TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 492

WorkerSecure

Test Plan Report

by

Hasret Türkmen

Asaf Emin Gündüz

Selin Sevcan Çakan

Niyazi Ahmet Metin

TABLE OF CONTENTS

1. Introduction

1.1. Purpose of the Test Plan

The purpose of WorkerSecure test plan is to create a strategy for validating our system. WorkerSecure is a safety-equipment detection project in working areas such as factories where work-related injuries are common. It utilizes YOLO model for accurate and efficient image processing.

This document addresses the test environment setup, risk management, the test approach, and the test cases.

1.2. Scope of Testing

The scope of WorkerSecure project includes the testing of the YOLO model, the code we used to train the data, the accuracy of the trained data.

Furthermore, the Model validation tests will ensure the responsiveness of the model and its accuracy on detecting helmets. Each test case within these areas will be labeled with a unique test ID for precise tracking and reporting purposes.

2. Test Items

The key features and system components are tested within a schedule to ensure well-build structure for WorkerSecure. The test items are crucial to meet a reliable and functional system with high performance.

- Helmet Detection,
- Real Time Alert System,
- System Adaptability to Different Environmental Conditions,
- Functionality of User Controls,
- User Interface Responsiveness and Usability,
- Data Handling and Security.

3. Features To Be Tested

The following section outlines the functionalities of WorkerSecure project that are critical and to be tested. Each feature has its own description and an explanation about its importance. Testing these features will help to show if meets the required safety standards and operational efficiencies.

1. **Helmet Detection Accuracy**
   - Description: While testing this feature, simply we want to test that with what accuracy the model identifies if somebody wears a hard-hat or not in real-time through video feeds.
   - Importance: The crucial part in our WorkerSecure project is this feature, the aim of this feature is to detect if workers comply with ISG rules or not with minimizing false negatives (failing to detect a helmet when it is worn) and false positives (detecting a helmet when it is not worn).

2. **Real-Time Alert System**
   - Description: The ability of alerting OH&S specialist when a worker is identified as not wearing hard-hat.
   - Importance: The faster alert time, safer environment for workers. If the program alerts OH&S specialist faster, workers will wear their hard-hat faster.
3. **System Adaptability to Different Environmental Conditions**
   - Description: There might be different light and weather conditions. This feature tests model's accuracy in adverse conditions.
   - Importance: Some workplaces / some factories could be always in adverse weather / light conditions. This feature shows reliability and effectiveness of WorkerSecure.
4. **Functionality of User Controls**
   - Description: Verifying that all user interface controls (buttons, sliders, alerts, etc.) perform as expected, facilitating user interaction without issues.
   - Importance: While occupational health and safety specialist using the program, they could do what they want to do in UI rapidly and correctly. Because, if it isn't working correct, the OH&S specialist can't warn the worker, and it impacts safety management.
5. **User Interface Responsiveness and Usability**
   - Description: Tests if the User interface responsive and useable for occupational health and safety specialists.
   - Importance: Ensures that safety officers or system administrators can effectively monitor and interact with the system without delays or usability issues, which could impact safety management.
6. **Data Handling and Security**
   - Description: Verifying that all data captured and processed by the system is handled securely, including access controls to prevent unauthorized access.
   - Importance: Ensures compliance with privacy laws and regulations, maintaining the integrity and confidentiality of potentially sensitive workplace surveillance data.

4. Test Approach

4.1. Model Validation Testing

Model Validation Testing is an important step in ensuring that the hardhat detection system performs accurately and reliably under each operational condition. Here's how we plan to do this testing:

Our main objective is to make our model correctly identify whether workers are wearing their safety equipment. We should maintain high accuracy across every environment and weather condition.

Data Preparation: We make sure that dataset includes many different conditions such as weather conditions and different sets of light. Split the dataset into training (70%), validation (20%), and testing (10%).

Performance Evaluation: When training process comes to an end, YOLO provides us some of the performance metrics for the trained model, such as mAP(96.9%), Precision(95.1%), Recall(93.3%). From the observations of these metrics and analyzing them we concluded that our model is more than enough for the specified tasks.

Environmental Testing: After training is completed, we tested the model in different kinds of environments representing different lighting, weather, and background scenarios to ensure it can operate effectively in any condition encountered on a construction site. We done this using our laptop webcam, thus when we have RTSP stream that is working properly we will once again to this test.

Real-Time Testing: By using our webcams, we evaluate the model's performance in real-time scenarios to ensure it can process and analyze images quickly and accurately, without any vital delay.

1. We will use tools like Selenium for consistent and efficient test execution when we complete developing our UI.
2. We are planning to develop scripts to automate the collection and analysis of test results, for faster improvements.

These methodologies ensure comprehensive validation of the model across all critical aspects, from its technical accuracy to its operational reliability in realistic conditions.

5. Test Environment

5.1. Hardware-Software Requirements

Hardware Requirements:

Testing will utilize NVIDIA GTX 1050TI GPUs with 4 GB of memory, which have been proven to handle the project’s image processing demands effectively. The CPU requirements include an Intel i7-7700HQ, which provides robust performance for managing large datasets.

Software Requirements:

The software environment for testing includes PyTorch 1.8.1 as the deep learning framework, chosen for its dynamic computation graph capabilities. The YOLOv7 implementation from Ultralytics will be used for object detection tasks, leveraging the CUDA Toolkit 11.2 for GPU acceleration.

