Helmet-Detection Using YOLO-V5

DLIP_Lab4 , Handong Global University, School of Mechanical and Control Engineering.

Helmet Detection Overview

Helmet Detection Using YOLO-V5

1. YOLO-V5 Environment Setting

Anaconda Environment Creation.

Create conda env

```
conda create -n yolov5 python=3.8
conda activate yolov5
```

Clone Github.

Clone Github to specific folder to develop program.

- Github link : https://github.com/SangHoon-Lee97/DLIP_Lab4_Helmet_Detection
- Original github source : https://github.com/ultralytics/yolov5

```
git clone https://github.com/SangHoon-Lee97/DLIP_Lab4_Helmet_Detection
```

Install all Dependency and Requirements

```
# update
conda update -yn base -c defaults conda
```

```
# install Lib for YOLOv5
conda install -c anaconda cython numpy pillow scipy seaborn pandas requests
conda install -c conda-forge matplotlib pyyaml tensorboard tqdm opencv

# install pytorch
conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch

# install serial for serial Communication using Arduino
pip install serial

# install pygame for warning alarm
pip install pygame

# Extra
conda install -c conda-forge onnx
```

(Additional Tips)

When ' ERROR: Invalid requirement:"opencv-python" ' occurs

- This error can also occur when opencv-python is appropriately installed.
- Comment line where 'check_requirements' function is used

2. Download Pre-Trained Model

Pre-Trained weight file download

- Github address

https://github.com/PeterH0323/Smart_Construction

- Related Blog

https://chowdera.com/2020/11/20201116172628738h.html

- Weight File Download link : helmet_head_person_s.pt file

https://github.com/SangHoon-Lee97/DLIP_Lab4_Helmet_Detection

Pre-Train Model Performance

The performance of a custom model trained based on YOLO-v5s is as follows. (epoch =50)

Classify	Precision	Recall	mAP
Person	0.846	0.893	0.877
Helmet	0.889	0.883	0.871
Head	0.917	0.921	0.917

3. System Structure

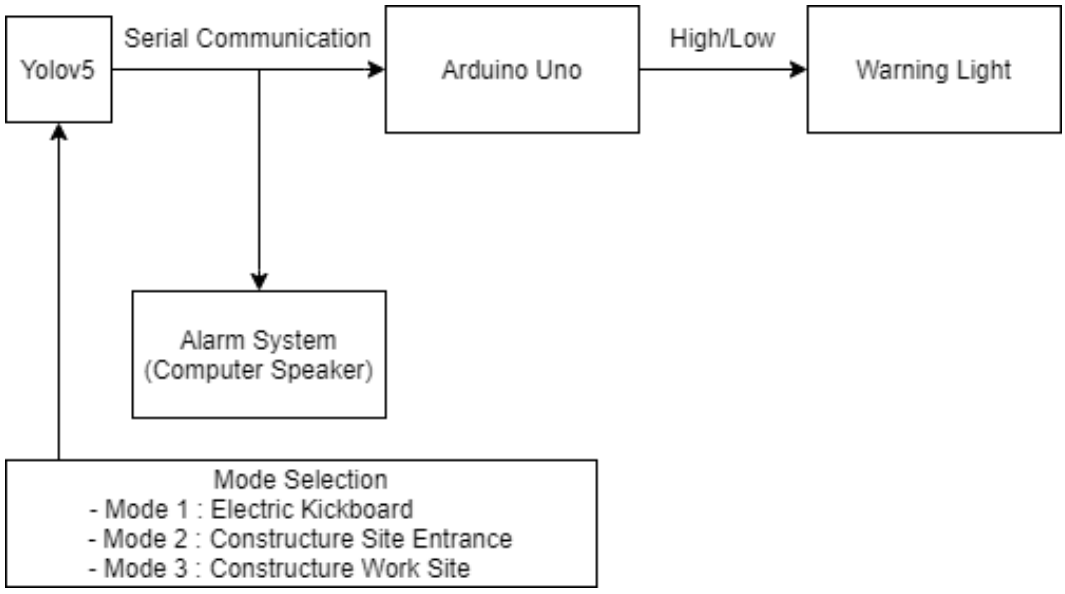


Figure above are System Structure of a program. First we choose mode for certain application. Then we use YOLO-v5 custom model to detect head , helmet and person. With detection result, we post process dected result. We have designed to give alarm and give warning light. Computer speaker is used in alarming system and serial communication with Arduino is used to give high and low voltages to turn on/off the warning light.

