WebSocket-Based Smart Surveillance Camera for Real-Time Detection of Occupational Health and Safety PPE Non-Compliance in Industrial Areas

Rivaldi Azis Sabarto
Electrical Engineering
Faculty of Science and Technology
Universitas Muhammmadiyah Sidoaarjo
Sidoarjo, Indonesia
rivaldiazis5@gmail.com

Indah Sulistiyowati
Electrical Engineering
Faculty of Science and Technology
Universitas Muhammadiyah Sidoarjo
Sidoarjo, Indonesia
indah_sulistiyowati@umsida.ac.id

Syamsudduha Syahrorini
Electrical Engineering
Faculty of Science and Technology
Universitas Muhammadiyah Sidoarjo
Sidoarjo, Indonesia
syahrorini@umsida.ac.id

Arief Wisaksono
Electrical Engineering
Faculty of Science and Technology
Universitas Muhammadiyah Sidoarjo
Sidoarjo, Indonesia
ariefwisaksono@umsida.ac.id

Abstract— In industrial settings, ensuring adherence to Occupational Health and Safety (OHS) Personal Protective Equipment (PPE) regulations continues to be a crucial challenge. The creation of a WebSocket-based smart surveillance camera system for the real-time identification and reduction of PPE infractions is discussed in the paper. The proposed system includes an ESP32-S3 microcontroller accompanied by an OV5640 camera module, acting as an edge-processing embedded platform. The Edge Impulse machine learning framework was used to train image classification and detection models, enabling efficient low-latency inference directly on the device. A websocket enabled web server streams video frames in real time for constant monitoring, with instant display using regular browsers without wasting bandwidth. Experimental results demonstrate that even with limited computational resources, the system is able to perform on-device inference with very high responsiveness and good detection accuracy. This technology provides a scalable and affordable way to enhance OHS compliance monitoring in industry, reduce reliance on manual supervision, and encourage proactive risk mitigation methodologies.

Keywords— ESP32-S3, OV5640, Edge Impulse, IoT, PPE, Webserver

I. INTRODUCTION

As occupational health and safety (OHS) violations in industrial environments have increased, security has become an increasingly important consideration. Various measures have been taken to protect premises, such as placing locking mechanisms on doors or deploying surveillance cameras (CCTV)[1]. However, these procedures are still seen as ineffectual in preventing workplace health and safety infractions because they frequently rely on manual supervision and retrospective examinations of recorded footage rather than proactive, real-time intervention[2].

With technical advancements, notably in the Internet of Things (IoT), internet use has become more widespread, including in the sphere of surveillance. One application is the usage of remote automated control systems built on devices like the ESP32-S3 and linked to a websocket platform[3]. This integration produces a cohesive system where sensors, cameras, and actuators communicate smoothly over a network, allowing for smarter, more responsive security and safety solutions[4].

The industry has the potential to improve security by the use of video surveillance systems. Modern video surveillance systems involve the use of a digital camera connected to a computer (PC) to enable data storage and monitoring by means of a WebSocket server [5]. The WebSocket server aids the bidirectional communication between the client and the server through the establishment of a single, persistent connection after an initial handshake with the client[6]. As opposed to the traditional web servers that make use of repetitive HTTP requests, the WebSocket approach allows the use of an open, constant connection that supports the instant bidirectional message transfer between the client and the server, including the transmission of lightweight messages or binary messages that do not need reloads of the page[7].

The ESP32-S3 has dual-core processing, Wi-Fi/Bluetooth connectivity, and on-chip AI acceleration, making it perfect for real-time image processing. Using AI models trained to detect PPE (safety helmet, and safety garment.), it can locally evaluate video feeds and provide alerts when violations occur, reducing data transfer and improving response time[8]. Modern surveillance systems improve security and workplace safety by giving real-time notifications for violations such as missing personal protective equipment, restricted area entry, or unauthorized machine use[9]. By connecting these technologies to cloud databases and dashboards, data may be stored for audits, trend analysis, and predictive risk detection, allowing preventive measures to be implemented before accidents occur[10]. Overall this represents a move from passive monitoring to proactive, automated safety enforcement, which benefits asset security, worker safety, and regulatory compliance[11].

