 오른쪽에 결과 영상 출력됨
50     # output = img_src.copy() #원본영상 그대로 리턴
51     output = img_gray.copy() # 그레이 영상 리턴
52     return output
```



# Results

```
54 def displayOutputImage(self, img_dst, mode):
55     img_info = img_dst.shape
56     if img_dst.ndim == 2:
57         qImg = QImage(img_dst, img_info[1], img_info[0], img_info[1] * 1, QImage.Format_Grayscale8)
58     else:
59         qImg = QImage(img_dst, img_info[1], img_info[0], img_info[1] * img_info[2], QImage.Format_BGR888)
60
61     self.pixmap = QtGui.QPixmap(qImg)
62     self.p = self.pixmap.scaled(600, 450, QtCore.Qt.IgnoreAspectRatio) # 프레임 크기 조정
63
64     # 결과 영상만
65     self.result.setPixmap(self.p)
66     self.result.update()
67
68     # if mode == 0:
69     #     self.photo.setPixmap(self.p)
70     #     self.photo.update() # 프레임 띄우기
71     # else:
72     #     self.result.setPixmap(self.p)
73     #     self.result.update() # 프레임 띄우기
74
75     sleep(0.01) # 영상 1프레임당 0.01초로 이걸로 영상 재생속도 조절하면됨 0.02로하면 0.5배속인거임
76
77 def procRunClicked(self):
78     self.out_check = True
```

```
82 def loadBtnClicked(self):
83     path = 'figure/'
84     filter = "All Videos(*.mp4; *.mov; *.avi);;MOV (*.mov);;MP4(*.mp4);;AVI(*.avi)"
85     fname = QtWidgets.QFileDialog.getOpenFileName(self, "파일로드", path, filter)
86     self.filename = str(fname[0])
87     self.fnameEdit.setText(self.filename)
88     self.video_thread()
89
90
91 def Video_to_frame(self, MainWindow):
92     cap = cv2.VideoCapture(self.filename)
93
