TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 491

WorkerSecure

Low Level Design Report

by

Hasret Türkmen

Asaf Emin Gündüz

Selin Sevcan Çakan

Niyazi Ahmet Metin

TABLE OF CONTENTS

1. Introduction

1.1. Project Overview

WorkerSecure is a project that aims to improve the safety of workers in construction areas. The improvement procedure involves a real-time monitoring system to be able to identify workers not wearing necessary safety equipment. YOLO model will be used for the monitoring system, enabling the detection process to be completed with image processing. The system will be able to detect the anomalies and notify the workers/person in charge to ensure safety.

1.2. Objectives of the Low-Level Design

This low-level design report aims to detail the system's components and interactions, and the integration and optimization of the YOLO model within the WorkerSecure. Based on the High-Level Design Report, the LLDD will serve as a technical roadmap. The YOLO model's details, process of data training, and optimization of the model will be discussed. The document also includes:

- The data collection and management workflow.
- Details of the deployment process to ensure the effectiveness of the model.
- Possible risks and ethical issues the project may have.

1.3. Overview of YOLO in the Context of the Project

The YOLO model plays a crucial role within WorkerSecure. Fast and accurate detection is enabled by the model to allow real-time processing and, the capability of detecting helmets under different conditions. In addition, the model's ability to process video frames efficiently without sacrificing detection quality highlights the model's role in the WorkerSecure project.

2. YOLO Model Specifications

2.1. YOLO Version and Configuration

As a result of our research, it was decided that using YOLOv7 would be compatible with our project due to its advancements in detection speed, accuracy, and efficiency in real-time object detection. The abilities that the model provides are compatible the the monitoring needs of WorkerSecure, where fast action is a lifesaver. The model will be configured with the defaults provided by the YOLOv8. These defaults are designed to be comprehensive, ensuring a strong balance between detection accuracy and processing speed. This approach will allow us to focus on applying the model directly to WorkerSecure without the need for complex optimization.

2.2. Model Architecture Overview

The architecture of YOLOv7 is different from the traditional approaches, that analyze images in a step-by-step manner. YOLO stands out because it processes an entire image in one go. This unique approach significantly improves speed and efficiency, making it a perfect fit for WorkerSecure's needs. The model's architecture with layers makes it able to extract features from images while being adaptable to different environments.

2.3. Training Data Specifications

For the training of the model in our WorkerSecure project, specific steps were taken to ensure the model performs well under different conditions. The different conditions include weather and time of day. To be comprehensive, our dataset will include:

High-quality images captured in brightened low-light conditions to simulate nighttime work environments. This ensures the model can accurately detect helmets even in the dark.

To improve the adaptation of YOLOv7 for night detection, some daytime images will be processed with a night filter. The filtering will be implemented through a night vision effect.

Training will also focus on a helmet being worn correctly. The system should trigger the alarm in every case where the helmet is not on the head, such as being carried in hand. This specification is to provide safety measures in the construction site without mistake.

The dataset including over 7000 images in an open-source dataset is carefully chosen to increase the model's accuracy across all working conditions.

2.4. Annotation and Labeling Methodology

The details of how it is planned to label our images for the model YOLOv7 are as follows:

- Selection of the Images: An open-source dataset including 7000 images was chosen. The dataset has various images of workers with and without helmets and under different light conditions.
- Labeling Process: Each bounding box is assigned a label such as “helmet” for helmets worn correctly and “no helmet” for incorrect helmets worn.
- Quality Check Process: Reviews are conducted to ensure accuracy and consistency across the dataset. This process involves team members checking each other’s work to minimize errors.
- Training Process: The model will be trained to recognize the helmet’s proper usage, not only its presence. This will ensure that the alarm is triggered in the right conditions.

3. Data Processing and Management

3.1. Data Collection Process

WorkerSecure captures video frames from the cameras that are placed around the workplace. The company should place the cameras in strategically, so that leaves as fewer blind spots as possible.

It is important that the frames collected from the cameras are processable, and clear enough to match the dataset WorkerSecure has developed.

3.2. Data Preprocessing Steps

- **Image Augmentation**: In order to have different angles, lightings and helmets, WorkerSecure uses rotating, cropping and lighting on its dataset.
- **Data Cleaning:** WorkerSecure uses data cleaning to remove any falsely labeled or irrelevant data from its dataset. It is needed to improve the accuracy of the model.
- **Normalization:** This technique is needed to standardize the pixel values of the frames we get from the cameras.

All these methods are used to utilize the YOLO architecture so that it creates better results.

3.3. Data Storage and Management

The helmet violations that WorkerSecure has detected are stored in a database, such as at which region in the workplace it happened and the time, with ensuring the data security and integrity. The purpose is to support real-time monitoring and historical analysis if any charts or diagrams are needed by the company.

4. Training Process

4.1. Training Environment Setup

For the training of our YOLO-based hardhat detection model, we've used the current resources that we have, but school provides us better environment so we will use those in the future but for now we have:

Hardware Configuration:

GPU: NVIDIA GTX 1050TI GPUs with 4 GB of memory. This GPU takes 2 days of non-stop working process to train a dataset with 7000 images inside.