This ESP32-S3-based PPE detection system brings out a novel way of safety surveillance in the industrial setting through the fusion of edge AI and real-world implementable factors[12]. Unlike typical PPE detection systems, which rely on costly cloud servers or only focus on the simple detection of PPE without any context-related interpretation of the results, the proposed work performs the entire inference on low-cost ESP32-S3 platforms and provides valuable feedback on the safety aspect as well. One of the novel aspects of the system is the real-time visual differentiation based on safety compliance of the workers into two categories: one wearing PPE like vests and helmets, denoted as green color-coded bounding boxes, and the other violating safety by the absence of proper safety gear, denoted as red color-coded bounding boxes[13]. The novelty of the system also lies in the optimization for mobile hotspot connectivity, and Wi-Fi connection at the site of deployment, preferably at construction areas, storage units, or any temporary working setup[14].

The integrated web dashboard provides live camera feeds at QVGA resolution (320x240) with overlaid statistical data, including compliance figures, inference latency, and violation notices, accessible on any computer connected to the same network[15]. Moreover, the system achieves fast inference times and upholds user privacy due to Edge Impulse's model executing the entire processing on the same device. The interplay of Edge AI, visualized compliance, mobility, and ease of development enables a functional safety monitoring system that can be implemented at significantly lower cost than a standard system, hence increasing workplace safety technology availability for smaller businesses and transient working sites[16].

The paper proposes a real-time monitoring and alert control system that is achieved using an ESP32 security camera system setup that has an integrated web server and an alert sound (buzzer) component. The project uses an ESP32-S3 and a suitable camera module (such as OV5640) to display the monitoring in a local website via Wi-Fi connectivity for real-time monitoring and alert triggering. Users are able to monitor and control outputs such as servos and buzzers using a common web browser interface. The project also provides self-surveillance capabilities that do not require human interaction to generate alerts through motion sensing and alert activation mechanisms that enable autonomous surveillance and alert triggering without human intervention and without the need to install applications like Blynk for alert triggering

II. METHODOLOGY

A. Machine Learning

Machine Learning (ML) is a powerful AI branch that may benefit businesses of all sizes and sectors. It is a solution for replacing resources with high operational expenses. The study aimed to evaluate the Edge Impulse platform as a machine learning tool[17]. The system uses low-code Frameworks abstract complicated ML approaches, including data processing and AI component structure. It indicates a reduction in time during the development process[18]. Edge Impulse provides a user-friendly interface with clear logic flow. The study focused on an application for object recognition, with the goal of reaching the system's potential. The system reported an autogenerated accuracy value of 97.9% after training and 89% after retesting the initial 20 mice photographed from various angles, indicating potential overfitting[19]. Although the system showed potential in classifying items. Adjustments to the image collection can enhance the model's recognition capacity, as the survey concluded with insufficient images[20].

B. Tensorflow Lite Micro

TensorFlow Lite for Microcontrollers (TFLM) is a software version of TensorFlow Lite built primarily for running machine learning models on DSPs, microcontrollers, and other memory-constrained devices[21].