4. Modified Code Descriptions From Original Source Code (Without Post Processing)

- Removed Code From Yolov5 in Ultralytics

1. Instead of using opt mode in the existing code, the associated parameter were set up using the argument at the bottom of the code to simplify it as follows

```
@torch.no_grad()
def detect(weights='yolov5s.pt',  # model.pt path(s)
           source='data/images',  # file/dir/URL/glob, 0 for webcam
           imgsz=640,  # inference size (pixels)
           conf_thres=0.25,  # confidence threshold
           iou_thres=0.45,  # NMS IOU threshold
           max_det=1000,  # maximum detections per image
           device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
           view_img=False,  # show results
           save_txt=False,  # save results to *.txt
           save_conf=False,  # save confidences in --save-txt labels
           save_crop=False,  # save cropped prediction boxes
           nosave=False,  # do not save images/videos
           classes=None,  # filter by class: --class 0, or --class 0 2 3
           agnostic_nms=False,  # class-agnostic NMS
           augment=False,  # augmented inference
           update=False,  # update all models
           project='runs/detect',  # save results to project/name
           name='exp',  # save results to project/name
           exist_ok=False,  # existing project/name ok, do not increment
           line_thickness=3,  # bounding box thickness (pixels)
           hide_labels=False,  # hide labels
           hide_conf=False,  # hide confidences
           half=False,  # use FP16 half-precision inference
           ):
    save_img = not nosave and not source.endswith('.txt')  # save inference images
    webcam = source.isnumeric() or source.endswith('.txt') or source.lower().startswith(
        ('rtsp://', 'rtmp://', 'http://', 'https://'))
```

```
def detect(opt):
    source, weights, view_img, save_txt, imgsz = opt.source, opt.weights, opt.view_img, opt.save_txt, opt.img_size
    save_img = not opt.nosave and not source.endswith('.txt')  # save inference images
    webcam = source.isnumeric() or source.endswith('.txt') or source.lower().startswith(
        ('rtsp://', 'rtmp://', 'http://', 'https://'))
```

- Added Code From Yolov5 in Ultralytics

1. Bring the packages for serial communication, timeit for FPS calculation, and mixer for speaker output.

```
import serial
import numpy as np
from timeit import default_timer as timer
from pygame import mixer
```

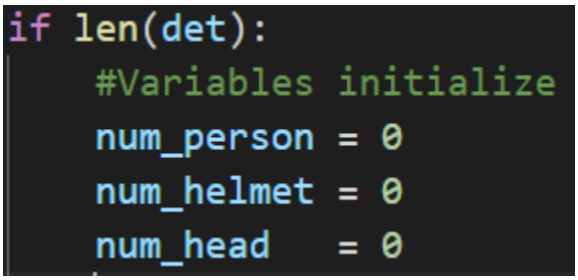
2. Below variables must be declared before the entire detecting for statement is operated.

```
t0 = time.time()
buf_person = []
buf_helmet = []
buf_head = []
buf_time   = []
idx2 = 0
accum_time = 0
curr_fps   = 0
prev_time  = timer()
fps_my     = "FPS: ??"
out_Text   = ''
out_Text2  = ''
Warning_Check = 0
Warning_Check2 = 0
Safe_Check = 0
SoundFlag1 = 0
SoundFlag2 = 0
play_count1 = 0
play_count2 = 0
use_arduino = True
```

If you want to run the code without using Arduino for serial communication, change 'use_arduino' to False

