University of Exeter Business School

Assignment Cover Sheet	
Candidate Number	730068016
Module Code	BEMM457
Module Name	Topics in Business Analytics
Assignment Title	Automated Safety Monitoring in Warehouses: An Object Detection Approach for High-Visibility Vest Compliance

Within the Business School we support the responsible and ethical use of GenAI tools, and we seek to develop your ability to use these tools to help you study and learn. An important part of this process is being transparent about how you have used GenAI tools during the preparation of your assignments.

The below declaration is intended to guide transparency in the use of GenAI tools, and to assist you in ensuring appropriate referencing of those tools within your work.

The following GenAI tools have been used in the production of this work: ChatGPT 4-o

☐ *I have used GenAI tools for brainstorming ideas.*

☒ *I have used GenAI tools to assist with research or gathering information.*

☐ *I have used GenAI tools to help me understand key theories and concepts.*

☒ *I have used GenAI tools to identify trends and themes as part of my data analysis.*

☐ *I have used GenAI tools to suggest a plan or structure of my assessment.*

☒ *I have used AI tools to give me feedback on a draft.*

☒ *I have used GenAI tool to generate images, figures or diagrams.*

☒ *I have used AI tools to proofread and correct grammar or spelling errors.*

☒ *I have used AI tools to generate citations or references.*

☒ *Other [please specify] Code writing and troubleshooting……*

☒ *I declare that I have referenced use of GenAI tools and outputs within my assessment in line with the* University referencing guidelines.

MSc Business Analytics

Topics in Business Analytics

Automated Safety Monitoring in Warehouses:

An Object Detection Approach for High-Visibility Vest Compliance

Student ID# 730068016

Word Count: 3004

Table of Contents

Executive Summary 4
1. Introduction 5
1.1. Project focus and rationale 5
1.2. Aim 6
1.3. Objectives 6
1.4. Research questions 6
2. Dataset 6
2.1. Ethics 6
2.2. Reliability 7
2.3. Data structure 7
3. Model Development and Analysis 8
4. Conclusion 13
5. Personal Reflection 14
6. References 15
7. Appendix 17

Executive Summary

Given the number of potential hazards present in a warehouse environment, including the risk of accidents involving forklifts, lorries and other heavy machinery, it is crucial to enforce strict health and safety standards, particularly through Personal Protective Equipment (PPE) compliance. High-visibility (hi-vis) vests play a vital role in reducing accident rates by increasing worker visibility, thereby helping to prevent “struck-by” incidents and collisions. This project investigated the potential of Object Detection Models (ODMs) to automate monitoring of hi-vis vest usage, aiming to enhance health and safety standards in warehouses. The project objectives included examining the ethical implications of the dataset, developing a basic ODM to identify hi-vis vest usage in images, and evaluating its performance and applicability for ensuring safety compliance in warehouses.

The model was trained on a publicly available dataset containing almost 3,900 images of workers in hi-vis vests and without. Despite some steps taken by the author to protect the identities of the individuals on the images, it remains unclear whether proper consent was secured for all images. Using the tools PyTorch and YOLOv5, three training runs were conducted, each with small variations in parameters to determine the optimal configuration.

The third training session produced the best results, but the model’s accuracy remained below 40%, making it unreliable for practical use in warehouses. All sessions plateaued after a point, showing that further training alone wouldn’t improve performance.

To create an improved model, a larger, more diverse dataset of hi-vis vests and augmented images could improve accuracy. However, the time and costs involved present significant limitations, especially if data augmentation is not properly optimised.

1. Introduction

1.1. Project focus and rationale

In warehouses, enforcing Personal Protective Equipment (PPE) requirements—especially high-visibility (hi-vis) vests—is both legally required and critical for worker safety. Research shows that approximately 75% of workplace struck-by fatalities involve heavy equipment (National Safety Council, n.d., para. 5), highlighting the importance of hi-vis clothing in accident prevention. Evidence also suggests that following safety protocols can boost employee morale, lower absenteeism, and increase productivity (Jinnett et al., 2017).

This paper explores how Object Detection Models (ODMs) can be created using a Python code, and analysed in terms of their hi-vis vest detection accuracy to help warehouses take a proactive, data-driven approach to uphold safety standards. A simple ODM model will be developed to better understand the training process, the model will be able to detect whether a person in a photo is wearing a hi-vis vest or not. The accuracy is expected to decrease as we increase the confidence level, for analytical purposes the model accuracy will be measured at the 0.5 confidence level, and the mean across a range of confidence levels from 0.5 to 0.95, which are standard metrics used to evaluate ODMs. The model will be trained using 'You Only Look Once' (YOLO), which is an object detection framework capable of processing images extremely quickly. This framework offers different model sizes — small, medium, and large — each optimised for a trade-off between speed and accuracy, which will be explored during the development phase.

Building this model requires thorough training on image datasets to accurately recognise hi-vis vests. For this project, the ODM will be trained using images from an open-source dataset (Roboflow Universe Projects, 2024). For convenience, the dataset can be accessed at: https://universe.roboflow.com/roboflow-universe-projects/safety-vests. The dataset includes images of people, some wearing hi-vis vests and some not, allowing the model to learn how to distinguish between these two scenarios. This approach guided the development of the project's aim, objectives, and research questions.

1.2. Aim

To develop and evaluate an ODM for hi-vis vest compliance in warehouses, emphasising accuracy and practical applications to enhance worker safety.