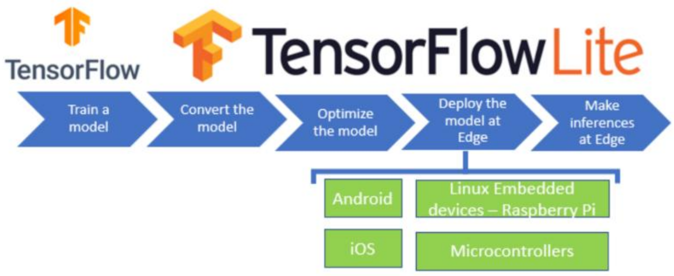

Figure 1. Implementation of Diagram TFLM

TFLM operates in four steps. First, TFLM reads the Resolver Operator specified in the application. The Resolver Operator manages the TensorFlow API that will be utilized in the microcontroller, minimizing its footprint[22]. The second stage involves TFLM reading the existing memory size. In the application implementation, the allotted memory space is commonly referred to as the "Arena" or "TensorArenaSize". This memory space will be used to store inputs, outputs, intermediate arrays, and other variables required by the TFLM interpreter. Developing a TFLM interpreter that takes the model, Resolver Operator, and TensorArenaSize is the third step[23]. After that, upon initialization, the interpreter will allot all necessary RAM. The interpreter will provide the application with a memory pointer that represents the model in the fourth stage. After then, input data will be added to this pointer. Following the completion of the input, the interpreter will execute the invocation process, calculate the model, and generate output[24].

C. Experiments

The research process involved numerous stages, including literature review, needs analysis, hardware and software design, dataset collecting, training model construction, and accuracy testing and validation.

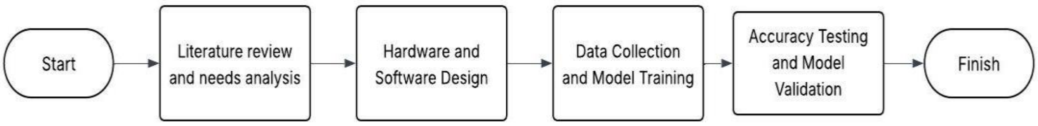

Figure 2. Research Method Stages

1) Literature Review and Needs Analysis

This is the first step of research, in which the researcher does a literature review or research on prior studies to gather supporting data and theories. By accumulating varied facts and supporting hypotheses from prior studies, the researcher can conduct a requirements analysis for the study[25]. The following are the requirements for this research

1) An AI (artificial intelligence) development platform. In this case, the platform employed is Edge Impulse Studio, which is available online at edgeimpulse.com.
2) The ESP32 S3 microcontroller and OV5640 camera serve as hardware for capturing images of Personal Protective Equipment to be classified.
3) Arduino IDE is a microcontroller development software for programming the ESP32 S3 and OV5640 Camera.
4) A laptop with 8GB RAM specifications and Windows 11 operating system.

2) Hardware and Software Design

According to the requirements analysis, the necessary hardware is an ESP32 S3 WROOM, OV5640 camera and a laptop. The design is simple link the laptop and the ESP32 S3 via a USB cable, as illustrated in Figure 3

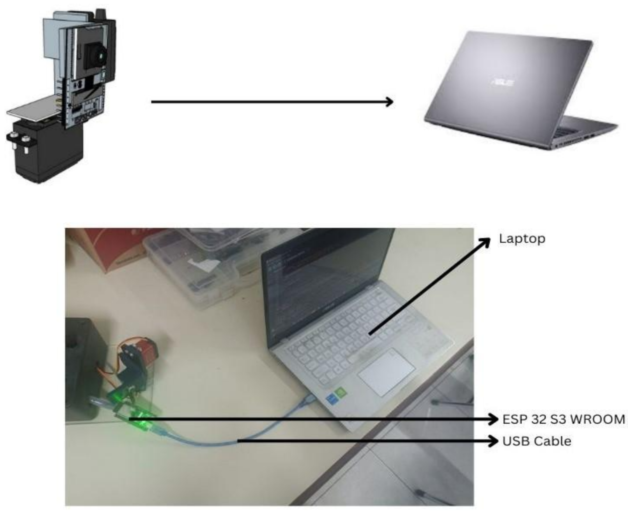

Figure 3 Hardware Design ESP32 S3 and Laptop

The software used is Edge Impulse Studio as an AI development platform and the Arduino IDE to program the ESP32-S3 and OV5640 Camera. Below is a screenshot of Edge Impulse Studio and the Arduino IDE.