3. Variables for measuring the number of people, helmets, and heads. It must be declared after

len(det) to be updated and used in Mode



5. Code Descriptions : Post Processing

Mode 1. Electrical Kickboard

```
if(opt.mode == '0'):
    if (num_person > 1):
        text_for_warning = "Warning! Only 1 Person Can Ride"
        if(num_person == num_head):
            text_for_warning = "Warning! Only 1 Person Can Ride. No Helmet Detected"
        elif(num_head>0 and num_person == num_helmet+num_head):
            text_for_warning = "Warning! Only 1 Person Can Ride. More Helmet needed!"
        elif(num_head == 0 and num_person == num_helmet + num_head):
            text_for_warning = "Warning! Only 1 Person Can Ride."
        text_color = (0,0,255)
        Warning_Check += 1
        Warning_Check2 += 1

        if(Warning_Check > 10):
            Warning_FLAG = 1
            Warning_Check = 0
            Safe_Check = 0
        if(Warning_Check2 > 200):
            Warning_FLAG = 1
            Warning_Check2 =0
            Safe_Check = 0
            SoundFlag1 = 0
            SoundFlag1 = 0
            play_count1 = 0
            play_count2 = 0

    elif(num_person == 1):
        if(num_person == num_head):
            text_for_warning = "Warning! No Helmet Detected!"
            Warning_Check += 1
            Warning_Check2 += 1
```

```
            if(Warning_Check > 10):
                Warning_FLAG = 1
                Warning_Check = 0
                Safe_Check = 0
            if(Warning_Check2 > 200):
                Warning_FLAG = 1
                Warning_Check2 =0
                Safe_Check = 0
                SoundFlag1 = 0
                SoundFlag1 = 0
                play_count1 = 0
                play_count2 = 0
            text_color = (0,0,255)

        elif(num_person == num_helmet):
            text_for_warning = "Good! You are Safe"
            Safe_Check += 1
            if(Safe_Check > 10):
                Warning_FLAG = 0
                Safe_Check = 0
                Warning_Check = 0
            text_color = (0,255,0)
```

This Code above is post processing code used only in mode 1(In the Code it is mode 0).

- The code consist of two big condition. It depends on number of person detected. Since Electric kickboard is illegal when more than one person is using same one. If-condition decides whether the number of person detected is 1 or more than 1.
- When more than one person is detected, it is designed to always give warning alarm. But more specifically it has 2 more conditions to consider.

  It gives "Warning! Only 1 Person Can Ride. No Helmet Detected" warning message when number of person is equal to number of head detected. Also, "Warning! Only 1 Person Can Ride. More Helmet needed!" warning message is given when number of person is not equal to number of head but more than one head is detected. Finally it gives "Warning! Only 1 Person Can Ride." when number of person is equal to number of helmet detected.
- When only one person is detected, it gives "Warning! No Helmet Detected!" message when number of person is equal to number of head detected. On the

other hand it gives "Good! You are Safe" message when number of person is equal to number of helmet detected.

- For both two big conditions, it gives warning flag or safety flag to trigger alarm and warning light system. Also when situation transition happens, for example warning condition to safety condition and vise versa, it initializes sound flag that is used to trigger alarm and various 'counts' to consider the changes of a situation.

Mode 2. Construction Site Entrance

```
elif(opt.mode == '1'):
    if (num_person > 1):
        text_for_warning = "Warning! Enter One By One"
        text_color = (0,0,255)
        Warning_Check += 1
        if(Warning_Check > 10):
            Warning_FLAG = 1
            Warning_Check = 0
            Safe_Check = 0

    elif(num_person == 1):
        if(num_person == num_head):
            text_for_warning = "Warning! No Helmet Detected!"
            Warning_Check += 1
            if(Warning_Check > 10):
                Warning_FLAG = 1
                Warning_Check = 0
                Safe_Check = 0
            text_color = (0,0,255)

        elif(num_person == num_helmet):
            text_for_warning = "Good! You are Safe"
            Safe_Check += 1
            if(Safe_Check > 10):
                Warning_FLAG = 0
                Safe_Check = 0
                Warning_Check = 0
            text_color = (0,255,0)
```

This Code above is post processing code used only in mode 2(In the Code it is mode 1).

- Severe injuries caused by not wearing helmets at workplace for 36%. Therefore, this code detects and allows people entering the workplace to

enter wearing helmets. This code has 2 big conditions. If-condition decides whether the number of person detected is 1 or more than 1.

- When more than one person is detected, it is designed to always give warning alarm. But more specifically it has 2 more conditions to consider.

  It gives "Warning! Enter One By One" warning message when number of person is equal to number of head detected. Also, "Warning! No Helmet Detected!" warning message is given when number of person is equal to number of head but more than one head is detected.

- When only one person is detected, it gives "Warning! No Helmet Detected!" message when number of person is equal to number of head detected. On the other hand it gives "Good! You are Safe" message when number of person is equal to number of helmet detected.

- For both two big conditions, it gives warning flag or safety flag to trigger alarm and warning light system. Also when situation transition happens, for example warning condition to safety condition and vise versa, it initializes sound flag that is used to trigger alarm and various 'counts' to consider the changes of a situation.

