INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

ROBERT GORDON UNIVERSITY ABERDEEN

CM4610 – Machine Vision

Module Co-ordinator – Nihal Kodikara

Assignment Type: Individual Coursework

“Safety Equipment Detection for Construction Sites”

Student Name: M.Hasinindu Jayashan De Silva

IIT ID - 20211295

RGU ID - 2312548

Table of Contents

List of figures

GitHub - https://github.com/HJayashan/PPE-Detection-for-Construction-safety

1. Introduction

Construction sites remain among the most hazardous places to work. In the United Kingdom, 45 construction workers died in workplace accidents during 2022 to 2023 the highest number of fatal injuries recorded for any industry (Health and Safety Executive, 2023). Many of these deaths were preventable if workers had worn proper safety equipment.

Personal Protective Equipment (PPE), especially safety helmets and high-visibility vests, is central to reduce risks on site. Helmets reduce the severity of head injuries caused by falling objects, while high-visibility vests make sure workers can be seen by vehicle operators and machinery drivers in the busy environments (Teizer et al., 2010). Although safety regulations requiring workers to wear this equipment, making sure everyone wears it remains difficult.

Currently, most construction sites rely on safety supervisors to check PPE usage manually. This approach has several weaknesses. First, supervisors cannot watch every worker all the time, especially on large or complex sites. Secondly, the level of inconsistency of manual inspections is due to supervisor fatigue and distractors (Park and Brilakis, 2012). Third, these inspections are done after the workers have been placed in the dangerous areas and not at the point of entry to the site. Lastly, automated tracking of the personnel using safety gear and the timestamps is not present (Fang et al., 2018)

The recent advances in artificial intelligence and computer vision offer a potential solution. Modern object detection models, including those that use the YOLO (You Only Look Once) architecture, are able to detect objects in an image at a very fast and precise rate (Redmon et al., 2016). Such systems can monitor compliance with safety equipment continuously, thus providing consistency of verification and automatic record-keeping even without the presence of a human factor that is constantly supervising the equipment.

The current paper is based on the formation of an automated personal protective equipment (PPE) detection system that would be applicable in a construction setting. The system uses YOLOv8 to determine the presence of safety helmets and high-visibility coats in real time (Jocher et al., 2023). More importantly, the system can be used to show the relevance of this technology at the construction site entry points and allow access based on the relevant possession of PPE. The people who are with the necessary PPE would be given site access, and those who are not would be refused entry. Such a transition implies the transition of problem detection to active prevention.

1.1 Project Objectives

This project addresses the question whether computer vision technology can detect PPE accurately and quickly enough for real-time access control at construction sites.

1. Achieve a minimum detection accuracy of 85% measured using mean Average Precision for identifying helmets and high-visibility vests in construction site images.
2. Process images faster so workers are not delayed at entry gates.
3. Build a working demonstration that shows how detected PPE could control gate access.
4. Evaluate the system performance using standard metrics so results can be compared with existing research.
5. Analyze practical issues like computational requirements, environmental conditions, and ethical issues to assess real world feasibility.

1.2 Scope and Limitations

The given project will focus on identifying two types of PPE: safety helmets and high-visibility vests as these are mandatory at all construction sites and address the most common reasons of deadly injuries and deaths. However, the training materials cover also other safety equipment like gloves, boots, and masks, the focus is on helmets and vests since they are the minimum safety requirements.

The system is designed to be used in controlled situations where the workers will go to a camera-controlled gate where they have an unhindered view of the gate. Coverage Performance can be reduced when operating under intense light e.g. bright sun or heavy shadows or in low light or when the equipment is heavily covered. Such limitations are inherent to camera-based solutions, and they would require additional solutions like increased lighting or the use of several cameras in the work environment.

The implementation is currently dealing with still images and not video continuously. Although this shows the feasibility of the concept, a production system would be enhanced with video-based processing which follows people through successive frames. This would require a greater degree of reliability which is out of scope of the current project.

The issue of privacy is solved by concentrating solely on the identification of equipment and avoiding the identification of workers. Facial recognition or tracking personal identity is not implemented in the system. However, a real implementation would need clear policies regarding data management, data storage and access to protect employee privacy.

1.3 Report Structure

The rest of this report is structured in the following way. Chapter 2 is a review of previous studies concerning object detection and PPE detection system. Chapter 3 outlines the methodology, which is dataset description, model selection, and training strategy. The system implementation is described in chapter 4. Chapter 5 shows the findings and performance analysis. Chapter 6 talks about implications of real-life application. The conclusion of chapter 7 gives the overall summary of the achievements and future work recommendations.

2. Background and Related Work

This chapter reviews the development of object detection technology and examines previous research on PPE detection systems.

2.1 Evolution of Object Detection

2.1.1 Traditional Computer Vision Methods

Before deep learning, object detection relied on hand-crafted features and traditional machine learning. One of the earliest successful systems was the Viola-Jones face detector (Viola and Jones, 2001), which could detect faces in real-time using simple rectangular features. While this was groundbreaking at the time, it only worked well for faces and struggled with other types of objects.

Later methods used more sophisticated features. The Histogram of Oriented Gradients (HOG) approach (Dalal and Triggs, 2005) became popular for detecting people and vehicles by analyzing edge patterns in images. HOG worked better than earlier methods but required careful manual design of features. Researchers had to decide exactly what patterns to look for, which was time-consuming and often did not work well when conditions changed.

These traditional methods had significant limitations for PPE detection. Construction sites have complex backgrounds, varying lighting conditions, and equipment that appears in many different colors and styles. Hand-crafted features struggled to handle this variety, making them unreliable for real-world construction safety applications (Park and Brilakis, 2012).

2.1.2 Deep Learning Revolution

Everything changed in 2012 when a deep learning system called AlexNet won the ImageNet competition (Krizhevsky et al., 2012). AlexNet used Convolutional Neural Networks (CNNs) to automatically learn features from images rather than relying on hand-crafted designs. This approach achieved much better accuracy than previous methods and sparked massive interest in deep learning for computer vision.

The success of AlexNet led to rapid development of object detection systems. R-CNN (Girshick et al., 2014) was one of the first to apply deep learning to object detection. It worked by first proposing potential object locations in an image, then using a CNN to classify what was in each location. However, R-CNN was very slow because it had to run the CNN separately for each proposed location, sometimes thousands of times per image.

Faster R-CNN (Ren et al., 2015) improved this by sharing computations across all proposals, making detection much quicker. While Faster R-CNN achieved good accuracy, it still took around 100 milliseconds to process an image, which is too slow for some real-time applications like gate access control where workers should not have to wait.

2.1.3 YOLO: Real-time Object Detection

The YOLO (You Only Look Once) system introduced a completely different approach (Redmon et al., 2016). Instead of proposing object locations and then classifying them separately, YOLO looks at the whole image once and predicts all objects simultaneously. This makes YOLO much faster than R-CNN-based methods, processing images in around 20-30 milliseconds while maintaining good accuracy.

YOLO has gone through several versions, each improving on the previous one. YOLOv3 (Redmon and Farhadi, 2018) added the ability to detect objects at multiple scales, making it better at finding both small and large objects. YOLOv4 (Bochkovskiy et al., 2020) introduced new training techniques that improved accuracy without sacrificing speed.

YOLOv8, released in 2023 (Jocher et al., 2023), represents the latest evolution. It includes several improvements: an anchor-free detection system that simplifies training, better feature extraction that improves accuracy, and optimized code that runs faster. These improvements make YOLOv8 particularly suitable for applications like PPE detection where both speed and accuracy are important.

2.2 PPE Detection Research

2.2.1 Early PPE Detection Studies

Early research on automated PPE detection used traditional computer vision methods. These systems typically looked for specific colors (like yellow or orange hard hats) and simple shapes (Park and Brilakis, 2012). However, they performed poorly in real construction sites because of varying lighting, dirt on equipment, and similar colors in the background.