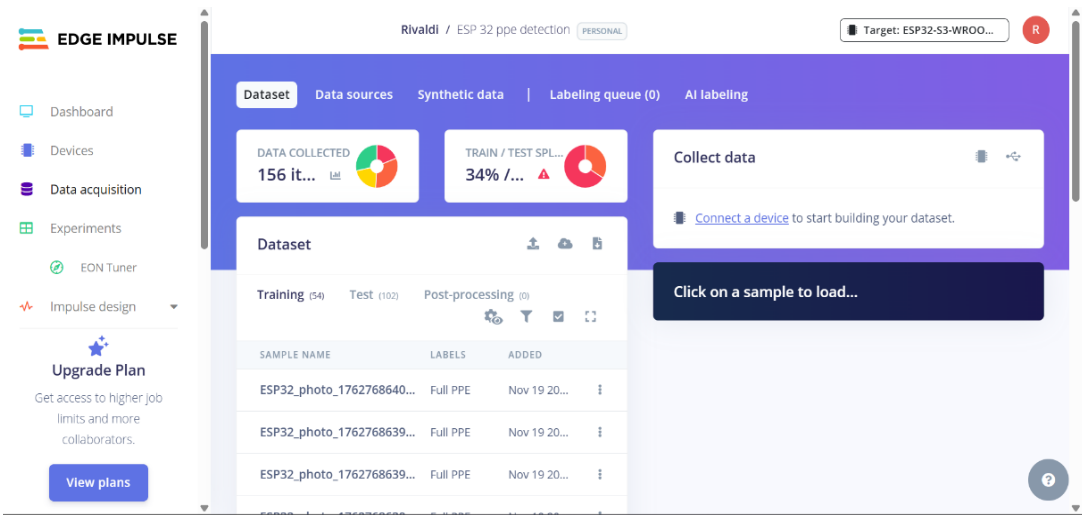

Figure 4 Edge Impulse Dashboard Display

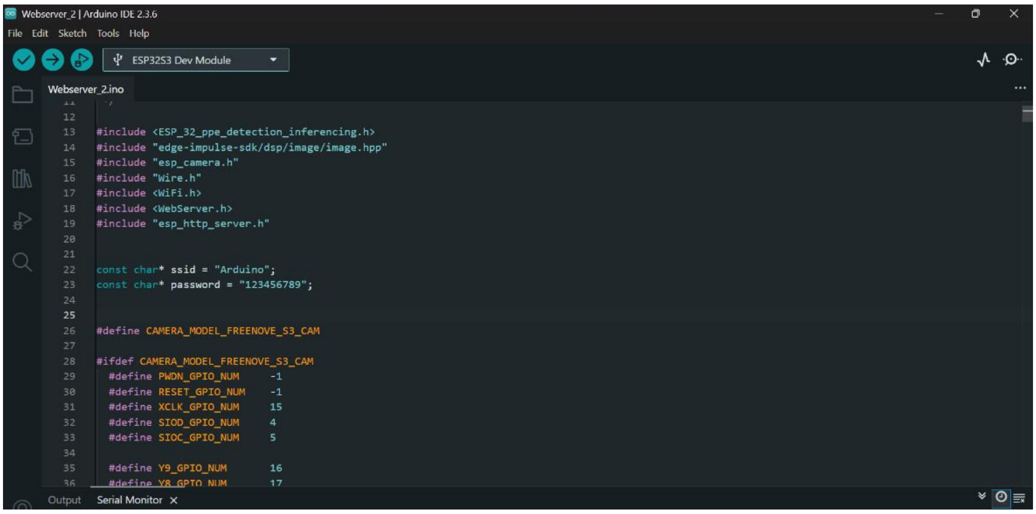

Figure 5 Arduino IDE Display

3) Data Collection and Model Training
   a) Data Collection

At this stage, the collected data is an image dataset, or images that will be used as a training model for object classification. In this case, the images collected will be types of safety wear. The sample image collection was obtained using an OV5640 camera programmed through the Arduino IDE. In this study, the researchers will use four types of objects to classify: Without PPE, Safety Garment, Helmet, Safety Garment and Safety Helmet. The following is the number of sample data used for model training:

Table 1 Total dataset sample for training model dan testing

Object	Train	Test
Without PPE	28	16
Safety Helmet	30	17
Safety Garment	20	11
Safety Helmet and Safety Garment	20	10
Data Collected		152

Table 1 explains the distribution of the dataset used to train and test the model. The dataset includes images that are divided into four classes according to the usage of PPE, which are workers not wearing PPE, workers wearing safety helmets, workers wearing safety garments, and those wearing both safety helmets and safety garments. A total of 98 training samples are obtained by collecting 28 images of workers not wearing PPE, 30 images of workers wearing safety helmets, 20 images of workers wearing safety garments, and 20 images of workers wearing both safety helmets and safety garments during the training process. For the testing process, the dataset includes 16 images of workers not wearing PPE, 17 images of workers wearing safety helmets, 11 images of workers wearing safety garments, and 10 images of workers wearing both safety helmets and safety garments, which make up 54 test samples. A total of 152 images are gathered to form the overall dataset used to train and test the model to check the usage of PPE at the industrial level.

The Edge Impulse Studio system automatically determines the amount of sample datasets when tagging submitted photos.

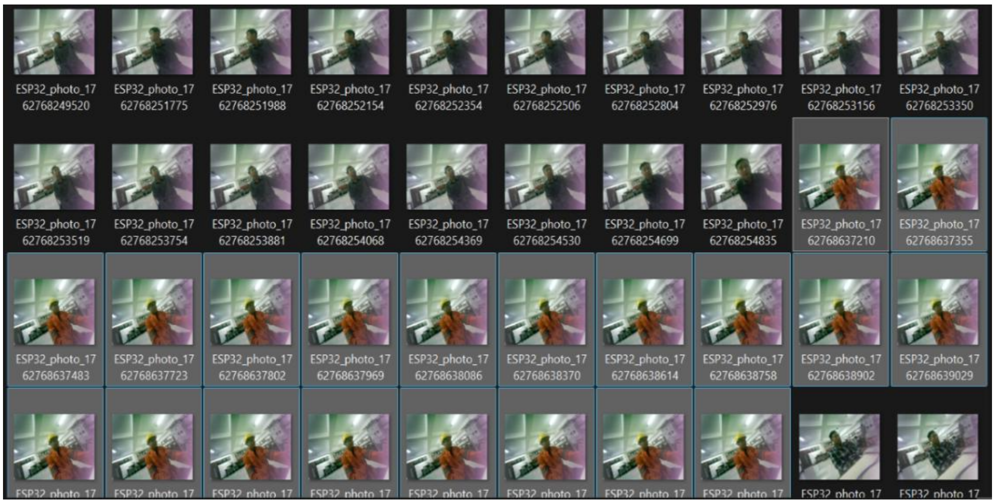

Figure 6 Sample Data

b) Model Training

After uploading the sample data to Edge Impulse Studio, the next step is to label each sample image individually so that they may be classified into the model to be trained. The data is then equalized or divided between the training and testing samples. This is accomplished by splitting the amount of training and testing samples into around 80/20 proportions. This was done to increase the accuracy of model training.