Mode 3. Construction Work Site

```
elif(opt.mode == '2'):
    if (num_person > 1):
        if(num_person == num_helmet):
            text_for_warning = "Good! This Area is Safe"
            Safe_Check +=1
            if(Safe_Check > 10):
                Warning_FLAG = 0
                Safe_Check = 0
                Warning_Check = 0
            text_color = (0,255,0)
        else:
            Danger_workers = num_person -num_helmet
            text_for_warning = "Warning! "+str(Danger_workers)+" Workers In Danger!"
            Warning_Check += 1
            if(Warning_Check > 10):
                Warning_FLAG = 1
                Warning_Check = 0
                Safe_Check = 0
            text_color = (0,0,255)

    elif(num_person == 1):
```

```
        if(num_person == num_head):
            text_for_warning = "Warning! No Helmet Detected!"
            Warning_Check += 1
            if(Warning_Check > 10):
                Warning_FLAG = 1
                Warning_Check = 0
                Safe_Check = 0
            text_color = (0,0,255)

        elif(num_person == num_helmet):
            text_for_warning = "Good! This Area is Safe"
            Safe_Check += 1
            if(Safe_Check > 10):
                Warning_FLAG = 0
                Safe_Check = 0
                Warning_Check = 0
            text_color = (0,255,0)

out_Text  = "Helmet Number: " + str(num_helmet) + " Person Number: "+str(num_person)
out_Text2 = "Status: " + text_for_warning
```

This Code above is post processing code used only in mode 3. (In the Code it is mode 2).

- Like the other modes above, there are two major conditions. Above Mode 2 is a system that operates at the entrance of an industrial site, but Mode 3 is a code that detects helmets based on industrial sites.
- There are two big conditions, whether the number of person detected is 1 or more than 1. When there are more people than one, the number of helmets and the number of people working are the same, this code shows "Good! This Area is Safe"

  Also it warned you if the number of helmets is less than the number of people, it gives you "Warning! "+(Danger_workers)+" Workers In Danger!"
- When only one person is detected and the number of helmet and the number of person are the same, this mode shows you "Good! This Area is safe!"
- Also it warned you if the number of head equals the number of people warned you "Warning! No Helmet Detected!"
- For both two big conditions, it gives warning flag or safety flag to trigger alarm and warning light system. Also when situation transition happens, for example warning condition to safety condition and vise versa, it initializes sound flag

that is used to trigger alarm and various 'counts' to consider the changes of a situation.

Alarm System (All mode in Common)

```
if(Warning_FLAG == 1) :
    SoundFlag1 = 1
    play_count1 +=1
    if(play_count1>3):
        alert.stop()
    else:
        if(SoundFlag2 == 1):
            alert.stop()
            alert = mixer.Sound('Danger.wav')
            alert.set_volume(0.7)
            alert.play()
        elif(SoundFlag2 == 0):
            alert = mixer.Sound('Danger.wav')
            alert.set_volume(0.7)
            alert.play()
    play_count2 = 0;

elif(Warning_FLAG == 0):
    SoundFlag2 = 1
    play_count2 +=1
    if(play_count2>4):
        alert.stop()
    else:
        if(SoundFlag1 == 1):
            alert.stop()
            alert = mixer.Sound('Success.wav')
            alert.set_volume(0.7)
            alert.play()
        elif(SoundFlag1 == 0):
            alert = mixer.Sound('Success.wav')
            alert.set_volume(0.7)
            alert.play()
    play_count1 = 0;
```

This code is designed to operate based on the Warning Flags and Sound Flags received in the mode. At 'Warning Flag' == 2, it is standby state and all LED is on OFF state.

Warning Flag is divided into two main condition. Those two main conditions are determined by Warning Flag. 'Warning Flag' 0 means it is safe, and 'Warning Flag' 1 means it is in danger.

- When 'Warning Flag' is 1, 'Sound Flag1' becomes 1.Play count is increased as each frame.

According to Sound Flag2 when Warning Flag is 1, It stops Mixer sound that was playing and play it.

when the play count reaches 3, the sound stops to prevent regeneration.

- When 'Warning Flag' is 0, 'Sound Flag2' becomes 1. Play count is increased as each frame.

According to 'Sound Flag1' when 'Warning Flag' is 1, It stops Mixer sound that was playing and play it. When the play count reaches 3, the sound stops to prevent regeneration.