Some researchers tried using machine learning with hand-crafted features. Kim et al. (2016) used HOG features with Support Vector Machines to detect hard hats, achieving reasonable accuracy in controlled conditions. However, these methods still struggled with the complexity and variety found in real construction environments.

2.2.2 Deep Learning for PPE Detection

The application of deep learning dramatically improved PPE detection performance. Fang et al. (2018) developed a system using Faster R-CNN to detect hard hats in construction site videos, achieving 94.5% accuracy. This was a significant improvement over traditional methods and demonstrated that deep learning could handle the complexity of real construction environments.

Wu et al. (2019) compared different deep learning architectures for hard hat detection, testing Faster R-CNN, SSD (Single Shot Detector), and YOLOv3. They found that YOLOv3 provided the best balance of accuracy (90.2% mean Average Precision) and speed (23 milliseconds per image). This work highlighted that YOLO-based systems were particularly well-suited for construction safety applications requiring real-time processing.

More recent work has expanded beyond just hard hats. Nath et al. (2020) developed a system that could detect multiple types of PPE including helmets, vests, gloves, and masks. Their system achieved good accuracy but was designed for offline analysis of recorded video rather than real-time access control. This represents an important gap that the current project addresses.

2.2.3 Datasets for PPE Detection

A major challenge in PPE detection research has been the lack of large, high-quality datasets. Early studies used small datasets of a few hundred images, which limited the performance of deep learning models. The COCO dataset (Lin et al., 2014), while containing 200,000 images, includes very few construction site scenes with PPE.

Recently, platforms like Roboflow have made it easier to share and access specialized datasets. The PPE Detection dataset used in this project contains over 2,000 images specifically focused on construction safety equipment, providing a much better foundation for training detection models than general-purpose datasets.

2.3 Transfer Learning

Transfer learning is a technique where a model trained on one task is adapted for a different but related task (Pan and Yang, 2010). For object detection, this typically means starting with a model pre-trained on a large dataset like COCO, then fine-tuning it for specific objects like PPE. This approach works because the early layers of neural networks learn general features like edges and textures that are useful across many different tasks.

Transfer learning offers several advantages for PPE detection. First, it requires less training data because the model already understands general visual concepts. Research has shown that transfer learning can achieve good performance with datasets of just 1,000-2,000 images, whereas training from scratch might need 10,000 or more (Yosinski et al., 2014). Second, it trains faster because only the task-specific layers need significant updating. Third, it often achieves better accuracy because the pre-trained features provide a better starting point than random initialization.

For this project, using a COCO pre-trained YOLOv8 model provides an excellent foundation. COCO includes many objects with similar visual characteristics to PPE (people, clothing items, protective gear), making the transfer to construction safety equipment particularly effective.

2.4 Performance Metrics

To assess this project, it is absolutely necessary to understand how to measure object detection performance. The most typical measures are the precision, the recall and the mean Average Precision (mAP).

The percentage of correct detections is referred to as precision. As an example, assume that the system results in 100 helmets and 95 of those are really helmets and the rest are false alarms, then the precision will be 95. A high precision thus means that there is a low level of false positives.

Recall refers to the percentage of the real objects which are picked up. To take an example, when there are 100 helmets in an image and the system identifies 90 of them correctly, the recall is 90. High recall means that there are minimal objects to be missed as the target.

Mean Average Precision (mAP) provides a unified measure of precision and recall in a single scalar measure and is the most common measure used in object detection studies. A detection is considered correct when the bounding box of the object covers the ground-truth box by a minimum of 50 /, which is calculated as Intersection over Union (IoU). mAP 05 The mAP with this 50% IOU threshold is computed. It has been demonstrated by empirical research (Nath et al., 2020) that an mAP at 0.5 over 85 percent indicates a production-ready performance in terms of safety use..

3. Methodology

The chapter outlines the research process used to come up with the personal protective equipment (PPE) detection system. It includes the dataset selection, the model architecture selection, the training process configuration, and the evaluation protocols adoption. The design of the methodology was geared towards producing a system with high accuracy, as well as the adequate speed of computation to be used in real time deployment.

3.1 High-Level Flow Diagram

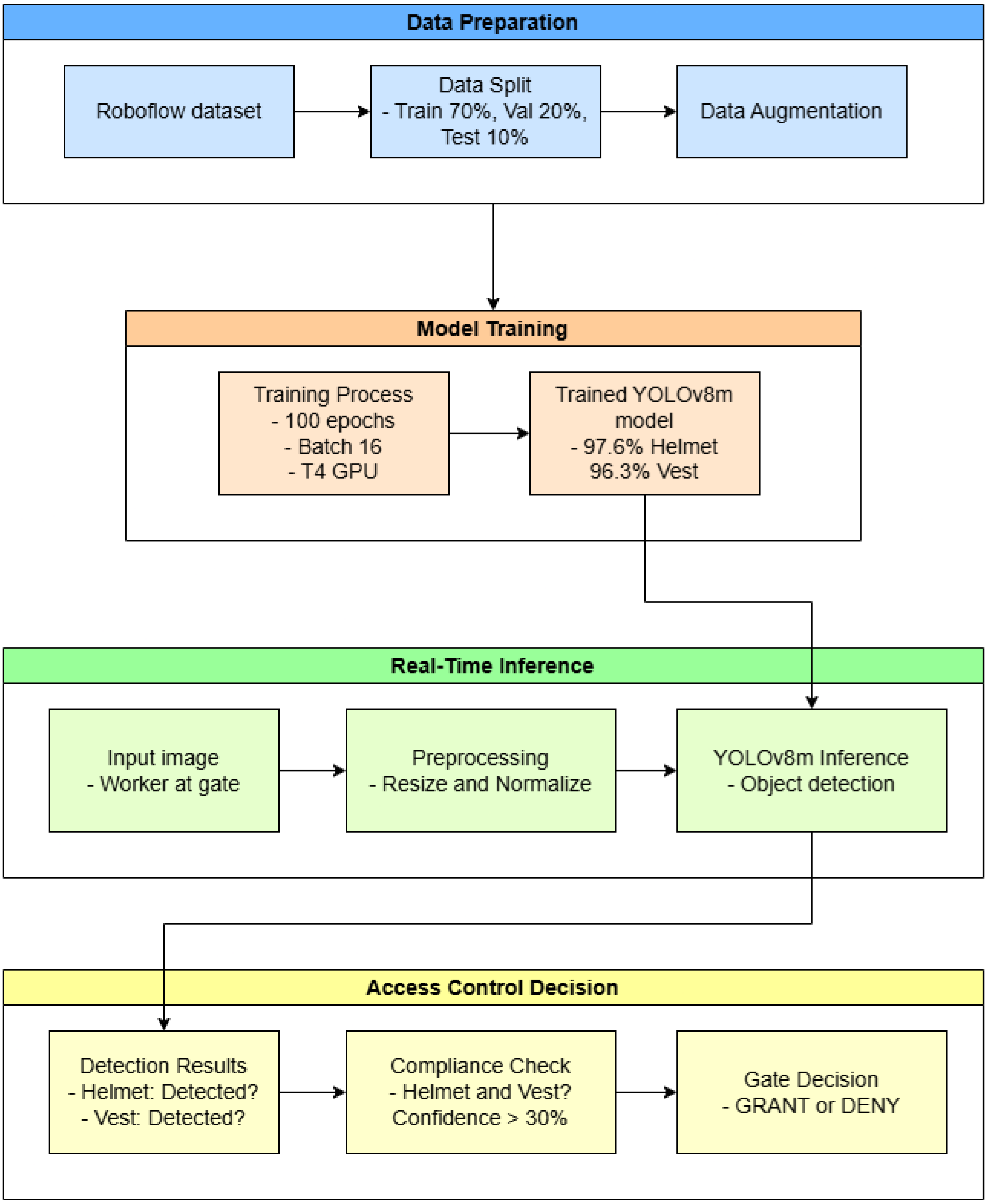


Figure 3-1: High Level Flow Diagram

3.2 Dataset

3.2.1 Dataset Selection

The PPE Detection data set was taken on Roboflow Universe, which is a computer-vision dataset sharing platform (Roboflow, 2023). here were three main reasons as to why the dataset was selected. First, it does not focus on generic categories of objects but construction safety equipment. Second, it has 2,114 images, which are enough to train a deep-learning model. Third, the annotations are of high quality and in YOLO format, which allows them to use with YOLOv8.