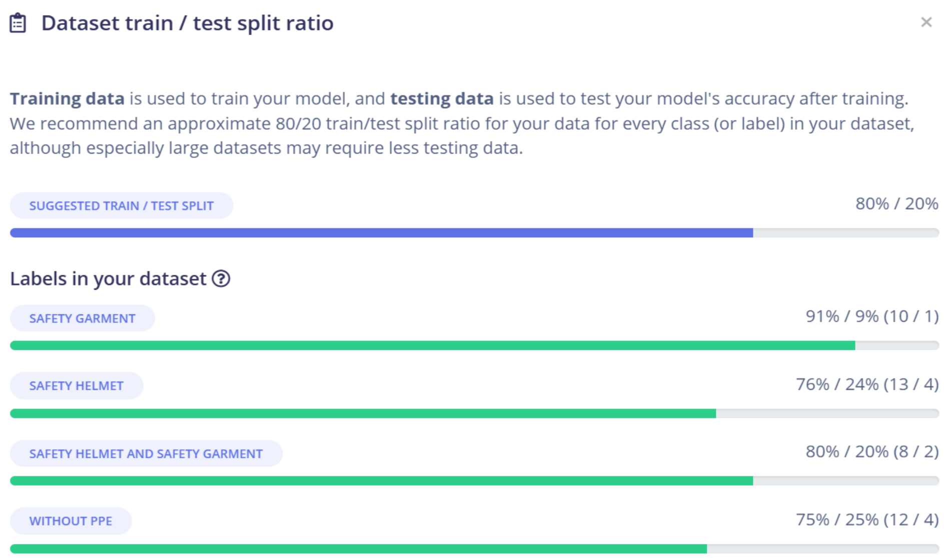

Figure 7 The collected data samples were labeled and split in a ratio of approximately

c) Accuracy and Validation Testing

During the model training process, Edge Impulse Studio uses a Confusion Matrix to calculate accuracy for each model training result. This matrix will compare each categorized object to other objects and its backdrop. Lower error equals more accuracy, and vice versa. The F1 score, or ultimate accuracy value, is calculated by averaging the accuracy values for each object. As a result, the higher the F1 score, the more accurate the training model. The F1 scoring formula can be found in the equation below.

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 8 Formula for calculating F1 Score

Edge Impulse Studio's Live Classification capability is used during the classification testing procedure. To accomplish this, take these steps:

1) Connecting the ESP32-S3 WROOM and OV5640 camera to the laptop.
2) Register the ESP32-S3 WROOM on the device list in Edge Impuse Studio.
3) Go to the Live Classification menu to start capturing images of objects and classifying them.
4) Prepare trained safety clothing. The object can be an image from the validation/test set.
5) Taking pictures and classifying objects.
6) If the classification process is successful, the image will be labeled and saved to the dataset.
7) If the image classification process fails, replace the object by using a different dataset or by improving the image capture quality.
8) If you have replaced the object with another dataset, repeat step 5.
9) The classification process was successfully carried out.

