REAL-TIME-PPE-DETECTION WITH-YOLO

Mustafa Tayyip Bayram 12237686

Table of Contents

ABSTRACT 2

1. Motivation 2

Methods and Obtained Results 3

2. Data 3

3. Theoretical part 5

Related Studies 5

Why and What is YOLO? 6

Performance Measure 7

General Pipeline 7

Hyperparameters 7

Expected Results 8

4. Implementation 9

5. Evaluation and Conclusion 9

6. Future Steps 11

References 11

List of Figures

FIGURE 1. TOP CAUSES OF CONSTRUCTION RELATED CAUSALTIES 2

FIGURE 2. IMAGE EXAMPLE FIGURE 3. LABEL FILE EXAMPLE 3

FIGURE 4. LABEL INSTANCES WANDB 4

FIGURE 5.MAP 7

FIGURE 6. PIPELINE

CODE 1. RETRAINING SNIPPET 9

CODE 2. DETECTION ON YOUTUBE VIDEO 9

7

FIGURE 7. YOLOV5 VS YOLOV7 8

FIGURE 8. YOLOV5S VS YOLOV5X 9

ABSTRACT

PPE (Personal Prospective Equipment) detection in real time with object detection YOLO is a system that uses deep learning algorithms to recognize items automatically in real time. This technique is especially beneficial in situations where object detection is vital to guaranteeing individual safety, such as building sites or manufacturing plants.

Real-time PPE detection with YOLO takes advantage of this algorithm to detect PPE in real-time. This includes detecting items like hard hats. In this report, just hard hats are trained and others equipment's such as vest, gloves, googles can be extended with required data. The system might use a camera to capture the stream of the workers, and the YOLO algorithm is applied to the stream to detect any PPE that is present or not.

GitHub Repository: https://github.com/mutabay/Real-Time-PPE-Detection-with-YOLO

1. Motivation

The motivation behind personal protective equipment (PPE) detection is to improve workplace safety by ensuring that employees are wearing the necessary equipment to protect them from hazards. The top cause of construction related casualties are falls, people getting stuck in the equipment, collisions. It can be prevented if workers wear appropriate personal prospective equipment's like vest, googles, hard hats etc. The issue with PPE is that it is frequently required for workers to wear it, but it can be uncomfortable, difficult to wear for long periods of time, and easily forgotten or misplaced. The failure to wear the proper PPE can result in serious injuries or even death in some cases.

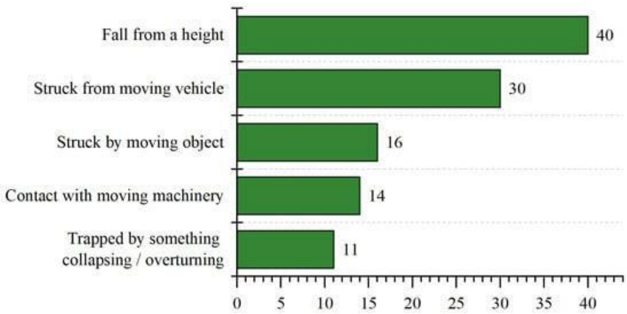


Figure 1. Top causes of construction related causalties

Detecting PPE in real time using object detection is a significant solution to this issue. Employers may track and monitor the use of PPE in real time, ensuring that employees are wearing the proper equipment and boosting workplace safety. The technology is especially important in high-risk situations like construction sites, manufacturing plants, and healthcare facilities. In this project, since there is a problem of finding data especially labelled, construction site hard hat detection has been applied.

Methods and Obtained Results

In general, there are several approaches to resolving this problem. As an example, adding numerous sensors to each PPE and providing control for all of it. If we take a single business area as an example, then thousands of sensors would be required otherwise. As a result, using computer Vision technology will be significantly less expensive, more environmentally friendly, and produce more accurate findings.

This problem was precisely addressed in this paper by adopting the YOLO architecture. These models can then be integrated with CCTV cameras and specific triggers in the following stage of the project to provide results.

Different YOLO models and versions were utilized in order to improve the result and obtain different findings. Standard models yielded more optimized results, as large and complex models require a lot of resources for real time checking. As a result, YOLOv5s gave the best result regarding to quickness to capture objects and accuracy. 0.92 mAP (mean Average Precision) result received with that result. According to the available sources, such as training environment, data, and workload, it is possible to state that it is a decent result.

It is feasible to claim that the most significant and risky aspect of the project is gathering adequate data and cleaning it. The data was collected from open source and cleansed adequately. Detailed information about the data is given below.