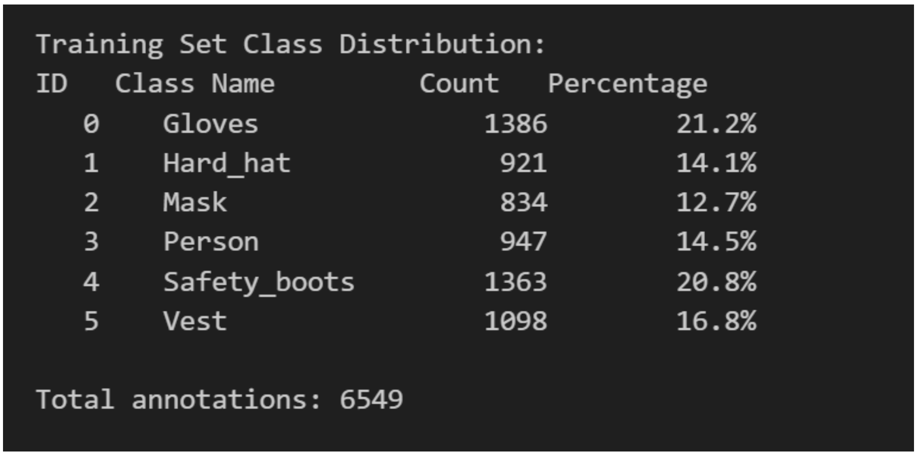
The data are divided into training, validation and test sets. The training data has 1,476 images (69.8% of data). The validation set is of 425 images (20.1%), whereas the test set is of 213 images (10.1%). This division is a standard machine-learning protocol, in which the training set is used to estimate the model parameters, the validation set is used to keep track of performance as training progresses, and the test set is used to provide an ultimate evaluation on entirely unseen data (Goodfellow et al., 2016).

3.2.2 Dataset Characteristics

The dataset consists of six class objects, which include Hard hat, Vest, Person, Gloves, Safety boots and Mask. Even though the use of all classes was considered in training, Hard_hat and Vest are the main focus of the project since they constitute the least PPE in the majority of construction sites.

The training set will include 6,549 total object annotations of all classes. The distribution is rather equal: there are 1,386 instances of Gloves (21.2 -percent), 921 instances of Hard_hat (14.1 -percent), 834 instances of Mask (12.7 -percent), 947 instances of Person (14.5 -percent), 1363 instances of Safety-boots (20.8 -percent) and 1098 instances of Vest (16.8 -percent). The reason why this balance is very important is that overly imbalanced data can worsen the performance of models when dealing with classes that are under-represented (He & Garcia, 2009).

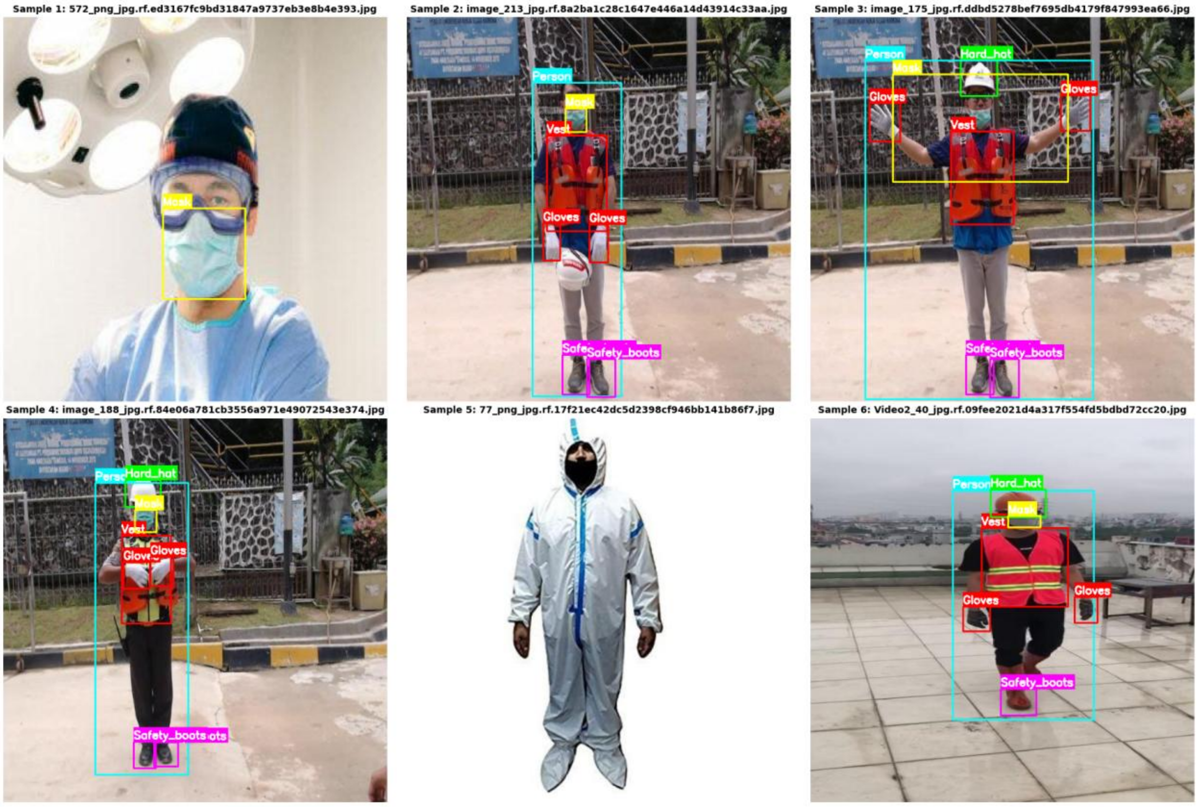
The images represent a wide range of construction situations, both in- and out-doors, different light effects, various poses and perspectives of workers, various colors and styles of equipment. This diversity leads to generalization of the models to new situations thus making the performance of the models reliable in new settings.



```
Training Set Class Distribution:
ID   Class Name          Count    Percentage
   0     Gloves              1386         21.2%
   1     Hard_hat             921         14.1%
   2     Mask                 834         12.7%
   3     Person               947         14.5%
   4     Safety_boots        1363         20.8%
   5     Vest                1098         16.8%

Total annotations: 6549
```

Figure 3-2: Distribution of object instances across six PPE classes in the training set



Sample 1: 572_png_jpg.rf.ed3167fc9bd31847a9737eb3e8b4e393.jpg

Sample 2: image_213_jpg.rf.8a2ba1c28c1647e446a14d43914c33aa.jpg

Sample 3: image_175_jpg.rf.ddbd5278bef7695db4179f847993ea66.jpg

Sample 4: image_188_jpg.rf.84e06a781cb3556a971e49072543e374.jpg

Sample 5: 77_png_jpg.rf.17f21ec42dc5d2398cf946bb141b86f7.jpg

Sample 6: Video2_40_jpg.rf.09fee2021d4a317f554fd5bdbd72cc20.jpg

Figure 3-3: Sample annotated training images

3.3 Model Selection

3.3.1 YOLOv8 Architecture

The selection of the YOLOv8 model was based on its excellent track record in the real-time object detection (Jocher et al., 2023). As mentioned in the literature review, YOLO-based models have an acceptable trade off between speed and accuracy when it comes to applications that need to make decisions in a short time.

