Table of Contents

[Staff Name Tag Object Detection Using Yolov5 Pretrained Model 2](#_Toc172707144)

[Step 1 : Extract Dataset from the video clips given 3](#_Toc172707145)

[Step 2 Label annotation using LabelImg 4](#_Toc172707146)

[Step 3 Create empty annotation for the rest of image that don’t have name tag label 4](#_Toc172707147)

[Step 4 Data Augmentation 5](#_Toc172707148)

[Step 5. Data Partition 7](#_Toc172707149)

[Step 6 Create the Yaml file 9](#_Toc172707150)

[Step 7 Install Yolo Model 9](#_Toc172707151)

[Step 8 Yolo Model Training 10](#_Toc172707152)

[Step 9 : Input Video and detect frame that have staff present 11](#_Toc172707153)

[9.1 Display List 11](#_Toc172707154)

[9.2 Display Image with Scaling Box 14](#_Toc172707155)

[9.3 Save and display in Video 16](#_Toc172707156)

# Staff Name Tag Object Detection Using Yolov5 Pretrained Model

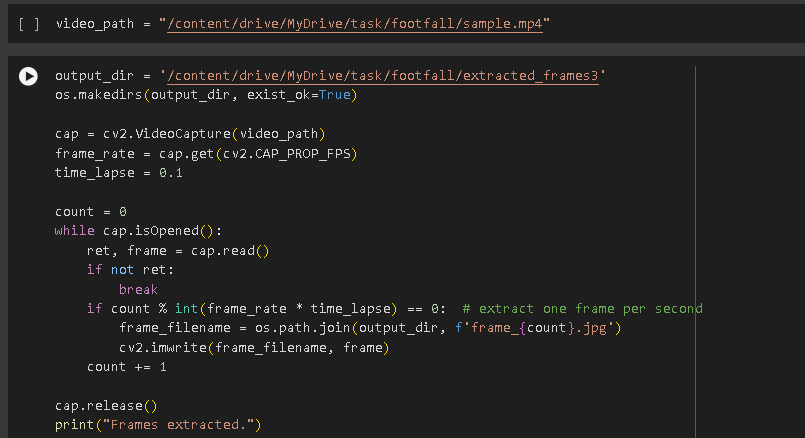
Assumption

* Only staff will wear the name tag so will just detect the name tag to identify which frame in the clips have the staff person.
* Since there is no dataset provided , so will extracted the frame per 0.1 second and use those frames a dataset for the model training .
* Since there are less dataset , so I will use YOLOv5 instead of other version because it is more stable and reliable than other version . Beside , YOLOv5 is more resource efficient and I have only limited resource on my computer .

Tools

* Google Collab
* Python
* Yolov5

# Step 1 : Extract Dataset from the video clips given



I have first set the input video path of the given video clip . Then I have set a directory path folder that will store the extracted frames . Video Capture function is use to initialize the video capture object with the given video file . This code will extract and save frames from the specifies video file at intervals of 0.1 seconds and saving each frame as JPEG In the output directory .

# Step 2 Label annotation using LabelImg

A room with people sitting at desks

Description automatically generated

Due to no dataset provided , I have to label the annotation by myself . I have use LabelImg to label all the extracted image dataset that consists of the name tag . I have name the label as name tag and store the image as YOLO style. The image that have the name tag label will automatically create a txt file that consists of class ID in this case is 0 , center x(x coordinate) , center y (y coordinate), width and heigh.

# Step 3 Create empty annotation for the rest of image that don’t have name tag label

A screen shot of a computer program

Description automatically generated

This code functions as to create empty annotation txt file for all the image that does not know consist of the annotation txt file . It will loop through each image and create an annotation file path by using their own image name . Then , it will check whether that file path exist or not , if no then it will create a text file with empty annotation .

# Step 4 Data Augmentation

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Due to the less amount of dataset , I have decide to perform data augmentation to increase the robustness of the Model . This code has applies data augmentation to the image and also the annotation as well. I have increase the dataset size by adding the horizontal flipping , brightness , rotation and shift scale rotation . So means that there are another 4 image will be created by each original image . Then each augmented image and its corresponding annotation box will be saved with the name then follow by 0,1,2,3,4 .

# Step 5. Data Partition

A screenshot of a computer program

Description automatically generated

After data augmentation , This code splits augmented images and labels into training and validation sets. Directories for training and validation data are created if they don't exist. The code lists all .jpg files in the images directory, shuffles them, and splits them into 80% training and 20% validation sets. It then moves the corresponding images and labels to their respective directories, renaming label files from .jpg to .txt.

A screen shot of a computer program

Description automatically generated

This code combine all the image file and label files into two separate directories which is combine\_train and combine\_test. The combine\_train and combine\_train directories path will be use to train the YOLO model .

# Step 6 Create the Yaml file

A screenshot of a computer program

Description automatically generated

Create a yaml file by insert the training dataset directories and validation dataset directories.

Then nc is the number of class which is 1 in our project and names is the label name which is ‘name tag’.

# Step 7 Install Yolo Model

A computer screen with a black background

Description automatically generated

A black screen with white text

Description automatically generated

Install the YOLO v5 model by using the git command to clone the YOLO model repository from GitHub. Then install all the necessary python package and dependencies that listed in the requirement txt.