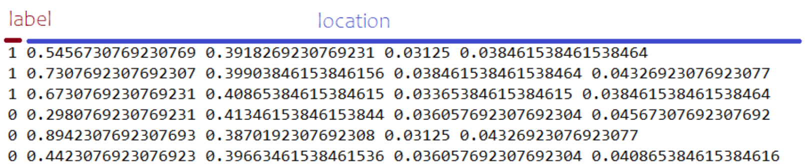
Finally, interpretations and reports were created using the necessary tools. These tools are Wandb, TensorFlow analysis tool.

2. Data

Roboflow Hard Hat Workers Dataset https://public.roboflow.com/object-detection/hard-hat-workers

Data consists of jpg format images and labels that belongs them. Every image has a special txt file that specifies which object it is were with coordinators. The formatting is organized based on the YOLO requirement. There are no missing values, and all images have representative files.

Figure 2. Image example


label location

```
1 0.5456730769230769 0.3918269230769231 0.03125 0.038461538461538464
1 0.7307692307692307 0.39903846153846156 0.038461538461538464 0.04326923076923077
1 0.6730769230769231 0.40865384615384615 0.03365384615384615 0.038461538461538464
0 0.2980769230769231 0.41346153846153844 0.036057692307692304 0.04567307692307692
0 0.8942307692307693 0.3870192307692308 0.03125 0.04326923076923077
0 0.4423076923076923 0.39663461538461536 0.036057692307692304 0.040865384615384616
```

Figure 3. Label file example

The data has some drawbacks. Most important one is that images are not realistic case of use case of this project for real life. The majority of the images are close-ups. If we had data labeled from afar and from those angles, as in CCTV cameras, we could create much more realistic situations and train models on this accurate data. Eventually, CCTV camera detections would be significantly more accurate. Since none of the agents (cameras) will ordinarily detect a worker in front of their face.

Shared By	License	Publication Date	Data Count
Northeastern University – China	Public Domain	September 2022	Totally 7035 Splitted Images

Train	Test	Validation
3688	1581	1766

Labels		
Head (Non-Hard Hat)	Hard Hat	Person (Removed)

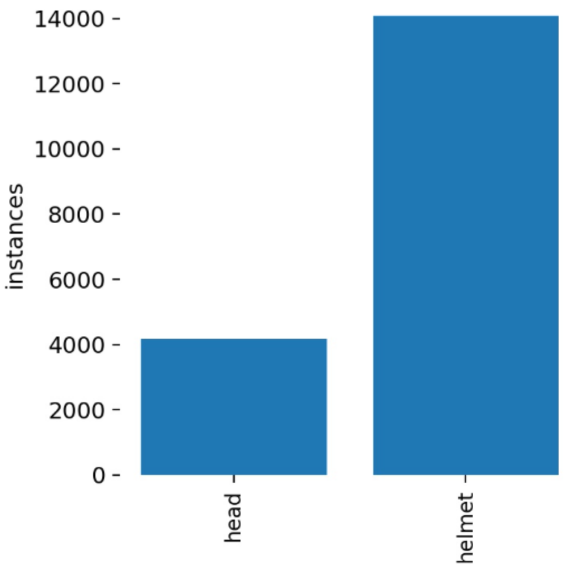


Figure 4. Label Instances wandb

The amount of data is not little, but if it were a little larger, it would contribute more to training. In terms of label balance, the Person label was present in very few images (about 30), therefore cleaning it made more sense because it disrupted the balance significantly. In overall, the data does not appear to be especially balanced. The figure above shows that there are 4000 Head instances and 14000 hard hat instances. To improve quality, it would be appropriate to balance the number of labels.

Data Augmentation is possible with an open-source project for YOLO. It didn't apply in this project but can be used for next versions.

Open-Source Project: https://github.com/srp-31/Data-Augmentation-for-Object-Detection-YOLO-

3. Theoretical part

The learning task of personal prospective equipment real-time detection with computer vision involves teaching a machine learning model to detect and identify specific types of equipment worn or carried by a person in real-time using computer vision techniques.

There are numerous studies on this subject in general. Several algorithms and data sets are discussed, and the results are compared. This research are crucial in the development of the algorithms discussed below.