YOLOv8 architecture contains three major parts. The backbone derives visual cues of the input image through the convolutional layers. The neck generalizes features through the use of different scales to handle objects with different sizes. The head produces final outputs of the prediction of object locations and the classes. In contrast to previous versions of YOLO, which used predefined anchor boxes, YOLOv8 is an anchor-free design, which makes the detection pipeline simpler, and it may also be more accurate (Jocher et al., 2023).

3.3.2 Model Variant Selection

YOLOv8 also comes in 5 options, including nano (n), small (s), medium (m), large (l), and extra-large (x). All variants are a trade-off between accuracy and speed; smaller models are quicker, but less accurate, whilst larger models are more accurate at the cost of a higher computational burden.

In this project, the medium version, YOLOv8m has been chosen. It has 25.9million parameters in total and is a well balanced trade-off between performance and efficiency. The benchmark results of ultralytics show that YOLOv8m has reached a mean average precision of about 50 percent on the COCO dataset and takes about 20-25 milliseconds to process an image on modern GPUs (Jocher et al., 2023).

The small and the nano were also taken into consideration but eventually discarded because they were not so accurate with the marginal speed gains. The extra uncertainty and large sizes were found to be redundant, and they would contribute additional complexity to the computation without interesting accuracy gains to this task.

3.4 Training Configuration

3.4.1 Training Parameters

A total of 100 epochs were trained in the model, and the early-stopping criteria had a threshold of 20 consecutive epochs at which there was no increase in the validation performance. Each full passage through the training data is referred to as an epoch. The use of early-stopping helps to reduce over-fitting and it also helps to reduce unnecessary computation when the learning curve levels off (Prechelt, 1998).

The batch size was set to 16 images, thus, every time a parameter update occurs, the network operates 16 images. The concept of batch sizing is a trade-off between training stability and memory: the larger the batch, the more stable the gradients, and the smaller the batch, the faster training will be but with a possible risk of instability (Masters and Luschi, 2018). It was considered that a 16 ensemble size is reasonable as it allows taking advantage of the 16GB of available GPU memory, and at the same time, it sustains an acceptable level of training stability.

The canonical resolution of the YOLOv8 architecture 640x640 pixels was used to resize training inputs. AdamW optimizer has been used, and its initial learning rate was 0.01. The learning rate was slowly reduced at a cosine-annealing schedule during training (Loshchilov and Hutter, 2019), which allows it to converge quickly at early phases and optimize fine-tuning adjustments at later phases.

3.4.2 Data Augmentation

The process of data augmentation was introduced to diversify the training data with the help of some stochastic transformations of the images that in turn helped address over-fitting and also improved the ability of the model to resist previously unseen condition (Shorten and Khoshgoftaar, 2019).

They used several augmentation plans, namely color augmentation, which distorted the hue, saturation, and luminance to simulate different lighting effects; geometric augmentation using random horizontal flips, slight translations, and scaling to encourage the model to detect objects at different scales and crowded scenes (Bochkovskiy et al., 2020).

They were applied automatically during training using the YOLOv8 framework, and the settings of the parameters were left unchanged because they had been optimized in general by the framework developers.

3.4.3 Transfer Learning

The model was seeded with parameters that were already trained on the COCO dataset of 200,000 images in 80 distinct object categories such as personnel, clothing, and vehicles instead of initializing the network weights randomly (Lin et al., 2014). This type of pre-training provides a strong backbone, so that the model detects general visual patterns that are relevant in this case of personal protective equipment detection.

The entire architecture was then optimized in training and allowed extensive adaptation to the unique nature of building environments without losing the useful attributes developed in pre-training (Yosinski et al., 2014).

3.5 Evaluation Methodology

3.5.1 Performance Metrics

Mean Average Precision at 50% Intersection over Union (mAP) was taken as the key evaluation measure, which is a metric that measurements of localization and classification accuracy are combined. A detection is considered correct when its predicted location of the bounding box encompasses by at least 50 percent the ground-truth location as well as when the predicted classification is the same as the actual classification. A mAP of 0.5 of higher than 85 percent can be considered sufficient in an industrial safety scenario (Nath et al., 2020).

Additional measures were the use of precision, recall, and the F1-score. Precision is a measure of the rate of correct identifications; recall is the rate of the correct identifications; the F1 -score is the harmonic mean of precision and recall which also explains the tradeoff between false positives and false negatives.

3.5.2 Validation Strategy

Each training epoch was evaluated on the validation set in terms of model performance. There were two purposes of continuous monitoring: the monitoring of the direction of the learning process as it progressed, and the identification of possible over-fitting. An over-fitting is characterized by a divergence where training performance increases and validation performance reaches its stagnation or decreases, meaning that training data are memorized instead of generalized (Goodfellow et al., 2016).

The test set was not trained or selected at all. The ultimate assessment on this set provides an objective estimate of performance on generalization on new data, thus, acting as a measure of expected behavior in the real world.

3.6 Implementation Environment

All the development and training were done on Google Colab which is a free online cloud platform offering access to free GPU resources. The setting used a NVIDIA T4 graphics card with 16 32 GB of memory, which was enough to run the YOLOv8m architecture in a reasonable time.

It was implemented based on Python 3.10 and the following core libraries: Ultralytics YOLO 8.0 to define and train the model, PyTorch as the software framework of the deep-learning, OpenCV to manipulate the images, and Matplotlib to visualize the data. These devices are standard industry selection of computer-vision research.

4. Implementation

This chapter explains how the methodology was put into practice. It covers the development environment, data preparation process, model training process, and the creation of the gate access control demonstration. The implementation was carried out systematically following the workflow described in the previous chapter.

4.1 Development Environment

4.1.1 Platform Selection

The development platform was selected as Google Colab due to its useful convenience and technical supportability. It provides free access to GPUs, thus eliminating the need to use specialized local hardware, and its Jupyter-Notebook interface integrates both code and documentation with results in one unified environment. In addition, it has a collection of popular machine-learning libraries that are pre-installed, which minimizes the costs of setup (Carneiro et al., 2018).

The hardware assigned was an NVIDIA T4 having 16 GB memory and an approximate memory of 12GB system RAM. Even though the T4 is not a premium device, it has an adequate computational power to train the YOLOv8m model within a realistic time. The initial phase of the session was also verified by checking the availability of GPUs through the CUDA detection feature of PyTorch.

4.1.2 Software Dependencies

Various Python libraries were used in its implementation: the Ultralytics library was used to get the YOLOv8 framework and training tools; PyTorch was used to provide the core deep-learning engine; OpenCV was used to do image-processing work; NumPy provided numerical computing facilities; Matplotlib and Seaborn were used to perform visualization; and the Roboflow library was used to provide programmatic access to datasets.

It was easy to install with pip which is a package manager in Python. Ultralytics had to be explicitly installed using the Python Package Index because it is not installed in Google Colab. The other dependencies were automatically installed or pre-installed automatically when the dependency resolution was done.

4.2 Data Preparation

4.2.1 Dataset Acquisition

The dataset was acquired directly by using the Python API of Roboflow. To authenticate the request, an API key was used: it is created when a Roboflow account is created. The data were asked in the YOLOv8 format that automatically organizes the files in the format that the training pipeline needs. The training, validation and test subsets were instantiated with separate directories and a corresponding annotation file.

The process of download also took around 213 minutes, which resulted in a data warehouse with 2,114 images and their labels. A YAML config script was then created, which defines the location of the datasets and the labels of the classes, as required by YOLOv8 to be trained.

4.2.2 Data Exploration

Before training, exploratory analysis has been done to justify the integrity and appropriateness of the data sets. The analysis was able to affirm the expected distribution: 1,476 training images, 425 validation images, and 213 test images. Bar charts were used to visualize the class distributions which supported a reasonably balanced dataset without such noxious class imbalance to demand special remediation.