1.3. Objectives

- To discuss ethics and reliability concerns surrounding the dataset.
- To build a basic ODM for identifying hi-vis vest usage in images.
- To interpret and summarise the ODM findings, showing how the analysis supports warehouse safety compliance.

1.4. Research questions

- (Q1) What is the model's accuracy at 0.5 confidence level and across a range of confidence levels from 0.5 to 0.95?
- (Q2) What challenges or limitations were faced in model training?

Tools for this project include Pandas for data management, PyTorch for model development, YOLOv5 for object detection, and Matplotlib for visualisation. Additionally, ChatGPT-4 will assist in troubleshooting.

2. Dataset

2.1. Ethics

The dataset contains images of workers, some wearing hi-vis vests and some not, raising privacy concerns about identifying individuals. To address this, the dataset creator used several techniques. Many images are stock photos with subjects who consented to public use, and others are taken from the neck down or at angles that obscure identities. Additionally, the dataset is licensed under CC BY 4.0 (Appendix I), allowing legal copying, redistribution, and adaptation (Creative Commons, n.d.). However, some images lack visible watermarks, and despite the licence, it does not guarantee the content was ethically or legally sourced. While these concerns fall outside the scope of this project, they are acknowledged as limitations.

2.2. Reliability

In ODMs, the standard measures of performance are Mean Average Precision (mAP), Precision and Recall. For simplicity, we will focus on mAP, which measures the overall accuracy of the computer vision model at the 0.5 confidence level (mAP_0.5) and across a range of confidence levels from 0.5 to 0.95 (mAP_0.5:0.95) (Solawetz, 2020). A confidence level indicates the model's certainty in its predictions, and mAP_0.5:0.95 is more important as it evaluates performance across varying levels, providing a more balanced and comprehensive accuracy measure. The reliability of a previous version of the dataset (v5) (https://universe.roboflow.com/roboflow-universe-projects/safety-vests/dataset/5) was evaluated by the creator using Roboflow Train (Nelson, 2020, para. 5), and the results can be found in Figure 1. The dataset that will be used for model training is a newer version of the same dataset (v7) (https://universe.roboflow.com/roboflow-universe-projects/safety-vests/dataset/7). The newer dataset contains 10,000 fewer images. It is therefore expected that the performance of v7 will not reach the same mAP as v5; however, v5 will be retained as a benchmark for comparison due to the lack of a better alternative.

mAP	89.8%
Precision	86.4%
Recall	87.7%

Figure 1: Measures of performance for the ODM dataset (v5). Adapted from https://universe.roboflow.com/roboflow-universe-projects/safety-vests/dataset/5

2.3. Data structure

The dataset is an open-source zip file containing 3,897 images in three folders—"train," "test," and "valid"—along with a data.yaml file, which is used to specify paths to these folders and define class names (hi-vis/no hi-vis). Each of the three folders also contains corresponding labels for the images in a separate "labels" folder, where each label file includes the coordinates for bounding boxes around objects of interest (hi-vis or no hi-vis). The current data structure will be preserved as it is simple, aligns well with the requirements for training object detection models, and is ready to use for building an ODM without further restructuring. Furthermore, the number of images will also remain unchanged, as it is sufficient to avoid overly long training sessions while ensuring adequate detection accuracy. During a test run, approximately 70 corrupted images were identified and deleted, as they prevented the code from working (Appendix II). This is less than 2% of the 3,897 images that were originally in the dataset, which should not be a big concern.

3. Model Development and Analysis

Before starting the model development phase, extensive research was undertaken to understand the initial steps, libraries to be installed, and system requirements. Upon consultation with ChatGPT, the researcher found the need to install PyTorch for model development, a few libraries for debugging, and the YOLO framework (Appendix II). Compared to other, more traditional object detection frameworks, YOLO is capable of processing images at an extremely fast speed, and achieving significantly better results (Redmon, Divvala, Girshick, & Farhadi, 2016, p. 1). An initial test run, performed on a Central Processing Unit (CPU), was interrupted after 1 epoch, (1 full training cycle through all the images in the "training" folder) due to the extreme inefficiency of the CPU. After consulting with ChatGPT (Appendix III), it was decided that 1 epoch was insufficient to generate accurate and consistent results, and more training sessions must be carried out to achieve better results. The training process can be sped up using a Compute Unified Device Architecture (CUDA) integrated version of PyTorch, which replaces the CPU, utilising instead the power of an Nvidia Graphics Processing Unit (GPU), provided the computer has the required hardware (Appendix II). The first major training session was performed using the parameters in Figure 2:

```
##### First Run
# Navigate to the yolov5 directory
%cd yolov5

# Model Training
!python train.py --img 640 --batch 8 --epochs 50 --data "C:/Users/10/OneDrive - University of Exeter/Exeter University/Units/Topics in Business Analytics/dataset/data.yaml" --weights yolov5s.pt --device 0 --name first_run

# where: img 640 – standardising image size across all images (640 being default), epochs 50 – how many times should the model go through the training dataset, data – file path to data.yaml, weights yolov5s – learnt parameters used to identify and recognise objects (using default yolov5s (small) model) device 0 – using GPU

# Result: Success
```

Figure 2: First Training Session Code. Adapted from https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/training_code.py

For the first run with a sensible number of epochs, the small YOLOv5s model was used due to its speed and low memory requirements (Jocher, Munawar, & Qaddoumi, 2023). The run was successful, generating several outputs (see below), including examples of training on batches (Figure 3) and a summary of results (Figure 7). The accuracy plateaued at approximately 0.7 for mAP_0.5 and 0.4 for mAP_0.5:0.95, indicating acceptable accuracy at the 0.5 confidence but weak performance across a range of confidence levels.