94     while True:
95         self.ret, self.frame = cap.read() #영상의 정보 저장
96         if self.ret:
97             if self.out_check == True:
98                 self.process_result()
99                 self.displayOutputImage(self.frame_out, 1)
100             else:
101                 self.displayOutputImage(self.frame, 0)
102         else:
103             break
104
105     cap.release()
106     cv2.destroyAllWindows()
```

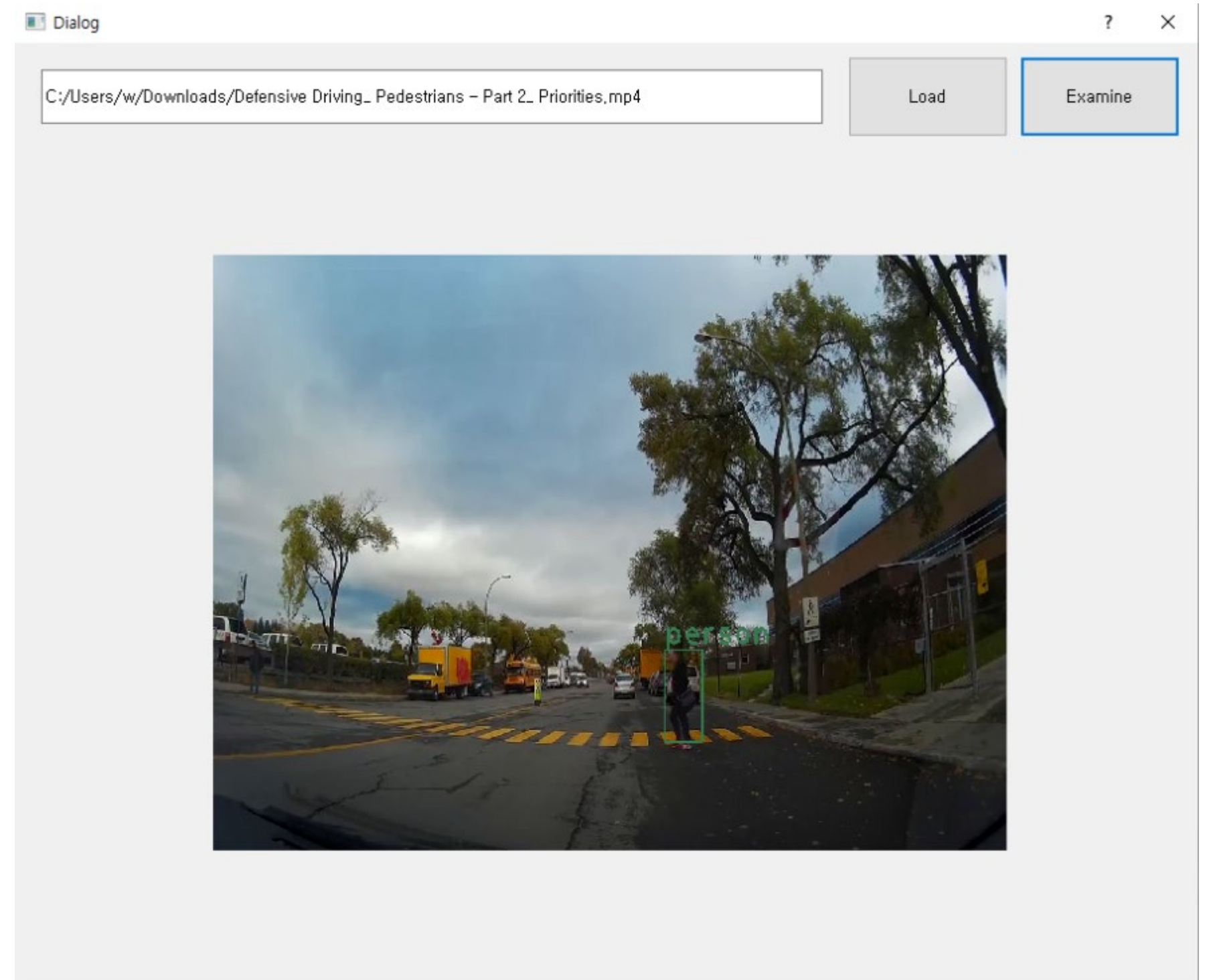
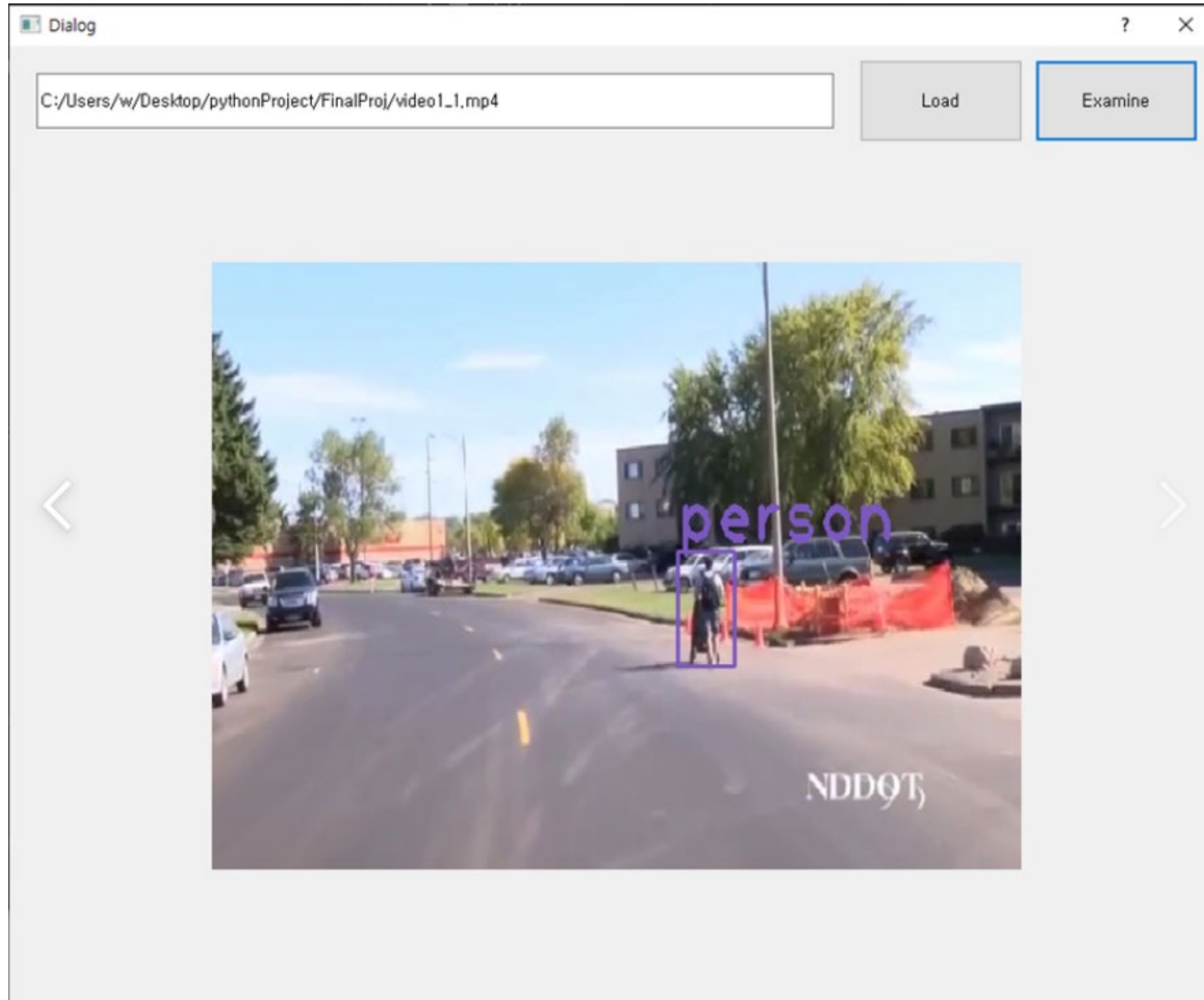
# Results

```
108 def process_result(self):
109     #self.frame_out = cv2.cvtColor(self.frame, cv2.COLOR_BGR2GRAY)
110     self.frame_out = self.frame.copy()
111     # 여기부터 작업할 코드 작성하면 됩니다.
112     # self.frame_out = cv2.resize(self.frame, None, fx=0.4, fy=0.4)
113     height, width, channels = self.frame.shape
114     # Detecting objects
115     blob = cv2.dnn.blobFromImage(self.frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
116     net.setInput(blob)
117     outs = net.forward(output_layers)
118
119     class_ids = []
120     confidences = []
121     boxes = []
122     for out in outs:
123         for detection in out:
124             scores = detection[5:]
125             class_id = np.argmax(scores)
126             confidence = scores[class_id]
127             if confidence > 0.4:
128                 # Object detected
129                 center_x = int(detection[0] * width)
130                 center_y = int(detection[1] * height)