Arduino Serial Communication (All mode in Common)

```
# Arduino Connection
if(use_arduino):
    arduino = serial.Serial('com5',9600)
```

This code is placed outside the for loop. It is used to set port number of Arduino in order to correctly connect device.

```
if(use_arduino):
    if(Warning_FLAG == 1):
        pass_val = str(1)
        pass_val = pass_val.encode('utf-8')
        arduino.write(pass_val)
    if(Warning_FLAG == 0):
        pass_val = str(0)
        pass_val = pass_val.encode('utf-8')
        arduino.write(pass_val)
```

This Arduino serial communication code is placed right under the alarm system code. It is placed on the same indent.

```
if(use_arduino):
  pass_val = str(2).encode('utf-8')
  arduino.write(pass_val)
```

This Arduino Serial Communication code is used when nothing is detected from the video and when all algorithm is done. It sends string of 2 to turn off the LED lights on the breadboard

```
// Code Used For Embedding. Not in Python
int LED_PIN1 = 2;
int LED_PIN2 = 3;
int data;

void setup() {
  Serial.begin(9600);
  pinMode(LED_PIN1, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
  digitalWrite(LED_PIN1, LOW);
  digitalWrite(LED_PIN2, LOW);
}

void loop() {
    while (Serial.available()){
    data = Serial.read();
  }

  if (data == '1'){   // Warning Situation.LED_PIN 1 : Red light.
    digitalWrite(LED_PIN1, HIGH);
    digitalWrite(LED_PIN2, LOW);
  }
  else if (data == '0'){ // Safe Situation.LED_PIN 2 : Green light.
    digitalWrite(LED_PIN1, LOW);
    digitalWrite(LED_PIN2, HIGH);
  }
  else if(data =='2'){ // Default Situation. Turn off all LED
    digitalWrite(LED_PIN1, LOW);
    digitalWrite(LED_PIN2, LOW);
  }
}
```

This code is embedded code in Arduino UNO Board.

This receives serial data that has transmitted from python program (Our main Algorithm code) and gives LOW and HIGH to LED PIN 1 and 2.

6. Evaluation of Model On Test Video

For Demo Test of a detection open Visual Studio code and use following command to operate.

```
# Test Video For mode 2 (Construction Entrance Site)
python detect.py --source TestVid.mp4 --mode 1
```

Additional Test Videos

```
# Test Video For mode 1 (Electic Kickboard)
python detect.py --source cap_TestDemo0_2.mp4 --mode 0
```

```
# Test Video For mode 2 (Construction Entrance Site)
python detect.py --source cap_TestDemo1.mp4 --mode 1
```

Test Environments and Setting

- PC : CPU = I7-9750H, RAM = 16GB, GPU = GTX1650
- Image Size : 480×640
- Iou threshold : 0.4
- Confidence Score threshold : 0.6

Mode 1.

We have evaluated model based on test video with metrics of accuracy , precision and recall. As the model we used classify 2 classes, accuracy, precision and recall have been calculated independently

Class	Accuracy(%)	Precision(%)	Recall(%)	FPS
Head	94.72	99.15	95.13	42
Helmet	95.52	88.99	98.32	

As you can see from the table above, this model seems to detect and work well in the test video. FPS of program is fast enough for embedded system application. In this mode of application, recall is more important than precision. It is because in our program, detecting whether person is wearing a helmet or not is more important task than just detecting a person's head.

As you can see, precision of head is really high compared to that of helmet. On the other side recall of helmet is relatively high compared to head. This result implies that it fits our application intention.

Mode 2.

At Mode 2, There is difference with Mode 1 in accuracy of person. In our video, there is a moment when a person's body comes out but face doesn't. Thus, person accuracy was lower than that of other classes.

Class	Accuracy(%)	Precision(%)	Recall(%)	FPS
Head	97.28	95.70	98.41	46
Helmet	89.95	81.5	100	

Mode 3.

There was no further performance evaluation for Mode 3. This is because it was not possible to set the appropriate video and criteria for evaluating Mode 3. If appropriate video is obtained later, performance evaluation will be tested.

7. Discussion and Future Study

Problems found :

- In mode 3, which is test video that is detecting many people at once, it fails to detect helmets correctly when it overlaps.
- There are some frames that it detects as a helmet even though actual object is hat.

- It fails to detect helmet when person is not wearing it. It only detects as helmet when person is wearing a helmet.
- We have failed to find appropriate video for mode 3.

Future Work :

- In order to distinguish helmets from hats and detect helmet without it being on the head, we would like to try training a new model through new datasets.
- Find or take a video that fit mode 3 and evaluate result.