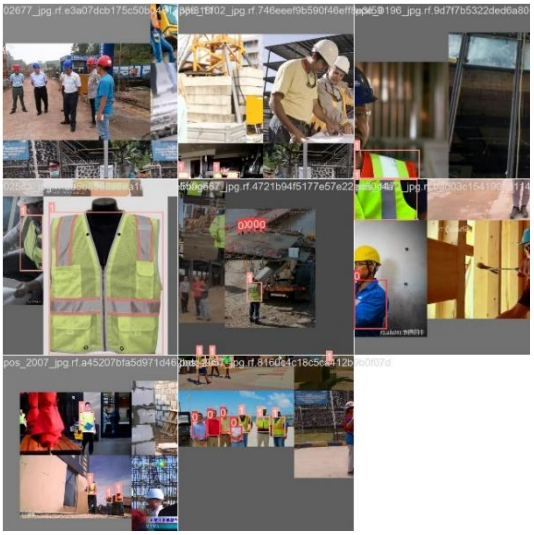


Figure 3: Model training example.

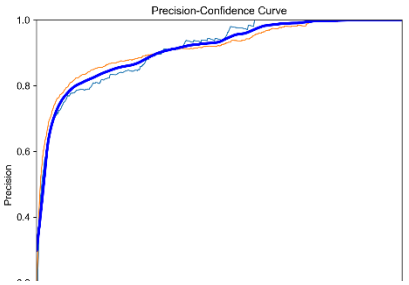


Figure 4: Precision-Confidence Curve.

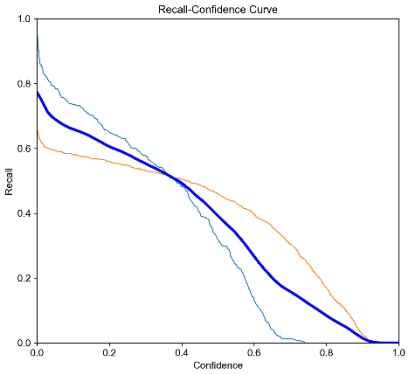


Figure 5: Recall-Confidence Curve.

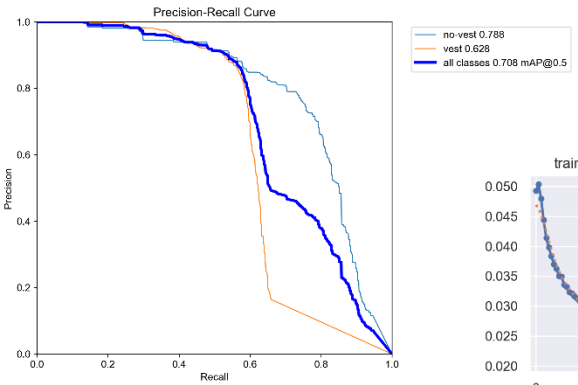


Figure 6: Precision-Recall Curve.

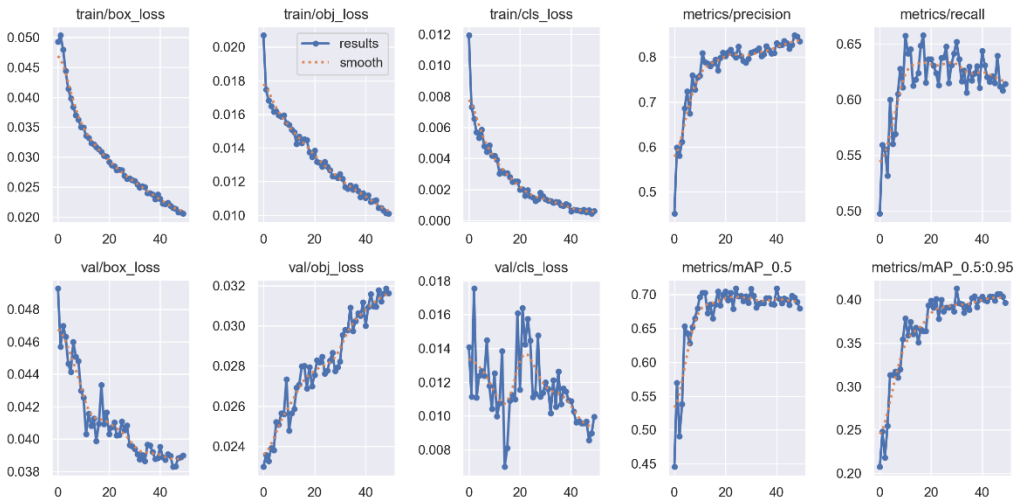


Figure 7: Summary of the results(x axis = epochs).