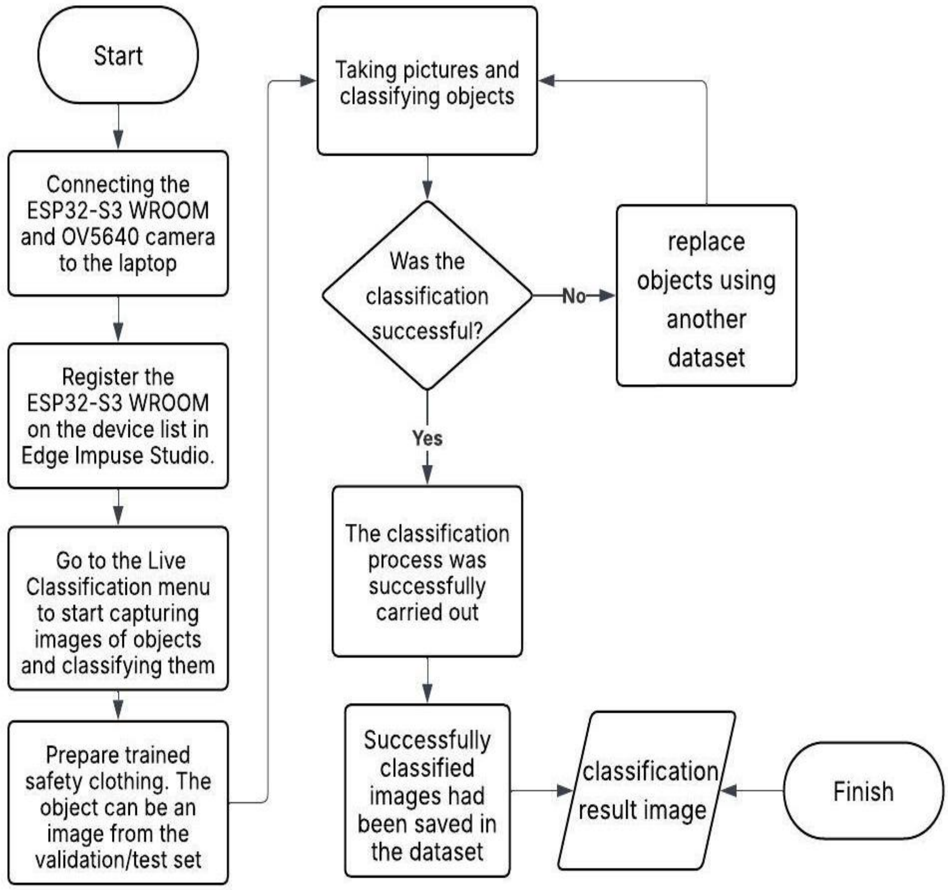

Figure 9 Flowchart of how object classification works using ESP32-S3 and OV5640 camera

III. Results and Discussion

Based on the results of the implementation of the research method, several findings from the research conducted can be discussed

A. Model Training Result

Edge Impulse Studio provides accurate results for each trained object instantly after model training. The training results are displayed as a confusion matrix. The researchers achieved the following test findings after completing the model training.