131                 w = int(detection[2] * width)
132                 h = int(detection[3] * height)
133                 # 좌표
134                 x = int(center_x - w / 2)
135                 y = int(center_y - h / 2)
136                 boxes.append([x, y, w, h])
137                 confidences.append(float(confidence))
138                 class_ids.append(class_id)
```

```
140 indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
141 font = cv2.FONT_HERSHEY_PLAIN
142 for i in range(len(boxes)):
143     if i in indexes:
144         x, y, w, h = boxes[i]
145         label = str(classes[class_ids[i]])
146         color = colors[i]
147         cv2.rectangle(self.frame_out, (x, y), (x + w, y + h), color, 2)
148         cv2.putText(self.frame_out, label, (x, y - 5), font, 3, color, 3)
149
150     ### output file ###
151     writer.write(cv2.resize(self.frame_out, (800, 600)))
152     #####
153
154 def video_thread(self):
155     thread = threading.Thread(target=self.Video_to_frame, args=(self,))
156     thread.daemon = True # 프로그램 종료시 프로세스도 함께 종료 (백그라운드 재생 X)
157     thread.start()
158
159
160 app = QtWidgets.QApplication(sys.argv)
161 window = Ui()
162 app.exec_()
```



# Results



04 |

참조

# References

---

- [1] OpenCV. (2021, June 16). OpenCV. <https://opencv.org/>
- [2] PyQt5. (2021, March 10). PyPI. <https://pypi.org/project/PyQt5/>
- [3] Python으로 OpenCV를 사용하여 YOLO Object detection. (2020, April 13). tistory. <https://bong-sik.tistory.com/16>

---

감사합니다

---