# Step 8 Yolo Model Training



This code is use to train the YOLO model . I have set the image size as 640 and use a batch size of 16 . This yolo model is going to train for 50 epochs . I have set the yaml file as the input data . The model will use the train directory path that set in the yaml as training dataset and the Val directory path as testing dataset.

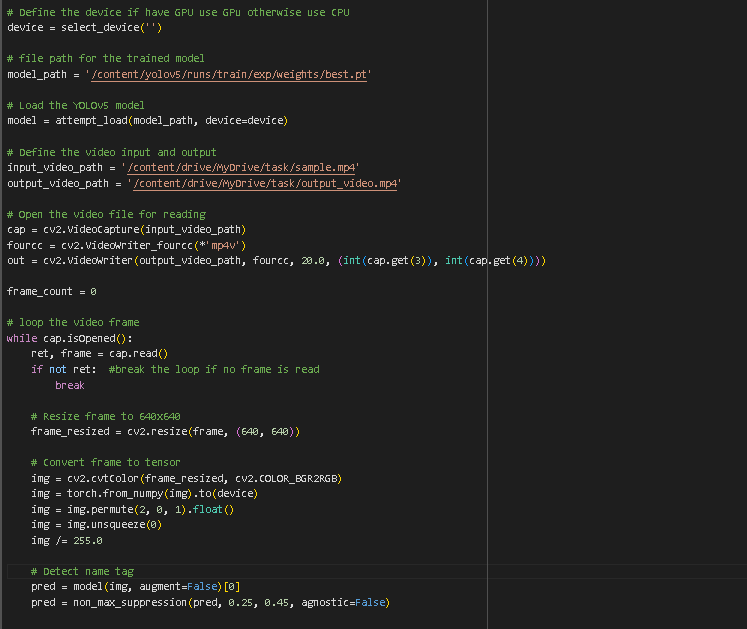
A black screen with white text

Description automatically generated

The yolo model output shows the completion of 50 training epoch with a 0.653 precision which can be improved by training more dataset and other technique but due to the lack amount of dataset and the limited computation unit , I would not able to perform a better model accuracy .

# Step 9 : Input Video and detect frame that have staff present

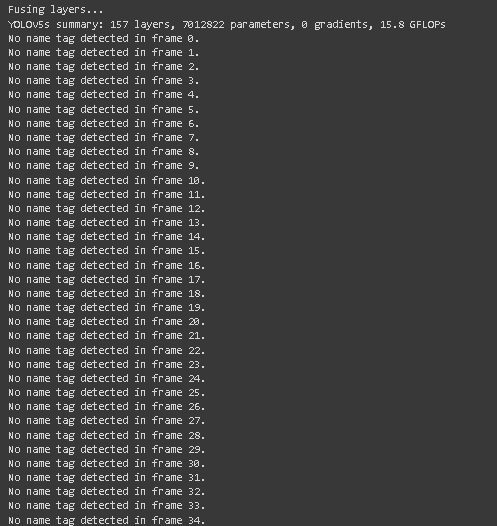
## 9.1 Display List



A computer screen shot of a program code

Description automatically generated

In this code , I have input the video clip given and expected to get a list of result that shows the frame that has detected the name tag . In this code , it will open the given video by using the cv2.video capture function , then it will loop the video frame to detect whether name tag is exist in the frame or not . if name tag exist , a box will scale and show the name tag label on top of the box. Below are the result :

A screenshot of a computer

Description automatically generated

## 9.2 Display Image with Scaling Box

A screenshot of a computer program

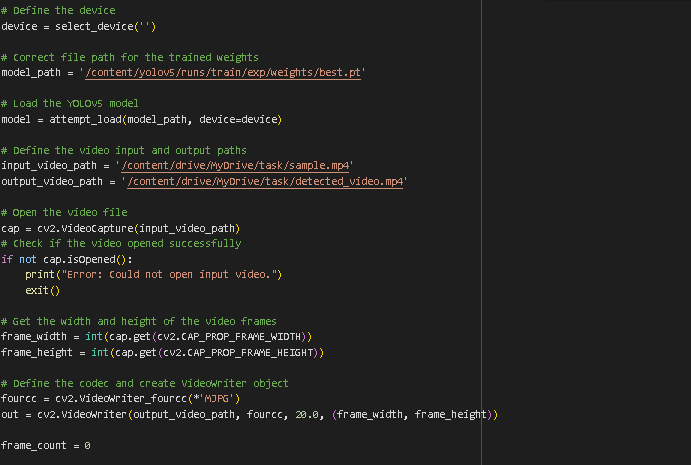
Description automatically generated

This code is similar to the code above but I have replace the input file from video clips to a specific image to detect the name tag and then display in the output using the cv2.iwrite function. Below are the output :

A group of people working in an office

Description automatically generated

## 9.3 Save and display in Video



A screenshot of a computer program

Description automatically generated

This code is also similar to the code above , but the expected output of this code would be save in a video format name detected\_video.mp4.  
The video will be attach in the submission Zip file.