Eager to try a different approach and achieve better results, the researcher decided to start a new training session using the best.pt file generated during the first run as the new weights (the model's learnt settings that help it recognise and detect objects in images):

```
##### Second Run
%cd yolov5
# 10x more epochs to increase performance, --noplots to reduce the volume of logs sent to the notebook
!python train.py --img 640 --batch 8 --epochs 500 --data "C:/Users/10/OneDrive - University of Exeter/Exeter University/Units/Topics in Business Analytics/dataset/data.yaml" --weights "C:/Users/10/OneDrive - University of Exeter/yolov5/runs/train/first_run/weights/best.pt" --device 0 --name second_run –noplots

# Result: Fail, stopped at 100 epochs due to yolov5's patience (by default, the script automatically stops if performance stops improving for several epochs)
```

Figure 8: Second Training Session Code. Adapted from https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/training_code.py

The number of epochs was increased to 500 to determine the point at which the model's performance would plateau. At 100 epochs, the model reached a plateau and automatically stopped training. Upon closer inspection, it was found that the best mAP at the 0.5 confidence level was 0.71752, and only 0.40601 at 0.5:0.95 (Appendices IV, V & VI), which was rather disappointing given the previous assumption that more epochs would generate better results. Research suggests that this may not always be the case, as extending training can sometimes cause a model to overfit. This means that the model will become very good at identifying hi-vis vests within the training images, but once unseen images are introduced, the accuracy will drop significantly (Machine Learning Models, n.d.).

```
##### Third Run
%cd yolov5
# --weights yolov5l.pt (large model) on 50 epochs and reduced batch size (6) to preserve memory
!python train.py --img 640 --batch 6 --epochs 50 --data "C:/Users/10/OneDrive - University of Exeter/Exeter University/Units/Topics in Business Analytics/dataset/data.yaml" --weights yolov5l.pt --device 0 --name third_run

# Result: Success, time needed = approx. 6h, results not as good as expected, but achieved the highest mAP_0.5:0.95 MAX value of 0.41816, and the highest mAP_0.5:0.95 mean of 0.377881 (see Appendix V)
```

Figure 9: Final Training Session Code. Adapted from https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/training_code.py

The third and final run was conducted using a different set of parameters (Figure 9). Instead of using the best.pt file generated during the second run, with the simpler YOLOv5s model, a more sophisticated YOLOv5l model was used, which produced better results at the cost of lower speed (Jocher, Munawar, & Qaddoumi, 2023).

After completing the final run, the results from all three sessions were collected and compared to identify the most accurate model.

First, two box plots were created using Matplotlib (Figures 10 & 11) to compare the descriptive statistics between mAP _0.5 and mAP_0.5:0.95.

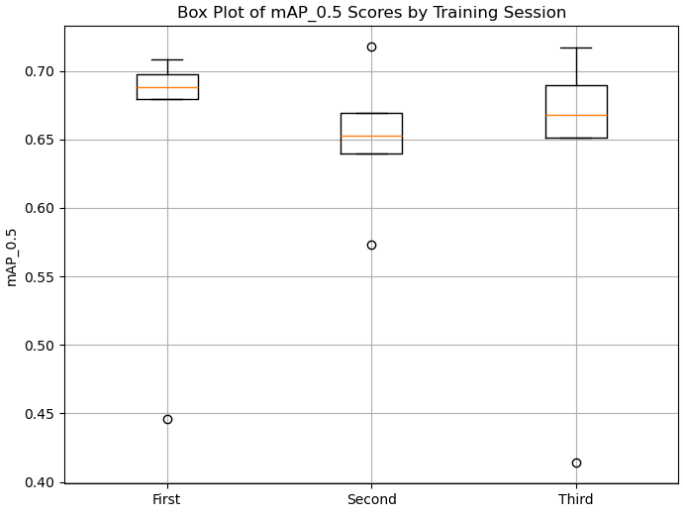


Figure 10: Box Plot for mAP_0.5. Reprinted from https://github.com/automat9/Business-Analytics/tree/master/Semester%201/Topics%20in%20Business%20Analytics/graphs/box%20plot

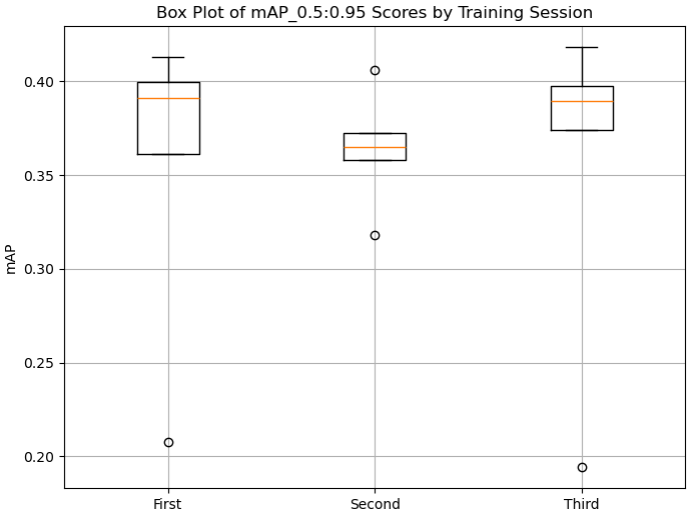


Figure 11: Box Plot for mAP_0.5:0.95. Reprinted from https://github.com/automat9/Business-Analytics/tree/master/Semester%201/Topics%20in%20Business%20Analytics/graphs/box%20plot

Unsurprisingly, if we set the confidence level at 0.5, we get higher mAP across all three sessions; however, the more important statistic mAP_0.5:0.95 shows that no model achieves a mean of even 0.4, indicating low accuracy (Appendix V). One key point is that the second run has a visibly higher minimum value than the other two sessions, because it used the best weights from the first run (best.pt), whereas the other two runs started from scratch with default YOLOv5's weights (YOLOv5s/l). A metadata time series graph highlights this difference (Figure 12). Interestingly, despite its strong start, the second model ended up performing the worst overall