CPU & RAM: Intel i7-7700HQ, this was enough for smooth handling of large datasets.

Software Configuration:

Deep Learning Framework: PyTorch 1.8.1, appreciated for its dynamic computation and ease of use.

YOLO Implementation: YOLOv7 from Ultralytics, provides efficient framework for our model.

Tools & Libraries: CUDA Toolkit 11.2 and cuDNN 8.1 for GPU acceleration.

This setup is enough for now for model training and experimentation, allowing us to develop an accurate hardhat detection system.

4.2. Training Parameters and Hyperparameters

In training our YOLO-based hardhat detection model, careful consideration was given to selecting appropriate parameters and hyperparameters to optimize performance. The choices are as follows:

Key Parameters:

- **Batch Size:** Set to 32, balancing the trade-off between memory usage and model update frequency.

- **Number of Epochs:** 100 epochs, ensures sufficient iterations for the model to learn from the dat.

- **Image Size:** 640x640 pixels.

Hyperparameters:

- **Learning Rate:** Initiated at 0.01 and adjusted using a cosine annealing schedule to decrease gradually, promoting convergence to a global minimum.

- **Optimizer**: Adam optimizer, selected for its adaptive learning rate capabilities, improving convergence speed and stability.

- **Weight Decay:** 0.0005, applied to regularize the model and prevent overfitting by penalizing large weights.

These parameters and hyperparameters were selected through number of experiments for the optimal model.

4.3. Evaluation Metrics and Validation Process

To know the accuracy and effectiveness of our YOLO-based model for hardhat detection we use an evaluation and validation process.

Precision Indicates how accurate the model's positive predictions are by measuring the correctly detected hardhats out of all positive detections made.

Recall Assesses the model's sensitivity by measuring its ability to identify all actual hardhats present in the dataset.

F1-Score Combines precision and recall into a single metric, highlighting the balance between accurately identifying positive cases and minimizing false positives. It is particularly valuable in scenarios with uneven class distributions.

Mean Average Precision (mAP) Computes the average precision across various thresholds, creating a single value that represents the model's overall performance of object detection tasks.

Validation Dataset A distinct dataset, independent of the training data, is employed to assess the model's performance. This dataset captures the diversity in scenarios and conditions that the model encounter in real-world applications.

5. Deployment and Integration

For the deployment of the WorkerSecure's YOLO model, a methodical approach will be followed. Some subtitles are prepared in this section to explain the strategy steps taken, integration with existing systems, and continuous maintenance.

5.1. Model Deployment Strategy

Local Machine Deployment: First, the model will be tested and conducted on one of our group member's local computers to ensure the model can be efficiently managed and monitored during development.

Model Testing: In the beginning, we'll test the model with some sample pictures and simple scenarios like test videos. Then, finding some videos from open-source platforms, we'll integrate the model with some available videos and available camera systems found on open-source platforms. Whereas the detection capability of the system will be tested in this process, it also will ensure compatibility with existing video surveillance setups, enhancing the system's applicability without necessitating complex infrastructure changes.

Manual Monitoring for Updates: We'll be manually monitoring the system performance with iterative updates applied based on real-time feedback. With this approach we'll be able to refine the detection accuracy and system reliability continuously.

Utilizing Open-Source Technologies for Model Deployment: We plan to use open-source platforms and algorithms in the WorkerSecure. OpenCV and YOLO algorithm will be utilized for image processing and object detection. We choose these platforms because of their advantages of being best solution for this problem in cost-effective manner.

5.2. Integration with Existing Systems

The deployment will include a simple process for integrating the model into existing monitoring frameworks. Thus, WorkerSecure application is compatible with almost every camera system with night vision. So, for a company to use WorkerSecure, the company doesn't need to buy new, customized cameras for WorkerSecure because the model will be designed to work with standard video feed formats, ensuring it can be easily connected to existing camera systems without the need for specialized equipment.

5.3. Monitoring and Maintenance Plan

- **Regular System Audits:** Scheduled audits will be conducted to assess system performance and accuracy, identifying any deviations or issues that may arise over time.

- **Feedback Loop with Users:** The feedback page on WorkerSecure web application will be established and used, allowing for the users and workers to give feedback for identification of areas for improvement and adjustments to enhance system functionality.
- **Software Updates:** The system will be periodically updated to incorporate improvements, bug fixes, and adjustments based on user (and worker) feedback and audit outcomes.

6. Performance Optimization

This section briefly outlines our team's approach to optimize WorkerSecure. It addresses important areas such as archiving and capturing data, notifying the customer, and customization.

- **Capturing Data Continuously:** WorkerSecure has access to cameras around to workplace, meaning the video frames from the cameras are always available to the system.
- **Immediate Alert:** When a violation is detected, the user will be alerted via notification. It does not only notify the user, but also suggests actions or information about the dangers of not wearing necessary equipment.
- **Data Archiving**: When a violation is detected, the information (such as place and time) will be stored in a database.
- **Robustness**: WorkerSecure needs to be robust under different conditions, therefore it is trained on a large and diverse dataset to improve the accuracy.
- **Customization**: Companies will be able to pick which type of notification they want according to their surroundings and working environment.
- **Feedback**: WorkerSecure team considers feedback to be a good way to improve the system, creating a better user experience.