5.2. Setup of the Testing Environment

Initial tests for the Model Validation Test are done using the laptop camera which we used for running the trained model. We checked whether the model could understand if sees a head or a helmet. We realized that even though the model successfully recognizes head of a human, it is likely to confuse helmets with other round objects. This is due to the round objects having similar colors to a helmet, such as yellow or white, and our model being not completely trained using the dataset we have.

The tests will continue as we train our model with more data.

6. Risks and Mitigations

There could be some various reasons that makes using the WorkerSecure project risky. Thus, it's beneficial for developers to think critically and consider risk management. To maintain the reliability and the usability of the project, developers should preestimate the risks and take some actions about possible solutions of them. In this section of the plan report, the potential risks, and possible solutions to manage them is discussed.

Risk-1 Inaccurate Helmet Detection: If the algorithm is affected negatively by different variation of helmets, specifically it's design, the accuracy may decrease. Also, accuracy may depend on the environmental conditions, specifically the lightning.

•How to Reduce This Risk: A rich dataset that consists of pictures of different variations of helmets, and different environmental conditions, specifically the lightning would reduce that risk.

Risk-2 Inconsistencies in Helmet Detections: Different camera angles may have a negative effect on the WorkerSecure's helmet detection.

•How to Reduce This Risk: Various test conditions with different camera angles to make sure the detection algorithms are efficient under each condition.

Risk-3 Delayed Alerts: The speed of network connection may influence the timing of alarms and notifications sent both to workers and system user.

•How to Reduce This Risk: Testing the alert system across different network speeds to optimize the alarm system.

Risk-4 Failure to Handle Concurrent Alerts: In a big working area that consists of more than one camera angle, the whole program shouldn't be alarmed for every helmet. (If somebody doesn't wear hard-hat on region1, the program shouldn't warn the people on all regions, program should warn only the workers on region1.)

•How to Reduce This Risk: In implementation, giving unique region IDs to the regions, and warning them using that ID would be possible solution to that problem. If two violence of laws detected in different areas, the alarm goes off according to these ID numbers of the regions.

Risk-5 Non-Responsive User Interface Control Buttons: The front-end code may lead to slow response times and unresponsive control buttons if it is implemented inefficiently.

•How to Reduce This Risk: Optimization of the front-end code to ensure control buttons functions effectively.

Risk-6 Breach of Data: Weaknesses in how data is sent and stored can make workers' information accessible to unauthorized parties.

•How to Reduce This Risk: Regular updates of security protocol done by authorized staff to identify the weaknesses and prevent them.

7. Test Cases

Testcase ID	Description	Test Steps	Input	Expected Output	Pass/fail criteria
TC-HDA-01	Test helmet detection accuracy in controlled environment.	1) Play pre-recorded video of workers, both with and without helmets. 2) Observe the detection response.	Video feed with clear visibility and various workers wearing helmets.	Accurate identification of helmet status for each worker.	Pass if all helmets are correctly identified with no false positives or negatives.
TC-HDA-02	Validate helmet detection under varied lighting conditions.	1) Play videos recorded during different times (day and night). 2) Analyze the accuracy of helmet detection.	Videos with varying light conditions.	Consistent helmet detection across all videos.	Pass if helmets are detected accurately in all lighting conditions, fail otherwise.
TC-RAS-01	Evaluate the alert system's response time.	1) Trigger a helmet detection event. 2) Measure the time taken for the alert to reach the OH&S specialist.	Video of a worker suddenly removing the helmet.	Immediate alert notification.	Pass if alert is generated within a predefined time limit, fail if delayed.
TC-SAD-01	Test model performance in adverse weather conditions.	1) Use video feeds recorded during adverse weather conditions. 2) Assess the detection success rate.	Videos captured in rain, fog, or heavy wind.	Accurate helmet detection despite adverse conditions.	Pass if system maintains high accuracy, fail if detection rates drop.
TC-FUC-01	Check the functionality of all interactive UI controls.	1) Engage each UI control including buttons, sliders, and alerts. 2) Note any performance or responsiveness issues.	Commands triggered via UI controls.	Immediate and correct response to each command.	Pass if all controls work as intended, fail if any do not function properly.
TC-UIR-01	Assess UI responsiveness under peak load.	1) Simulate maximum number of concurrent users. 2) Perform standard operations within the UI.	High frequency and variety of user interactions.	No delays or crashes in UI performance.	Pass if UI remains responsive, fail if it becomes sluggish or unresponsive.
TC-DHS-01	Verify data encryption and access controls during data transmission.	1) Simulate data transmission between the site cameras and the central server. 2) Check for encryption and unauthorized access attempts.	Encrypted data packets being transmitted.	Data remains secure and inaccessible to unauthorized users.	Pass if data is transmitted securely without any breaches, fail if encryption fails or data is accessed by unauthorized entities.
TC-DHS-02	Test the security of stored data against unauthorized access.	1) Attempt to access stored data using various unauthorized methods. 2) Check security logs for any breach alerts.	Unauthorized access attempts.	Access denied and alerts triggered.	Pass if all unauthorized attempts are blocked and alerted, fail if a breach occurs.

8. References

1- Gerald D. Everett; Raymond McLeod, "Test Planning," in Software Testing: Testing Across the Entire Software Development Life Cycle, IEEE, 2007, pp.79-92, doi: 10.1002/9780470146354.ch5.

2- Hitch, L. P., & Hirsch, D. (2001). Model training. *The Journal of Academic Librarianship*, *27*(1), 15-19, doi: 10.1016/S0099-1333(00)00203-2