Related Studies

COVID 19 Real Time PPE Detection | YOLOv4 mAP 79%

Deep Learning for Site Safety: Real Time Detection of Personal Protective Equipment YOLOv3 with different models |YOLOv3 mAP 72.3%

Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches | YOLOv5s mAP 82.65

Object Detection

Object detection is a crucial aspect of computer vision and image processing technology. It involves detecting instances of semantic objects such as humans, cars, buildings, etc. The method for detecting real-time PPE involves capturing streams (images) of workers in the workplace. The images are then processed to an object detection algorithm to detect the presence of PPE.

In general, there are two types of object detection gatherings. Object detection at the image and instance levels. Image-level object detection is concerned with recognizing the presence of an object in an image, but instance-level object detection takes a step further by locating each instance of the object and drawing a bounding box around it. [1]

Real Time Object Detection

There are numerous types of object detection algorithms available. However, this job must be completed in real time, a quick-working algorithm is required. Real-time object detection algorithms are deep learning algorithms that detect and identify items in real time. These algorithms use computer vision techniques to examine images or movies quick-working way. Below are some real-time object detection algorithms that I considered to use:

YOLO (You Only Look Once): YOLO is a popular real-time object detection technique that predicts the presence of items within an image using a single neural network. YOLO is designed to operate in real-time and can recognize many objects within a single image with high efficiency. [2]

SSD (Single Shot Detector): SSD is another prominent real-time object identification approach that predicts the existence of objects within an image using a single neural network. [3]

Faster R-CNN (Region-based Convolutional Neural Network): Faster R-CNN is an object detection algorithm that is designed to detect objects in real-time. Faster R-CNN uses a deep neural network to generate region proposals and classify objects within these regions. [2]

Why and What is YOLO?

YOLO is employed after reading a number of articles regarding these algorithms. For this project, the demand that the algorithm be fast, efficient, and accurate at the fastest possible rate is suitable for YOLO. Besides, YOLO is subdivided into several subtypes and offers convenience in terms of arranging speed, efficacy, and accuracy. Algorithms that provide edge values were chosen for subtype selection. The most discussed versions, V5 and V7, were chosen as versions and S and X models adopted as edge models.

Figure 2. Subtype small

Figure 3. Subtype X-Large

Several studies have been conducted in this area, and a literature review of these studies shows that YOLO-based PPE detection methods have achieved high accuracy and efficiency. There was even 97.6 accuracy.

According to the research, YOLOv7 is better than YOLOv5 in terms of accuracy. The MAP (mean average precision) of YOLOv7 on the COCO dataset is 56.8, and YOLOv5 MAP (mean average precision) on the COCO dataset is 55.0. [4]

YOLOv7 benchmarks show that it's 120x faster than YOLOv5. But the main thing that needs to be understood is these benchmarks have been generated with good GPUs

Performance Measure

The aim of the project is that the solution is fast and accurate. It should be able to capture images in streams and do it correctly. Another important consideration is that the models must be resource efficient as they will be housed inside small processors.

Mean Average Precision (mAP): Gives overall measure of the accuracy.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

$$AP_k = \text{ the AP of class } k$$

$$n = \text{ the number of classes}$$

Figure 5.mAP

Processing Speed: Measure of the time taken by algorithm to process an image.

General Pipeline

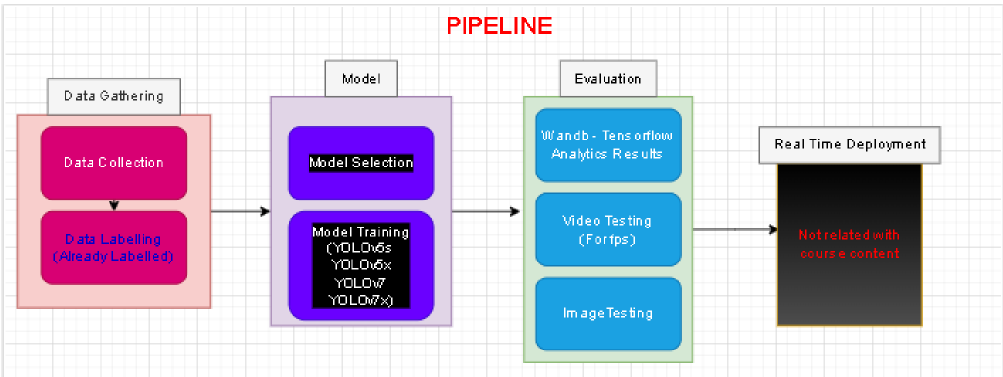


Figure 6. Pipeline

Hyperparameters

The YOLO algorithm contains a number of hyperparameters that can be adjusted to improve its performance. Here are some of the most critical hyperparameters and how they affect the outcome [5] [6]:

Learning Rate: The learning rate controls how rapidly the neural network updates its weights during training. A quicker learning rate can allow the weights to converge more quickly, but it can also cause the network to overshoot and miss the optimal solution [5].