7. Risk Analysis and Mitigation

7.1. Potential Risks and Challenges

Using our YOLO-based hardhat detection system for construction site security may involve some risks and challenges.

Data Privacy: Gathering images or videos from construction sites might raise concerns about privacy, especially if individuals are identifiable. Breaking data protection rules could lead to legal issues and harm the organization's image.

Accuracy and Bias: The system may be biased or inaccurate, particularly if it's trained on data that doesn't represent different perspectives. This can lead to uneven performance in various situations or among different people. Biased or incorrect detections can affect safety standards and cause hazards on construction sites.

Hardware Failures: The system relies on hardware components like cameras and processors, which can fail. These failures can halt the monitoring service and make the system less effective in maintaining safety.

Scalability Concerns: As the project grows or is spread across multiple locations, it may become harder for the system to keep working well on a larger scale. If the system can't scale well, it might not be able to be deployed as widely, which will cost more and make things harder.

Environmental Conditions: How well the system can find things can be affected by changes in the environment, such as how much light there is, the weather, and how the site is set up. When the model is used in different environments, it may not find things as well, which could make workers less safe and make the system less reliable.

8. Compliance and Ethical Considerations

8.1. Data Privacy and Security Measures

To address data privacy and security in our project, we will implement simplified measures that are feasible for our level:

- **Basic Data Handling:** We'll ensure that all data used, especially video feeds, is handled responsibly, with access limited only to WorkerSecure users. (The safety experts employed by the company using the WorkerSecure.)
- **Ethical Considerations:** WorkerSecure is designed with global expansion in mind, not limiting itself to the laws of any single country or region. It guarantees that the videos processed, and data stored will not be shared with any third parties. Live video feeds are used solely to determine compliance with safety regulations, with no other purpose. Before using WorkerSecure, the purpose and operation of the system are clearly explained to client companies and their employees. Any additional precautions or restrictions requested by client companies are considered by project owners for potential inclusion.
- **Data Minimization:** WorkerSecure only collects and processes the data necessary for the project's objectives to reduce privacy risks and some reclaims. For example, WorkerSecure doesn't collect the workers Identification number.
- **Transparency with Data Usage:** Workers will be informed how and why their data is used within WorkerSecure project. The aim of usage will be clearly explained.

8.2. Adherence to Ethical Standards (e.g., ACM Code of Ethics)

In aligning the WorkerSecure project with ethical standards, the ACM Code of Ethics provides a foundational guide. Key principles include:

• **Professional Responsibility:** Professional Responsibility states that a project should be in the public interest and have a positive impact on the professional environment. In this context, WorkerSecure aims to create a safer space, increasing the safety of both employees and the company. So, companies and their workers will be more positive in their work life thanks to WorkerSecure.

- **Avoid Harm:** This principle, emphasizing the prevention of harm through technology use, is essentially the main goal of the WorkerSecure project. The WorkerSecure system actively works to prevent potential harm to workers by detecting the absence of safety helmets and issuing warnings, utilizing technology to enhance worker safety and health.

- **Privacy and Confidentiality:** This principle emphasizes ensuring the privacy of employees and users. In that manner, WorkerSecure guarantees that the videos processed, and data stored will not be shared with any third parties. Live video feeds are used solely to determine compliance with safety regulations, with no other purpose.

- **Honesty and Fairness:** Honesty in disclosing the capabilities and limitations of technology, along with fairness in its application, is essential. WorkerSecure maintains transparency about its function — to improve safety on construction sites — and is designed to be fair in its application, without discriminating against any worker. It provides unbiased monitoring to enhance workplace safety for everyone equally.

9. Glossary

- **Adaptive Processing:** To maintain efficiency, adjusting the computational resources automatically.
- **Data Augmentation:** To create more diverse dataset, applying various transformations like scaling, rotation, color changing.
- **Model Pruning:** Removing irrelevant parameters from a neural network to improve efficiency.
- **Neural Network**: An NN is a model designed to process data, like the logic of human brain. It can learn from the data or interpret it.
- **Real-time Monitoring:** The system's capability to provide immediate analysis.
- **YOLO**: You Only Look Once. Real-time object detection system. It allows for accurate identification of objects in video frames.

10. References

- **Digital Image Processing Algorithms and Applications, by Ioannis Pitas, 2000, https://books.google.com.tr/books?hl=tr&lr=&id=VQs_Ly4DYDMC&oi=fnd&pg=PA1&dq=image+processing+algorithms&ots=jKlciHwkYo&sig=QV7q1n5LIBbhoMQ_szklmc_lZPw&redir_esc=y#v=onepage&q=image%20processing%20algorithms&f=false**

- **ACM Code of Ethics and Professional Conduct**

- **Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.**

- Low Level Design Document, Department of Computer Science, Western Michigan University, 2.0, https://nest.cs.wmich.edu/report/AGORA-Low-Level-Design-Document-rev2.pdf