

# Trucks' drivers' detection software

**OVERVIEW , MATERIALS AND METHODS .....2**

- Dataset .....2
- Models .....2
- Process .....2

**RESULTS.....3, 4,5**

## Overview and Methods:

### Dataset:

- ❖ Gathered simple [data](#) of approximate 25 images , and labeled them with [Labellmg](#)

### Model:

- ❖ Given the task of detecting from unclear images like given samples , which are mostly from CCTV cameras , decided to use [YoloV5](#), as it's known for speed , high accuracy
- ❖ The data was small so I needed many epochs so it can learn ,and very high accuracy cause the testing samples are CCTV unclear images
- ❖ Also used YoloV5 as it gave me best results better with the cropping process
- ❖ After prediction, cropped the 'drivers' category , and blurred/anonymized 'passenger' category

### Project Pipeline and Process:

- ❖ Trained the model with these [data](#) , and saves the best weights "**best.pt** " ,to use them later on the detection
- ❖ Changed the **coco128.yaml** file so the model can only read 2 classes: "Driver" and "passenger" classes instead of the whole 80 classes of COCO dataset
- ❖ Changed some function in the **detected.py** file so it crops exactly at the driver's area, subtracted every 20 pixels from 3 corners, the next image shows the changes in **detected.py**

```

        annotator.box_label(xyxy, label, color=colors[c, True])
    if save_crop:
        if names[c]=='driver':
            save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] / f'{p.stem}.jpg')

            x_min= int(xyxy[0].item())-20
            y_min= int(xyxy[1].item())-20
            x_max= int(xyxy[2].item())-50
            y_max= int(xyxy[3].item())

            start_point=(x_min,y_min)
            end_point=(x_max,y_max)

            blurred_img = cv2.GaussianBlur(imc, (21, 21), 0)

            mask = np.zeros((553, 446, 3), dtype=np.uint8)
            mask = cv2.rectangle(mask, start_point, end_point, (255, 255, 255), -1)

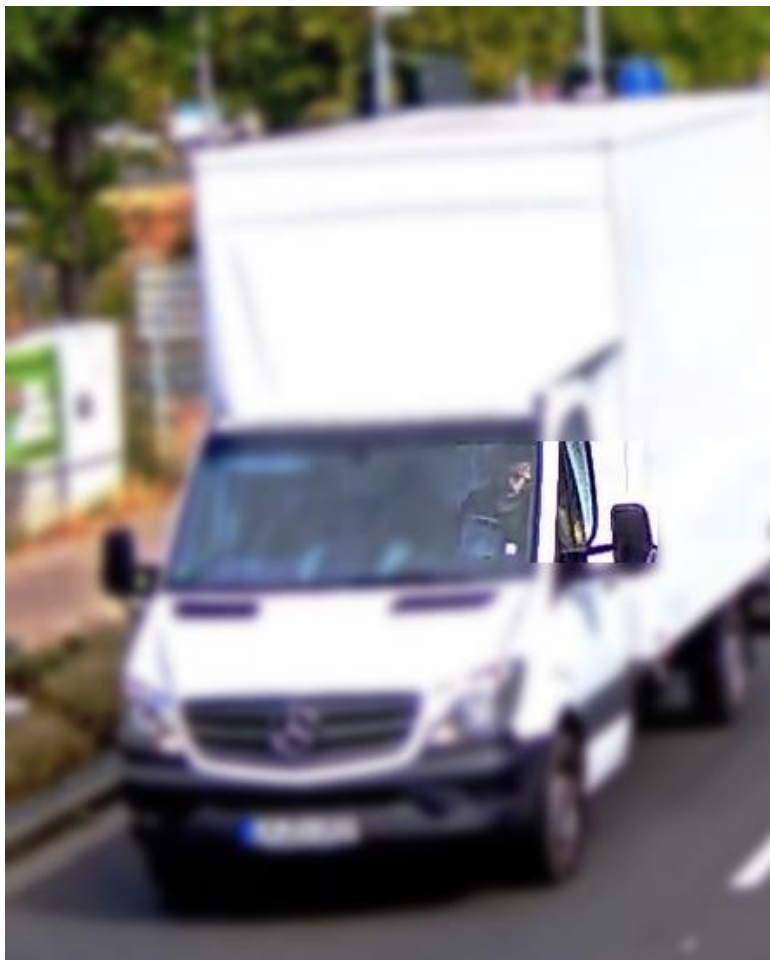
            out = np.where(mask==(255, 255, 255), imc, blurred_img)

            cv2.imwrite("/content/drive/MyDrive/yolo5/results/out1.png", out)

```

## RESULTS

The result for this method is:



But it causes problems if the image sized differed

So tried another method , and this was the results , where I crop only the detected category

Where crops get saved at runs/detected/exp/crops

<Figure size 432x288 with 0 Axes>



**Detection final results:**



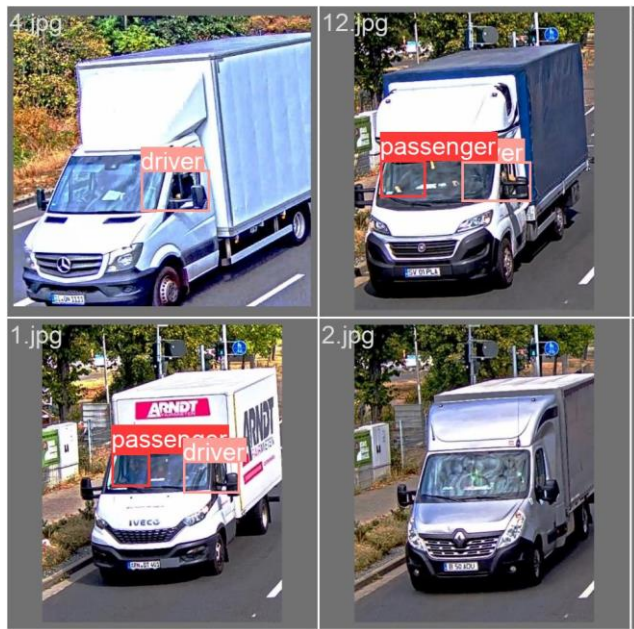




Although I trained the model on “passengers” class, the model **failed** to detect any of them in the CCTV images.

It gives high confidence but only for the “drivers” category

### Part of the labeled data



## The model mAP: 0.942

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
150/159	4.75G	0.0357	0.01763	0.002115	6	640: 100
	Class	Images	Instances	P	R	mAP@.5 mAP@
	all	12	13	0.918	0.5	0.942

**Finally**, I couldn't Docker the software as I was writing it on Google Colab , Docker isn't supported by Google Colab.

I use colab , cause I faced some problems in PyCharm , and also Yolo need NVidia and Cuda .... "My laptop is AMD RadeonGraphics"

So I'm sending the colab notebook with all the dependencies in a public mounted drive on the notebook.

[The notebook](#)

[The repo folder mounted on a driver](#)