Batch size: The batch size controls how many training examples are handled at once before the network's weights are updated. A higher batch size can improve training speed, but it may also necessitate more memory, which might result in overfitting. [Batch size is 32 in all training samples]

Input image size: The input image size determines the resolution of the images processed by the network. Although a higher input image size can improve detection accuracy, it also necessitates more processing resources. [Image size is 416 in all training samples]

Number of layers: The number of layers in the YOLO network architecture can have an effect on its performance. A deeper network can capture more complicated information, but it also requires more memory and training time. Main difference between X models and S models.

Overall, choosing the right hyperparameters for YOLO can significantly affect its detection accuracy, training time, and memory requirements. It is important to carefully select and tune these hyperparameters based on the specific task and dataset being used.

Expected Results

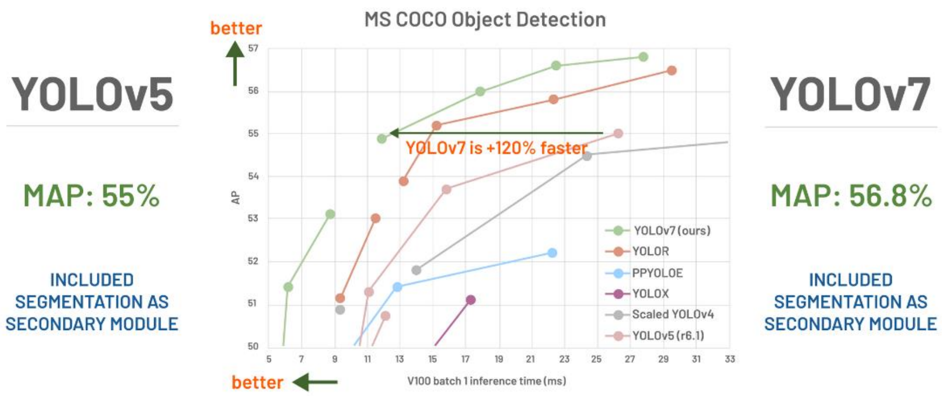


Figure 7. yolov5 vs yolov7

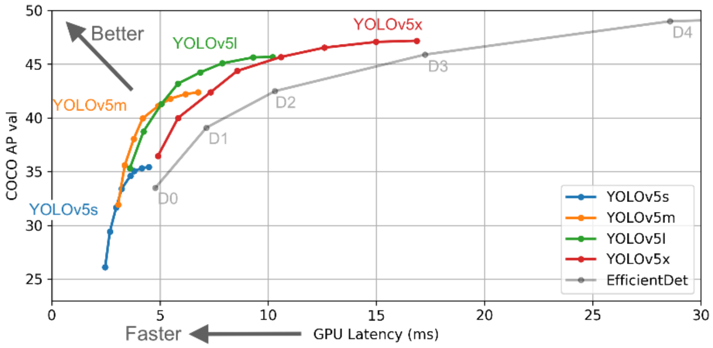


Figure 8. yolov5s vs yolov5x

4. Implementation

Data Collection Platform: Roboflow
Algorithms: YOLOv5s, YOLOv5X, YOLOv7, YOLOv7X
Implementations Platform: Dataspell for easy tasks to work on local. Google Colab for training and resource required steps.
Evaluation Tools: Wandb, TensorFlow Model Analysis
Drawing Tool: Draw.io

```
!python train.py --img 416 --batch 64 --epochs 10 --data /content/data.yaml --weights /content/yolov5/runs/train/yolov5s_results/weights/last.pt --cfg /content/yolov5/models/custom_yolov5s.yaml --cache --name yolo5s_results_retrained
```

Code 1. Retraining snippet

```
!python detect.py --weights /content/yolov5/runs/train/yolo5s_results_retrained/weights/best.pt --img 416 --conf 0.6 --source https://www.youtube.com/watch?v=PB7a1GUuSdc&ab_channel=BoschGlobalSoftwareTechnologies --name yolo5s_test_results_video --save-conf --save-txt
```

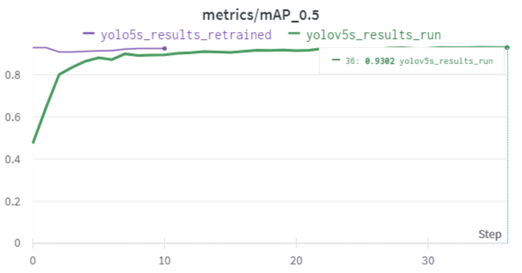
Code 2. Detection on youtube video

5. Evaluation and Conclusion

All the results are available on wandb.