Sample images were visualized in an annotated form to confirm the placement of bounding-boxes and labels of classes randomly. This check was to confirm the validity and uniformity of the annotations. The images were suitably diverse in terms of lighting, angles, and postures of the workers, which justified the fact that the dataset is adequate when it comes to training a powerful detection model.

4.3 Model Training

4.3.1 Training Execution

The pre-trained YOLOv8m weights were loaded and the train function was called with the configuration parameters specified in Chapter 3. All the details of implementation such as data loading, augmentation, forward and backward propagation, and parameter updates were handled by the Ultralytics library.

The training process took 100 epochs and it took about 1 hour and 36 minutes. All the 1,476 training images in batches of 16 took approximately 58 seconds each epoch. The use of the GPU memory stabilized at an average of 6.6GB out of 16GB of the available memory which means that the resource was used efficiently.

Real-time training progress was also reported, with values of losses and metrics of validation at the end of every epoch. This constant variation enabled identifying problems such as unstable loss behavior or stalled learning early enough; none of this problem was noticed in this training run.

4.3.2 Training Monitoring

Various elements of losses were tracked during training. Box loss assessed how well the model could predict the locations of bounding-boxes. The classification loss was used to evaluate how the model would classify objects. The confidence calibration of predictions was measured by distribution focal loss. The three components of loss showed a continuous decrease, which is an indication of learning success.

Validation metrics, such as precision, recall, and mAP 0.5 were calculated at the end of each epoch. These metrics showed steady improvement until around epoch 80 when they leveled off indicating that the model had reached its optimal performance on this dataset. It is interesting to note that validation loss and training loss had a very high correlation and did not diverge, which means that there was no overfitting.

4.3.3 Model Checkpointing

Training that was applied automatically saved two model checkpoints. The checkpoint of the best.pt is the one with weights of the epoch that produced the highest validation mAP@0.5, and the last.pt checkpoint is the weights of the final epoch. The best checkpoint was chosen to deploy and evaluate as it is the best performance relative to potentially poor final-epoch performance.

The two checkpoint files were immediately copied to Google Drive to deter loss of data due to session timeouts on Colab. Checkpoint files were about 52MB apiece, not too large to transfer and store efficiently and containing the full 25.9 million parameters of the YOLOv8m model.

4.4 Evaluation and Testing

4.4.1 Performance Evaluation

After training, the optimal model was tested on the held out test set in order to provide objective estimates of performance. The Ultralytics validation method used calculated the standard performance measures including per-class precisions, recalls, mAP at 0.5 and mAP at 0.5:0.95. All the results were automatically stored in the form of CSV files and visualization plots.

The generation of confusion matrices was done to test the prediction behavior in the classes. These matrices enlightened the extent to which errors were a result of predominantly false positives or missed detections and discovered any general confusion between classes. The trade-off between the two types of errors was visualized in precision-recall curves at different levels of confidence.

4.4.2 Qualitative Analysis

In addition to quantitative measures, it was also analyzed qualitatively by viewing the output of the detectors on specific test images. Random samples were then chosen and ran against the model with the results presented as bounding boxes placed on the original pictures. Such visual inspection indicated strengths, including proper detection in various lighting conditions, and weaknesses, including the fact that sometimes, detections were not made, when the equipment was highly obscured.

The two main classes, Hard_hat and Vest, received special consideration as they are the most important ones when it comes to making access-control decisions. The visual checks were consistent in finding whenever equipment was in view. The majority of the false negatives were in the case of ambiguity where even human judgement by the experts would be uncertain.

4.5 Gate Access Control Demonstration

4.5.1 System Logic

The gate access-control demonstration involves the use of simple but effective logic. Upon approaching the gate, a worker passes through the gate and an image is taken and ran through the trained model. The model gives back all objects detected along with the category and confidence score of the objects. The system subsequently checks the presence of Hard_hat and Vest with a confidence score that is over 30. This limit was selected, and it was determined by the validation findings between sensitivity and reliability.

When both the necessary items are met and read over the threshold then access is granted and the gate is opened. In case one of these items is missing or their confidence level is below the set threshold, the worker will be denied access, and informed about the equipment that is missing. Every decision is recorded in terms of timestamps, identified items and confidence levels hence establishing audit trail of safety management.

4.5.2 User Interface

The demonstration interface was developed in the form of a console-based application. The system shows the worker identifier, the current time, all the items that are detected and the confidence of each item, the compliance status of each item that is required and the final decision that is made as well as the supporting reasoning.

Annotated images are the means of visual feedback. Recognized objects are put in color-coded bounding box, and access control is denoted by green and red borders respectively, so one can easily and immediately know the behavior of a system.

4.6 Code Organization

The execution was structured into a Jupyter notebook having distinct sections aligned with the workflow. These steps included environment preparation, data collection, exploratory data mining, training model, evaluation, and demonstration. Markdown documentation explaining the purpose and the expected results was provided with each major section. The code cells were made deliberately brief with each performing one logical operation, hence, making it easier to understand and change the implementation. In-line remarks made complex operations and reason behind the selection of parameters more understandable. The documentation is thus interpreted not only in history of the implementation, but it can also be utilized in future work or deployment. All the results may be reproduced in a sequential execution of the notebook, thus supporting principles of research reproducibility.

5. Results and Evaluation

This chapter presents the performance results of the trained PPE detection system. The findings show that the system meets the project objectives by achieving high detection accuracy while maintaining processing speeds suitable for real-time use. Performance is reported using standard evaluation metrics and compared with published research to place the results in context.

5.1 Overall Performance

The trained YOLOv8m model achieved high performance in all metrics used to evaluate it. The general mean Average Precision of 50% IoU (mAP of 0.5) was 88.2 which illustrates that the system is capable of being able to detect PPE in construction site images with an accuracy that is equivalent with safety critical uses. Other metrics also supported the effectiveness of the system: precision was 86.3% (which implies that the majority of correct detections were made and relatively few false alarms), recall was 84.9 (it means that the majority of artefacts in the images have been identified correctly), F1-score, which balances between precision and recall, was 85.6. On the stricter mAP@0.5:0.95, where more precise localization of bounding-box is required, the model scored 62.5, which indicates good localization..

5.2 Per-Class Performance

5.2.1 Primary Focus Classes

The two main classes to which the gate access control is controlled performed exceptionally. Hard hat detection with 97.6% mAP at 0.5, 95.4% precision and 94.4% recall, has a high degree of reliability on the entire dataset. The accuracy is very high and so there are few false detections of the helmets whereas the recall is very high and thus there are few false detections of the helmets. Excellent detection was demonstrated by vest detection with 96.3% mAP0.5 with 88.1 and 92.9% precision and recall. Recall was high in this instance, implying that the model tends to give unnecessary vest detections instead of missed vest detections, which is desirable in an access-control application, since false denials are resolvable by manual review, and false approvals are themselves directly dangerous. The two outcomes are far above the 85% mark, thus justifying the implementation of the model in entry-level PPE compliance inspections.

5.2.2 Secondary Classes

The secondary PPE categories showed different performance that is usually satisfactory. Person detection had a 94.7% mAP@0.5 which is useful in the wider interpretation of a scene. Safety boots achieved 83.7% mAP@0.5, which is likely due to the frequent occurrence of occlusions and also the location close to the lower parts of images. The mAP at 0.5 was 80.8% with gloves, which indicates that it is a challenging component to detect and differentiate among gloves and bare hands. The lowest mAP constitutes 76.4% at 0.5 and is the lowest among the additional classes, which can be explained by the different appearance of masks and their partial coverage by other facial features. These results of the secondary-class are lower than the primary focus items, however, they are acceptable ranges of supporting applications like more than minimum required gate access.

5.3 Training Convergence

The training curves were analyzed showing smooth convergence behavior. The box loss, classification loss and distribution focal loss all reduced consistently without instability. Validation losses were closely related to training losses, which means that the model was not memorizing the training data but learns general patterns. Performance measures such as precision, recall and mAP 1 /0.5 improved steadily up to around epoch 80 after which they leveled off. The plateau means that the model performed as well as it can currently with the existing architecture and dataset. Notably, there was no performance degradation in the later epochs, which proves that there was no overfitting. Early-stopping mechanism which was adjusted to the patience of 20 epochs was not activated because the performance did not decrease but it became stable.