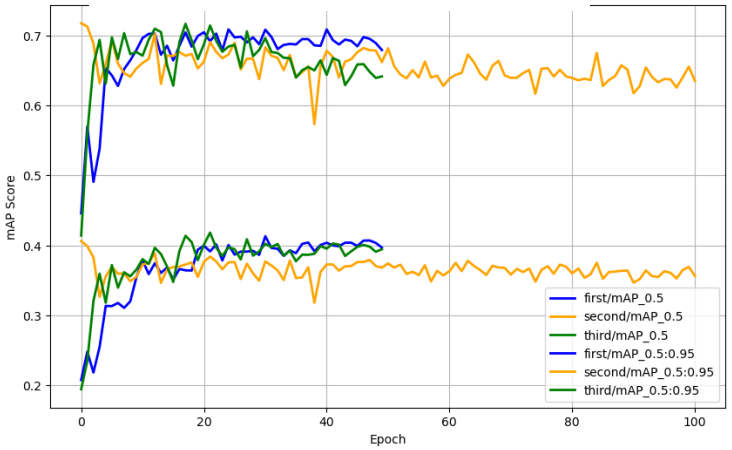


Figure 12: Metadata Time Series. Reprinted from https://github.com/automat9/Business-Analytics/tree/master/Semester%201/Topics%20in%20Business%20Analytics/graphs/meta

(Appendix V). The third model performed slightly better than the others, but all three plateaued at similar levels. This indicates that none of the three models is reliable or accurate enough to effectively support safety compliance in warehouses.

All tables, codes and graphs can be found at: https://github.com/automat9/Business-Analytics/tree/master/Semester%201/Topics%20in%20Business%20Analytics.

To create an improved future model that achieves higher results, it would be beneficial to follow v5's example and apply data augmentation techniques (e.g. sharpening, flipping, or blurring) to the training images (Appendix VII). The right combination can significantly improve accuracy; however, not all combinations yield better performance, as shown in Figure 13. Furthermore, finding the most effective combination can be time-consuming, while automated data augmentation methods are often too costly (Mumuni & Mumuni, 2022, p. 22).

Trials	Used Methods			Sensitivity (%)	Specificity (%)	Accuracy (%)
A	Without Data Augmentation			76,36	70,73	73,55
B	Rotation changing	Mirroring	Zooming	69,77	58,76	64,26
C	Histogram equalization	Color changing	Rotation changing	93,63	95,28	94,46
D	Contrast changing	Brightness changing	Sharping	92,14	94,65	93,40
E	Contrast changing	Sharping	Blurring	99,85	99,82	99,84
F	Sharing	Blurring	Mirroring	81,72	75,80	78,76
G	Zooming	Color changing	Brightness changing	92,52	91,53	92,03
H	Rotation changing	Sharing	Zooming	65,73	76,03	70,88
I	Sharping	Sharing	Color changing	85,17	88,76	86,96
J	Rotation changing	Blurring	Brightness changing	86,52	85,21	85,86
K	Histogram equalization	Zooming	Brightness changing	92,35	78,38	85,36

Figure 13: Performance of different combinations of data augmentation methods. Reprinted from "Comparison of Traditional Transformations for Data Augmentation in Deep Learning of Medical Thermography" by A. H. Ömek, & M. Ceylan, 2019, *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, p. 194. doi: https://doi.org/10.1109/TSP.2019.8769068

Increasing the number of training images may also reduce overfitting and improve accuracy, but this must be weighed against the time and financial costs of obtaining additional data (Adey, 2021). Including hi-vis vests in different colours would improve generalisation, as the dataset currently focuses on green and orange, while some companies also use blue or yellow. The current ODM can also be expanded to identify other objects, such as pallets, forklifts, or even machinery, enabling the company to enforce safety standards while gaining better insights into operational activities. For instance, detecting forklifts and pallets could help monitor equipment utilisation, identify bottlenecks, and improve workflow efficiency. These expansions would require a more diverse dataset, and potentially higher computational resources; however, they could significantly increase the model's practical value for both safety compliance and operational analysis.

4. Conclusion

This project aimed to develop and evaluate an ODM for monitoring hi-vis vest compliance in warehouses. The objectives included examining the ethical and reliability concerns of the dataset, building a basic ODM for identifying hi-vis vest usage in images, and interpreting the findings to assess how the model supports warehouse safety compliance.

It was found that the dataset raised some ethical concerns, most notably the lack of guarantee that all the images were obtained ethically. The analytical tools PyTorch and YOLOv5 allowed for a relatively smooth model training and performance evaluation. In total, three runs were carried out, all using slightly different parameters to identify the best fit. Relevant data were then extracted using suitable code and presented in the form of graphs.

The results show that the third model performed best, with acceptable accuracy at a confidence level of 0.5. However, its average precision across confidence levels from 0.5 to 0.95 was below 0.4, indicating limited reliability. This suggests the model works well in controlled conditions but may not perform as accurately in new environments. All three training sessions plateaued at similar levels of performance after a certain number of epochs, highlighting the need for more data and augmentation techniques to improve accuracy without causing overfitting.