YOLOv5: https://wandb.ai/mutabay/YOLOv5?workspace=user-mutabay

YOLOv7: https://wandb.ai/mutabay/YOLOR?workspace=user-mutabay



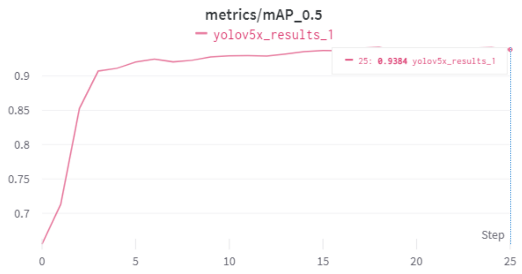
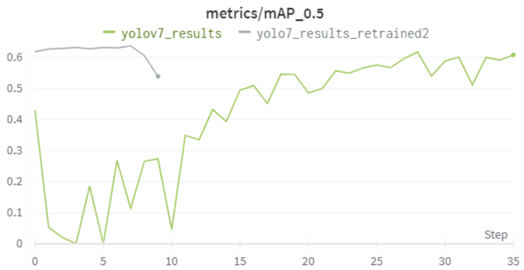


Figure 9. yolov5s mAP

Figure 10. yolov5x mAP



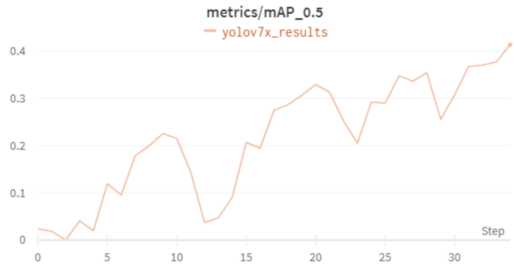


Figure 9. yolov7 mAP

Figure 10. Yolov7x mAP

	YOLOv5s	YOLOv5X	YOLOv7	YOLOv7X
mAP_0.5	0.93 (46 epoch)	0.93 (25 epoch)	0.61 (46 e.)	0.414 (34 e.)
FPS Proficiency	+	-	+	-

The algorithms that are the fastest, most effective (YOLOv5s, YOLOv7) or most accurate (YOLOv5X, YOLOv7X). Surprisingly version 5 discovered more accurate results, contrary to the conclusions of most sources. Due to a lack of resources, X-type algorithms could not be trained to the intended threshold (36 epochs). Even if they have already been trained, it is possible to infer that they are ineffective since they exceed the effectiveness criteria far too much.

Eventually.

- X models are so complex, and they are not fast as much as base models.
- v5 models are more accurate. (Research shows that v7 is much more accurate.)
- v5s model is the fastest (it depends on the system)
- v5s model is the optimal with new starting (fast and accurate)

6. Future Steps

- Data augmentation can be used to diversify data and better recognition.
- More suitable data can be use, more balanced, more realistic regarding to use it in CCTV cameras. (Perspectives and angles)
- Data number should be more for getting better accuracy results.
- Data labels can be increased, for example adding google detection extension will be more useful for general solution.
- Training can be made with better resources, especially required for X models.
- Try new hyperparameters.

References

[1] Z. Hassan, "Instance Segmentation Vs. Object Detection: 3 Things You Need To Know folio3.ai," 2022. [Online]. Available: https://www.folio3.ai/blog/instance-segmentation-vs-object-detection/#:~:text=So%2C%20the%20difference%20between%20instance,instance%20present%20in%20visual%20input..

[2] R. Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms," 2018. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[3] S.-H. Tsang, "Review: SSD — Single Shot Detector (Object Detection)," 2018. [Online]. Available: https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11.

[4] M. Rizwan, "YOLOv7 and YOLOv5 Comparison on Embedded Devices and Computer Systems | cameraanalyze.co," [Online].

[5] M. R. Munawar, "How do Hyperparameters of YOLOv5 Work? medium.com," 2022. [Online]. Available: https://medium.com/augmented-startups/how-hyperparameters-of-yolov5-works-ec4d25f311a2.

[6] ultralytics, "About hyperparameters," 2022. [Online]. Available: https://github.com/ultralytics/yolov5/discussions/7319.