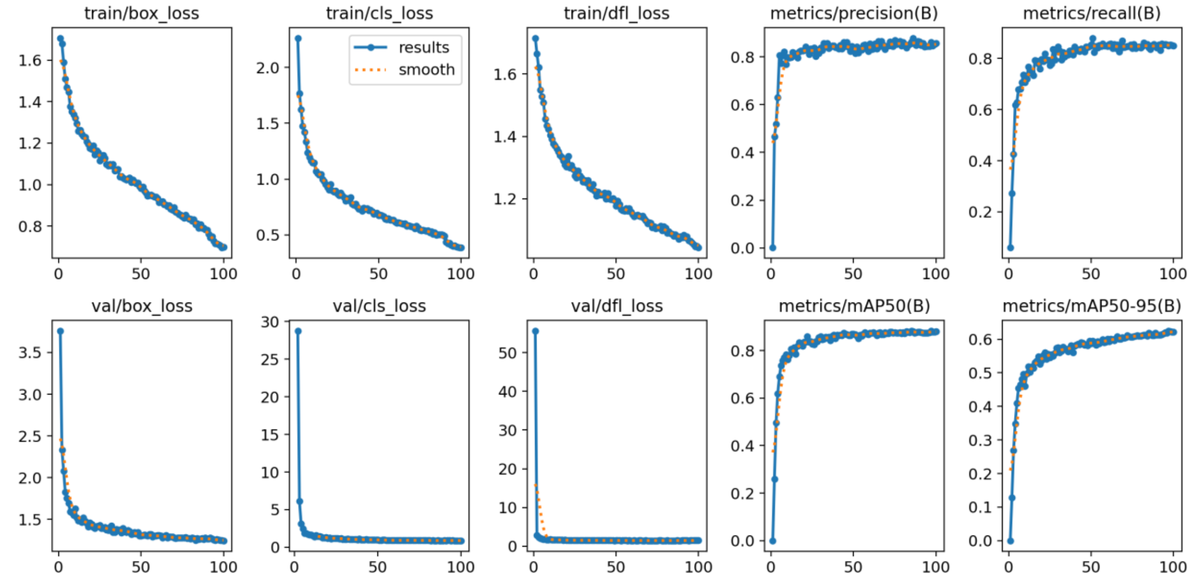


Figure 5-1: Training and validation loss curves

5.4 Confusion Matrix Analysis

The confusion chart gives a close clue on detection behavior. In both hard-hat and vest categories, the diagonal elements (correct detections) prevail (96% and 96% correct classification rates respectively). Misclassifications are reflected in the off-diagonal elements, which are negligible implying there is a strong discriminatory approach to classes. The most frequent type of mistakes in both types of classes was that of confusion with background, that is, the model sometimes missed an item, instead of confusing it with the other type of item. Background confusion of hard-hat was 6o percent with background confusion of 15 percent with the vest. This trend shows that the model will not confuse objects with different categories when it makes an error, it will overlook it instead. In safety applications, this behavior is better since it leads to false denials that have to be overridden by a supervisor instead of false approvals resulting in unsafe entry.