Assumptions, such as more epochs generating more accurate results and relying on pre-trained weights for the first and third runs, influenced the results and highlighted areas for improvement. Overall, this project underscores the promise of ODMs for safety compliance but also emphasises the current limitations of both the dataset and the methods used in this model's development for achieving consistently accurate results.

5. Personal Reflection

Gibbs' Reflective Cycle will be used throughout this section as it provides a clear structure (Figure 14). The project began with identifying interesting datasets, leading to the selection of a dataset for ODM training that captured the researcher's interest. Early doubts arose due to limited machine learning knowledge and uncertainty about the difficulty and time requirements. The dataset's two categories increased confidence, but this optimism was somewhat naïve. Although the number of images in v5 was much smaller than in v7, the dataset still contained almost 3,900 images. A simpler ODM could have been trained on a smaller sample, which would have significantly reduced the time required for model training. On the other hand, choosing a larger dataset allowed the researcher to better understand the time commitments of model training. Consequently, after each run, the researcher was able to refine the timing of training sessions to optimise writing and analysis efficiency. However, looking back, starting with a smaller dataset would have made the initial testing faster and allowed more time to adjust the code based on early results.

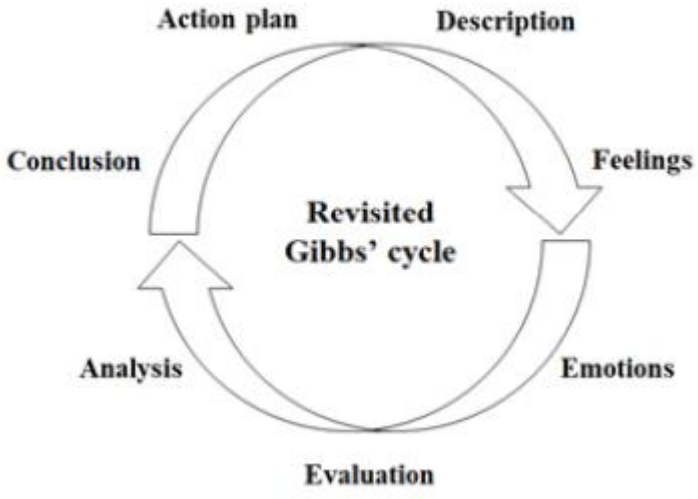


Figure 14: Analysis of Gibbs' Reflective Cycle and the Role of Emotions in Reflective Practice. Reprinted from "Gibbs' Cycle Review - Emotions as a Part of the Cycle," by F. Galli & K. J. New, 2022, *Revista de Educación, Motricidad e Investigación, 19*, 92-101.

A significant source of worry was the difficulty in understanding the metrics used in object detection (OD). For example, it was very frustrating to find any data indicating at what level mAP_0.5:0.95 would be considered good. Extensive consultations with ChatGPT were conducted at this stage, both to understand the concepts and to identify academic papers on these metrics. Because of the lack of relevant data, the researcher had to rely on intuition to determine whether the model's performance was good or poor, which was not a strictly academic approach and caused some frustration. This approach was also less ideal given that it was the researcher's first experience with ODMs, so he had not yet developed realistic expectations for evaluating model performance. However, given the time constraints of the project, it was better to provide at least a rough provisional evaluation than to skip it

altogether. In retrospect, it would have been helpful to conduct a more thorough preliminary review of OD literature to establish better benchmark expectations.

The most frustrating part of the assignment was getting the initial test code to run. The researcher relied heavily on ChatGPT to identify the syntax for model development, as searching for relevant syntax on Stack Overflow or watching lengthy tutorials felt too time-consuming. This reliance increased frustration when ChatGPT's occasional inaccuracies led to errors and bugs that were time-consuming and challenging to debug. At times, the researcher felt that the most critical part of the assignment was slipping out of his control. However, through more targeted research, he gradually gained a better understanding of the syntax provided by ChatGPT and was able to perform future runs with custom parameters that worked as intended. To balance this reliance, the researcher created box plot graphs and other visualisations, which gave a sense of accomplishment and confidence in coding. For future assignments, he plans to minimise reliance on AI, using it only for essential troubleshooting, and to start projects earlier to allow more time for research, learning, and refining code.

6. References

1. Adey, B. (2021, April 27). *Investigating ML Model Accuracy as Training Size Increases.* Telstra Purple. https://purple.telstra.com/blog/investigating-ml-model-accuracy-as-training-size-increases

2. Creative Commons. (n.d.). *Attribution 4.0 International (CC BY 4.0)*. Creative Commons. Retrieved November 6, 2024, from https://creativecommons.org/licenses/by/4.0/

3. Galli, F., & New, K. J. (2022). *Gibbs' cycle review - Emotions as a part of the cycle*. *Revista de Educación, Motricidad e Investigación, 19*, 92-101. https://doi.org/10.33776/remo.vi19.7224

4. Jinnett, K., Schwatka, N., Tenney, L., Brockbank, C. V. S., & Newman, L. S. (2017). Chronic conditions, workplace safety, and job demands contribute to absenteeism and job performance. *Health Affairs, 36*(2), 237-244. https://doi.org/10.1377/hlthaff.2016.1151

5. Jocher, G., Munawar, R., & Qaddoumi, B. (2023). *YOLOv5 tips for best training results*. Ultralytics. Retrieved November 5 2024, from

https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/#training-settings

6. Machine Learning Models. (n.d.). *Optimizing machine learning: Determining the ideal number of epochs*. Retrieved November 12, 2024, from https://machinelearningmodels.org/optimizing-machine-learning-determining-the-ideal-number-of-epochs/

7. Mumuni, A., & Mumuni, F. (2022). *Data augmentation: A comprehensive survey of modern approaches*. Array, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

8. National Safety Council. (n.d.). *Struck by objects*. National Safety Council. Retrieved November 5, 2024, from https://www.nsc.org/workplace/safety-topics/struck-by-objects

9. Nelson, J. (2020, November 3). *Evaluating Object Detection Models with mAP by Class*. Roboflow. https://blog.roboflow.com/mean-average-precision-per-class/

10. OpenAI. (2024). ChatGPT (November 30 version) [Large language model]. https://chat.openai.com/auth/login

11. Örnek, A. H., & Ceylan, M. (2019). *Comparison of traditional transformations for data augmentation in deep learning of medical thermography*. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (pp. 191-194). IEEE. https://doi.org/10.1109/TSP.2019.8769068

12. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, real-time object detection* [Conference paper]. arXiv. https://arxiv.org/abs/1506.02640

13. Roboflow Universe Projects. (2024). Safety Vests Computer Vision Project. [Dataset]. Roboflow Universe.

14. Solawetz, J. (2020, May 6). *What is Mean Average Precision (mAP) in Object Detection?* Roboflow. https://blog.roboflow.com/mean-average-precision/

7. Appendix

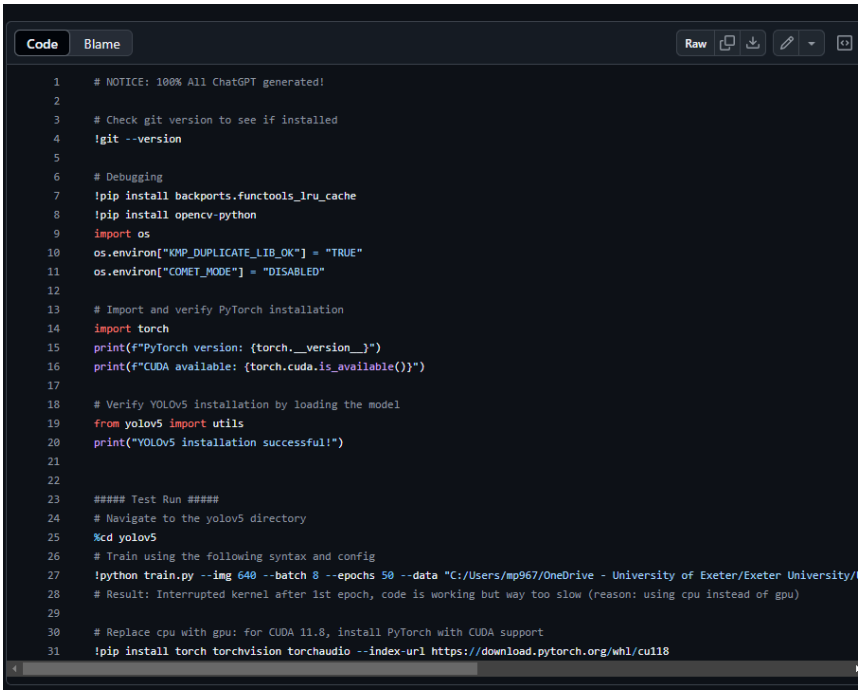
Cite This Project

LICENSE CC BY 4.0

If you use this dataset in a research paper, please cite it using the following BibTeX:

```
@misc{
safety-vests_dataset,
title = { Safety Vests Dataset },
type = { Open Source Dataset },
author = { Roboflow Universe Projects },
howpublished = { \url{ https://universe.roboflow.com/roboflow-universe-projects/safety-vests } },
url = { https://universe.roboflow.com/roboflow-universe-projects/safety-vests },
journal = { Roboflow Universe },
publisher = { Roboflow },
year = { 2024 },
month = { jul },
note = { visited on 2024-11-06 },
}
```

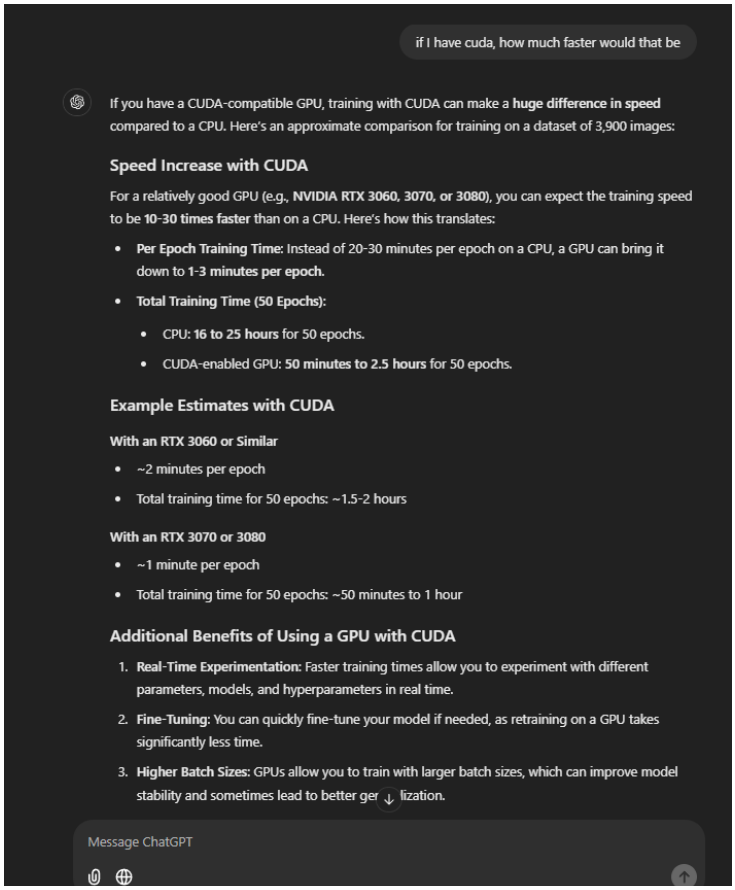
***Appendix I**:* CC BY 4.0 Licence possessed by the dataset used for this project. Adapted from https://universe.roboflow.com/roboflow-universe-projects/safety-vests