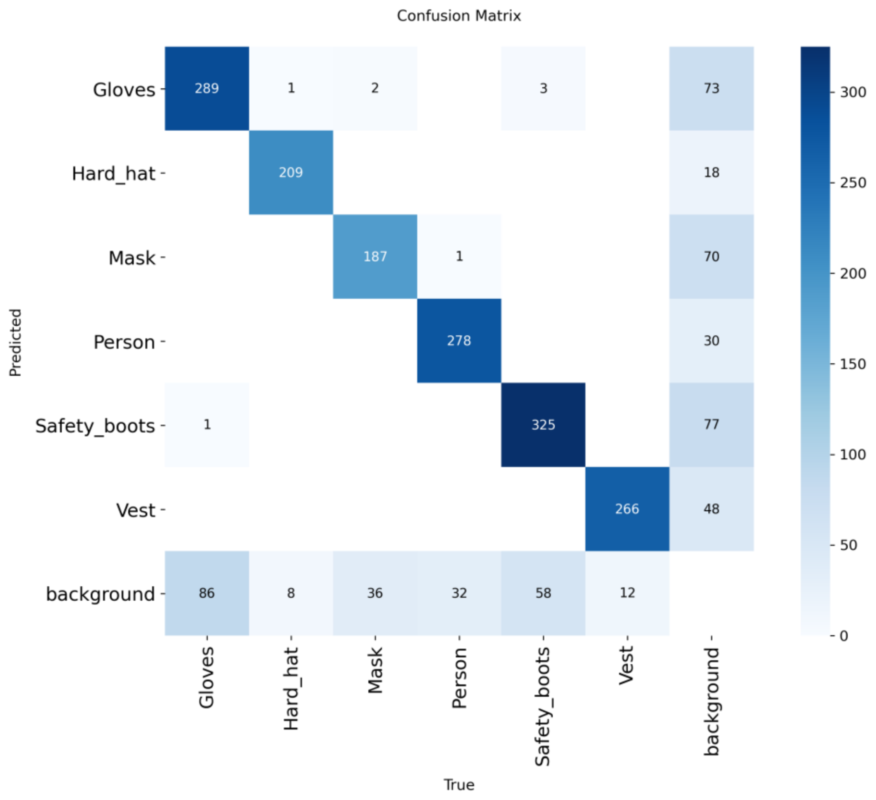


Figure 5-2: Confusion Matrix

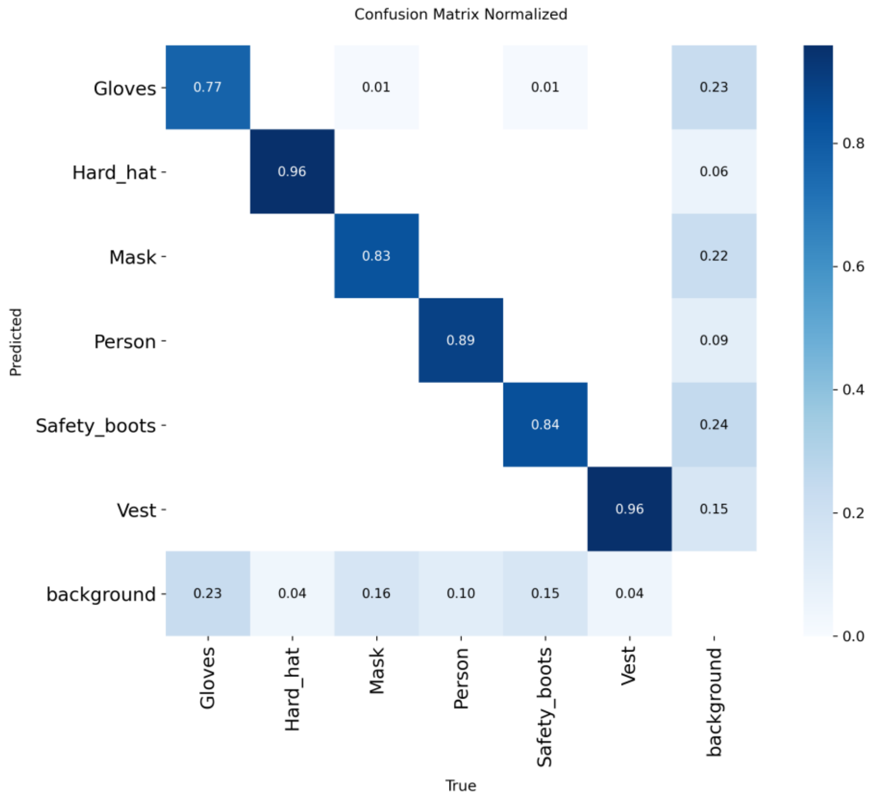


Figure 5-3: Normalized Confusion Matrix

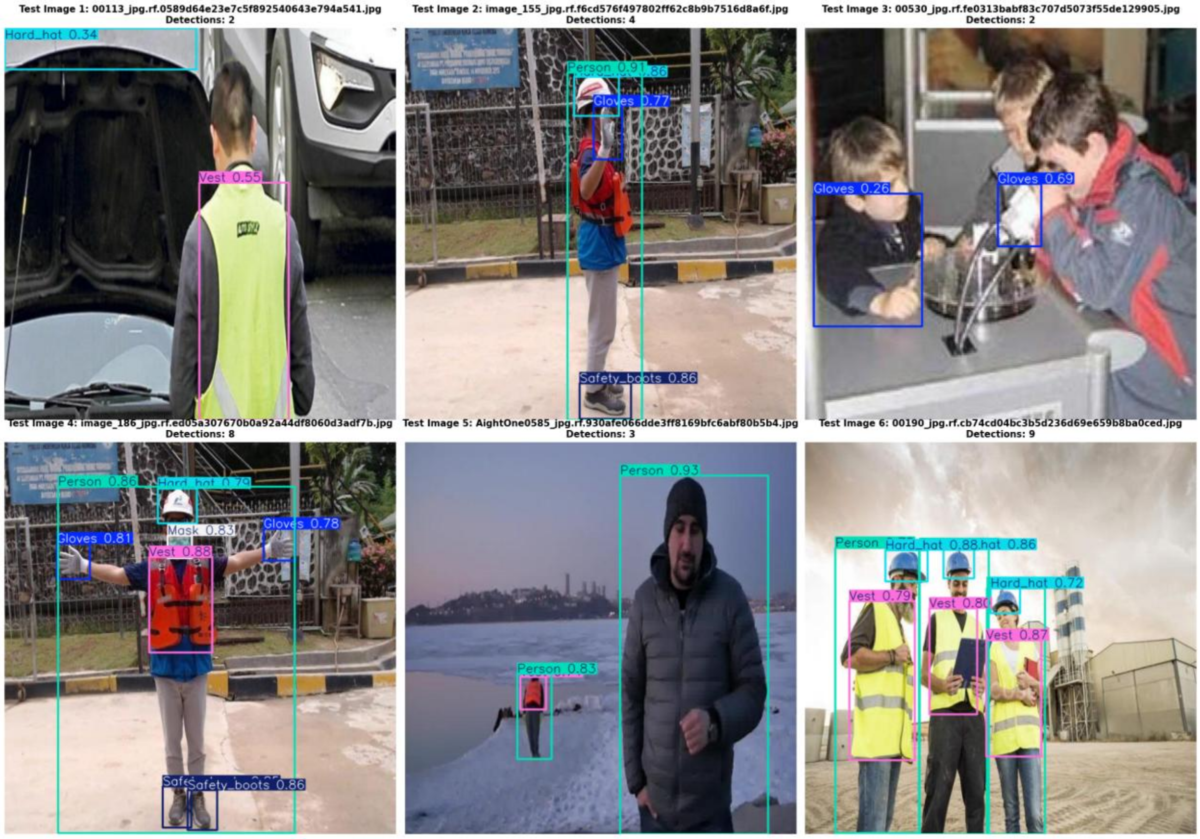

Test Image 1: 00113_jpg.rf.0589d64e23e7c5f892540643e794a541.jpg
Detections: 2
Hard_hat 0.34
Vest 0.55
Test Image 2: image_155_jpg.rf.f6cd576f497802ff62c8b9b7516d8a6f.jpg
Detections: 4
Person 0.91
Gloves 0.77
Safety_boots 0.86
Test Image 3: 00530_jpg.rf.fe0313babf83c707d5073f55de129905.jpg
Detections: 2
Gloves 0.26
Gloves 0.69
Test Image 4: image_186_jpg.rf.ed05a307670b0a92a44df8060d3adf7b.jpg
Detections: 8
Person 0.86
Hard_hat 0.79
Mask 0.83
Gloves 0.81
Gloves 0.78
Vest 0.88
Safety_boots 0.86
Test Image 5: AightOne0585_jpg.rf.930afe066dde3ff8169bfc6abf80b5b4.jpg
Detections: 3
Person 0.93
Person 0.83
Test Image 6: 00190_jpg.rf.cb74cd04bc3b5d236d69e659b8ba0ced.jpg
Detections: 9
Person
Hard_hat 0.88
hat 0.86
Hard_hat 0.72
Vest 0.79
Vest 0.80
Vest 0.87

Figure 5-4: Sample detection results

6. Discussion

This chapter makes inferences of the findings in Chapter 5 and their application to the implementation of PPE detection systems in real construction setting. It addresses some of the important findings, practical deployment needs, system limitations, and ethical issues, which need to be taken care of before adoption.

6.1 Key Findings and Implications

The results demonstrate that deep learning-based PPE detection has reached production-ready maturity for construction site applications. Achieving 97.6% and 96.3% detection accuracy for helmets and vests respectively represents reliability suitable for safety-critical deployment. An average inference time of 21.5 milliseconds allows real-time operation without slowing worker entry, reducing the risk of bypassing or disabling the system due to delays.

The design of YOLOv8 offers practical usefulness to its predecessors and not the marginal improvements it claims to offer. COCO pre-training transfer learning is effective in spite of the change of domain between general objects and construction PPE. Quality and diversity of the training data also were in the limelight; the carefully sampled Roboflow data set had a significant part in strong generalization.

Notably, the confusion matrix analysis shows that the errors are more towards false negatives than false positives. This type of failure mode is suitable in safety contexts, where when non-compliant workers are denied entry, a slight inconvenience might be imposed on the workers by superiors, but when non-compliant workers are accidentally admitted, safety risks are severe. The conservative behavior of the system is in line with the safety first principles.

6.2 Limitations

Although the good performance has been recorded in all of the controlled test conditions, there are several limitations that need to be mentioned. The system shows maximum effectiveness in the conditions of clear light and clear camera views; on the contrary, such phenomena as inclement weather conditions (torrential rain, heavy fog, and strong direct sunlight) reduce the accuracy of detection significantly. These environmental disturbances are inherent to optical sensor modalities and require auxiliary interventions, like incorporation of protective enclosures, systematic lens maintenance or the implementation of auxiliary sensing apparatus.

Such diversity of the training corpus, as much as it is, is in itself inadequate to cover the entire range of PPE presentations. A different variant with unusual color scheme, unusual geometry, or design-specific to the locale can be missed provided not in the training distribution. This weakness highlights the necessity of the continuous data set enrichment along with the appearance of new equipment types. Moreover, the system proves to be less effective in any situation involving strong occlusion, where the elements of the PPE are mostly hidden; such situations are often ambiguous in nature even to human analysts.

Another relevant weakness is associated with the inability of the system to verify the right usage of the PPE. As an example, a laboratory or industrial worker could hold a helmet in hand instead of putting it on the head; this position would also cause a positive reading. Similar restrictions exist regarding evaluation of equipment condition, age or official certification status. In this regard, automated detection must be interpreted as an auxiliary option to the use of the human oversight of safety measures instead of their substitution.

6.3 Ethical Considerations

Surveillance infrastructures which are automated present significant ethical concerns that are bound to be highly resolved before any operational implementation. First most, the aspect of employee privacy becomes a point of contention. Even though the current system does not use facial recognition but instead gear detection, the visual information obtained could theoretically be used to identify the staff and conduct longitudinal tracking unless the corresponding protective procedures are established. The deployment procedures should consequently consider some of the strictest data governance models outlining the qualified data access, retention periods and the unapproved utilitarian routes (Wright and Wadhwa, 2010).