Code Blame Raw

```
# NOTICE: 100% All ChatGPT generated!

# Check git version to see if installed
!git --version

# Debugging
!pip install backports.functools_lru_cache
!pip install opencv-python
import os
os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"
os.environ["COMET_MODE"] = "DISABLED"

# Import and verify PyTorch installation
import torch
print(f"PyTorch version: {torch.__version__}")
print(f"CUDA available: {torch.cuda.is_available()}")

# Verify YOLOv5 installation by loading the model
from yolov5 import utils
print("YOLOv5 installation successful!")


##### Test Run #####
# Navigate to the yolov5 directory
%cd yolov5
# Train using the following syntax and config
!python train.py --img 640 --batch 8 --epochs 50 --data "C:/Users/mp967/OneDrive - University of Exeter/Exeter University/
# Result: Interrupted kernel after 1st epoch, code is working but way too slow (reason: using cpu instead of gpu)

# Replace cpu with gpu: for CUDA 11.8, install PyTorch with CUDA support
!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

***Appendix II**:* ChatGPT code used to install relevant libraries, prevent bugs, and perform a test run (OpenAI, 2024). Reprinted from https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/prerequisites%26test.py



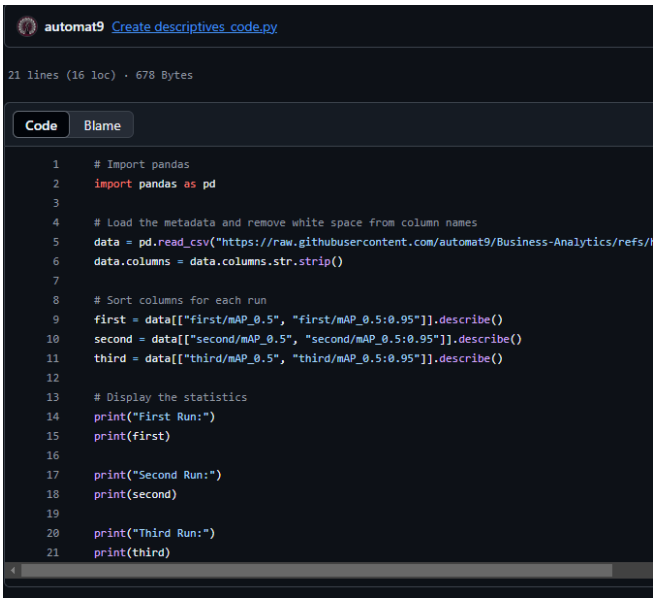
***Appendix III**:* ChatGPT listing the benefits of using CUDA as opposed to the CPU approach (OpenAI, 2024).

	Descriptives	First/mAP_0.5	Second/mAP_0.5	Third/mAP_0.5
2	Count	50.0	101.0	50.0
3	Mean	0.672929	0.654837	0.663819
4	Std	0.052689	0.021346	0.045529
5	Min	0.44569	0.5733	0.414
6	25%	0.679142	0.63987	0.651232
7	50%	0.68774	0.65282	0.66787
8	75%	0.69731	0.66925	0.68989
9	Max	0.70865	0.71752	0.71664

***Appendix IV**:* Descriptive statistics for each run (mAP at 0.5). Reprinted from: https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/results/descriptives/descriptive_statistics_mAP_0.5.csv

1	Descriptives	First/mAP_0.5:0.95	Second/mAP_0.5:0.95	Third/mAP_0.5:0.95
2	Count	50	101	50
3	Mean	0.368914	0.364987	0.377881
4	Std	0.049069	0.012086	0.039858
5	Min	0.20744	0.31796	0.19421
6	25%	0.361355	0.3581	0.374265
7	50%	0.390985	0.36494	0.38932
8	75%	0.39972	0.37245	0.39776
9	max	0.41304	0.40601	0.41816

***Appendix V**: D*escriptive statistics for each run (mAP at 0.5:0.95). Reprinted from: https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/results/descriptives/descriptive_statistics_mAP_0.5-0.95.csv



***Appendix VI**:* Code used to extract the descriptives in appendices IV and V. Reprinted from: https://github.com/automat9/Business-Analytics/blob/master/Semester%201/Topics%20in%20Business%20Analytics/runs/results/descriptives/descriptives_code.py

***Appendix VII**:* Data augmentation steps performed on the v5 model of the hi-vis dataset.
Reprinted from: https://universe.roboflow.com/roboflow-universe-projects/safety-vests/dataset/5