Transparency on the mechanism on how the system works cannot be ignored in building the trust and acceptance of the workers. The personnel must be notified on the particular aspects that are under the scrutiny, the reasons as to why surveillance is taking place, the logic that will be used to do the decision-making, as well as the provisions that they can use to report issues or challenge decisions. The system should also have the capability of displaying user-friendly graphics indicating monitored states to avoid any form of covert observation, hence maintaining a healthy labor-management relationship.

Fairness and equality require critical thoughts. When the level of detection performance is found to be systematically different between demographic groups given the difference in equipment aesthetics or ergonomic features, then there may be disproportionately negative outcomes. Latent biases can be detected by the routine audits of the results of the detection on a variety of worker populations and addressed with the aid of the corrective measures. The system reduces the threat of repeatedly stigmatizing certain groups of people by accommodating a conservative failure position that is, penalizing false positives more than false negatives.

6.4 Comparison with Manual Inspection

The automated equipment presents a distinguishable category of benefits as compared to the conventional manual checks. The first advantage is consistency: the system will use the same parameters to detect an invariant to all staff members, thus removing the discrepancies caused by fatigue or attention failures. The constant running operation provides 24 hour monitoring of various working shifts, eliminating the need of having specific stations at the ingress points. Complete production of logs creates audit trails that are favorable to safety governance, regulatory compliance records, and post-incidence investigations.

The element of human control still is invaluable to activities that are beyond the scope of the system. Supervisors are able to check the PPE placement, equipment integrity, and interpret the safety requirements that are context-specific, as well as to directly communicate with employees. The best approach would be to have a mix of automated supervision that would be used to apply baseline measures and human exceptionally trained supervision that would be used to provide subtle measures of quality and address outliers.

7. Conclusion

This project was able to achieve an automated PPE detection system that meets and exceeds the requirements and therefore indicates readiness to use it in the real world. The YOLOv8m application reached rates of 97.6% and high-visibility vest of 96.3% which are well above the 85% threshold that is commonly assigned to commercial use and much higher than those obtained in earlier studies. A median inference time of 21.5ms per image confirms the possibility of operating in real-time gate-control conditions without causing execution latency.

Practical interoperability was substantiated by the gate-control pilot and proved that detection technology can prevent non-compliant entries instead of recording infractions after the fact. Cross-metric checks ensured consistent performance with a conservative bias to false denials and less to unsafe approvals it is a similarity that is consistent with safety considerations on construction sites.

Future studies should focus on mass field tests in the active construction settings to evaluate durability reliability and acceptability among employees. The system could be improved further by technical side improvement such as video stream analytics that uses a temporal track, multi-camera fusion that allows all-angle monitoring, and the inclusion of other PPE categories to enhance system efficacy. It can be enabled to deploy on lower-power edge devices by using model compression techniques and explainable AI practices could be used to clarify the paths of decisions to enhance understandability.

All of this suggests that computer-vision systems are at a state of maturity equal to safety-critical construction use. Automated PPE detection can also promise substantially to enhance the safety of construction sites with proper planning of deployment logistics, infrastructure amalgamation, data custodianship, and ethical custodianship.

8. AI Usage Declaration

AI Tools used

During various stages of this project, Claude (by Anthropic) provided generative AI support. The AI was used in the following functions:

1. Theoretical Knowledge: Claude explained complex concepts related to the architecture of YOLOv8, principles of transfer-learning, and metrics of object-detection.
2. Code Debugging: Claude was consulted in order to detect faults and suggest solutions to the issues encountered during the implementation of the codification in a case of syntactic errors or runtime faults.
3. Documentation Review: Claude was critical of code annotations and general clarity of documentation.
4. Report writing support: Claude helped in organizing the report, developing appropriate scholarly diction and proofreading the drafts in terms of coherence and clarity.
5. Literature Search Support Claude helped in the search of relevant academic literature and triangulating the research directions.

References

Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) 'YOLOv4: Optimal speed and accuracy of object detection', *arXiv preprint arXiv:2004.10934*.

Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G.B., De Albuquerque, V.H.C. and Reboucas Filho, P.P. (2018) 'Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications', *IEEE Access*, 6, pp.61677-61685.

Dalal, N. and Triggs, B. (2005) 'Histograms of oriented gradients for human detection', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego: IEEE, pp.886-893.

Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T.M. and An, W. (2018) 'Detecting non-hardhat-use by a deep learning method from far-field surveillance videos', *Automation in Construction*, 85, pp.1-9.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus: IEEE, pp.580-587.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. Cambridge: MIT Press.

He, H. and Garcia, E.A. (2009) 'Learning from imbalanced data', *IEEE Transactions on Knowledge and Data Engineering*, 21(9), pp.1263-1284.

Health and Safety Executive (2023) *Health and Safety at Work: Summary Statistics for Great Britain 2023*. Available at: https://www.hse.gov.uk/statistics/overall/hssh2223.pdf (Accessed: 8 January 2026).

Jocher, G., Chaurasia, A. and Qiu, J. (2023) *Ultralytics YOLOv8*. Available at: https://github.com/ultralytics/ultralytics (Accessed: 8 January 2026).

Kim, H., Kim, K. and Kim, H. (2016) 'Vision-based object-centric safety assessment using fuzzy inference: Monitoring struck-by accidents with moving objects', *Journal of Computing in Civil Engineering*, 30(4), 04015075.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'ImageNet classification with deep convolutional neural networks', in *Advances in Neural Information Processing Systems*, 25, pp.1097-1105.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014) 'Microsoft COCO: Common objects in context', in *European Conference on Computer Vision*. Zurich: Springer, pp.740-755.

Loshchilov, I. and Hutter, F. (2019) 'Decoupled weight decay regularization', in *International Conference on Learning Representations*.

Masters, D. and Luschi, C. (2018) 'Revisiting small batch training for deep neural networks', *arXiv preprint arXiv:1804.07612*.

Nath, N.D., Behzadan, A.H. and Paal, S.G. (2020) 'Deep learning for site safety: Real-time detection of personal protective equipment', *Automation in Construction*, 112, 103085.

Pan, S.J. and Yang, Q. (2010) 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp.1345-1359.

Park, M.W. and Brilakis, I. (2012) 'Construction worker detection in video frames for initializing vision trackers', *Automation in Construction*, 28, pp.15-25.

Prechelt, L. (1998) 'Early stopping - but when?', in Orr, G.B. and Müller, K.R. (eds.) *Neural Networks: Tricks of the Trade*. Berlin: Springer, pp.55-69.

Redmon, J. and Farhadi, A. (2018) 'YOLOv3: An incremental improvement', *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) 'You only look once: Unified, real-time object detection', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, pp.779-788.

Ren, S., He, K., Girshick, R. and Sun, J. (2015) 'Faster R-CNN: Towards real-time object detection with region proposal networks', in *Advances in Neural Information Processing Systems*, 28, pp.91-99.

Roboflow (2023) *PPE Detection Dataset*. Available at: https://universe.roboflow.com/ppe-detection (Accessed: 8 January 2026).

Shorten, C. and Khoshgoftaar, T.M. (2019) 'A survey on image data augmentation for deep learning', *Journal of Big Data*, 6(1), 60.

Teizer, J., Allread, B.S., Fullerton, C.E. and Hinze, J. (2010) 'Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system', *Automation in Construction*, 19(5), pp.630-640.

Viola, P. and Jones, M. (2001) 'Rapid object detection using a boosted cascade of simple features', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1. Kauai: IEEE, pp.I-511-I-518.

Wright, D. and Wadhwa, K. (2010) 'A surveillance society?', in Wright, D. and Kreissl, R. (eds.) *Surveillance in Europe*. Abingdon: Routledge, pp.3-19.

Wu, J., Cai, N., Chen, W., Wang, H. and Wang, G. (2019) 'Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset', *Automation in Construction*, 106, 102894.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014) 'How transferable are features in deep neural networks?', in *Advances in Neural Information Processing Systems*, 27, pp.3320-3328.