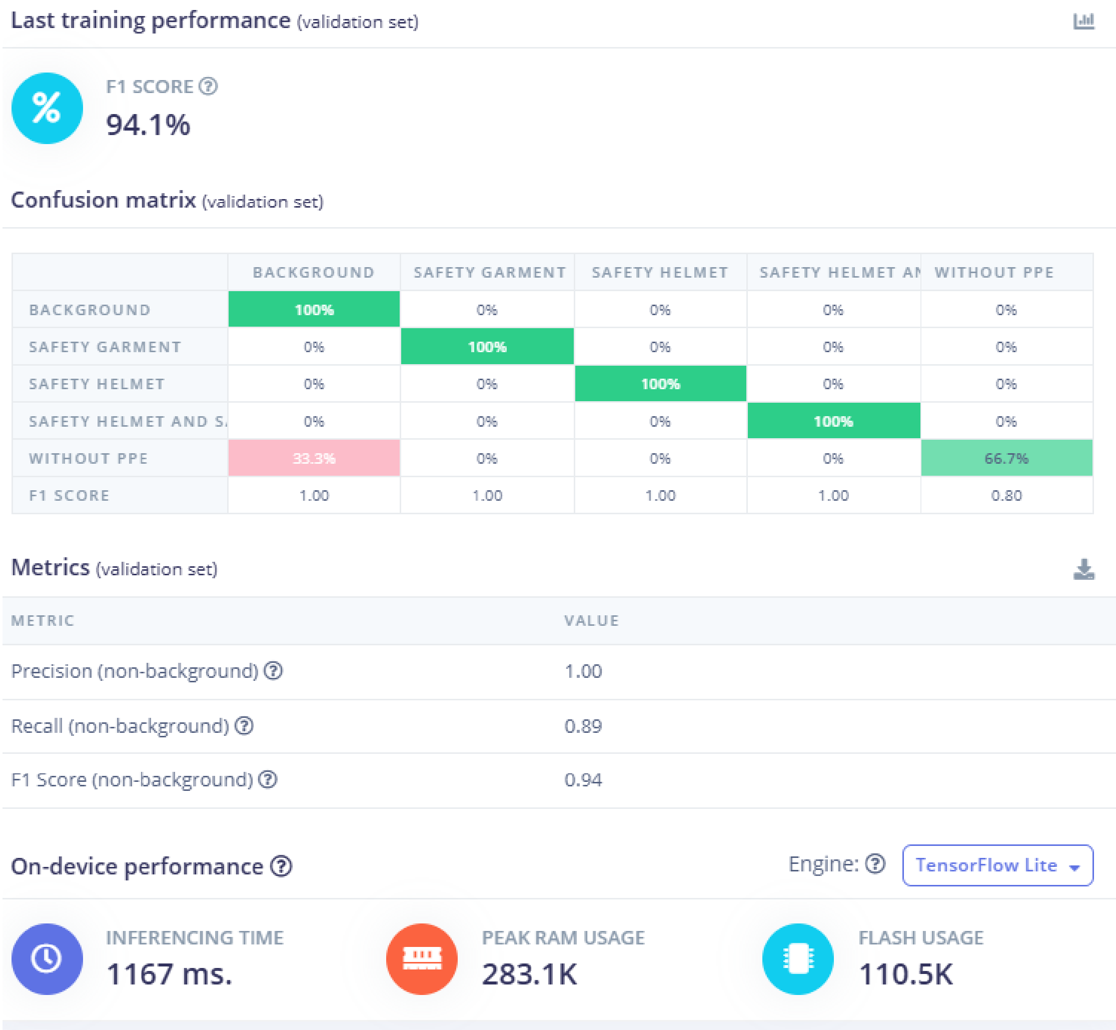

Figure 10 Model Training Results as a Confusion Matrix

This matrix yielded the following results:

1) Safety Helmet and Safety Garment with 100% accuracy
2) Safety Helmet with 100% accuracy
3) Safety Garment with 100% accuracy
4) Without PPe with 66.7% accruracy
5) F1 score with 63.2% accuracy

Several points may be made about the model training results above, including the high accuracy of the Safety Helmet and Safety Garment, Safety Helmet, and Safety Garment models. However, the Without PPe model had a lower-than-average F1 score. This was due to the uploaded sample data being less accurately categorized. Another factor was the limited RAM utilization caused by the target hardware, the ESP32-S3, which only has 512KB of RAM. As a result, the system compacted the learned model such that it could be deployed to the ESP32-S3.

B. Live Classification Demonstration

The final model can be deployed to the device that will perform classification. In this case, the device is an ESP32-S3 . To deploy the model, an Arduino library must be generated and uploaded to the ESP32-S3. To deploy the model, navigate to the Deployment menu in Edge Impulse Studio and pick the device to utilize. Edge Impulse will then generate the Arduino library, which will be available for download once complete. The library can then be loaded into the Arduino IDE and uploaded to the ESP32-S3 . After successfully deploying the application to the ESP32-S3, the next step is to run an object classification demonstration experiment with the ESP32-S3 and the OV5640 CAM, utilizing photos of workwear from the testing dataset, as described in Chapter 3. The following is a demonstration of utilizing the ESP32-S3 to capture photographs of work safety apparel. In this test, the ESP32-CAM and OV5640 CAM correctly identified items in test dataset photos displayed on a computer monitor. The test results can be found in Table 2.

Table 2 System Test Results

Object	Classification Result	Accuracy	Number of Trials
Without PPE	Without PPE (Success)	60.5 %	1 time (immediately detected)
Safety Helmet	Safety Helmet (Success)	76,2 %	1 time (immediately detected)
Safety Garment	Safety Garment (Success)	81,6%	1 time (immediately detected)
Safety Garment and Safety Helmet	Safety Garment and Safety Helmet (Success)	84,0%	1 time (immediately detected)

The following image shows the results of an image capture experiment using the ESP32-S3 WROOM and OV5640 CAM, along with the classification results.

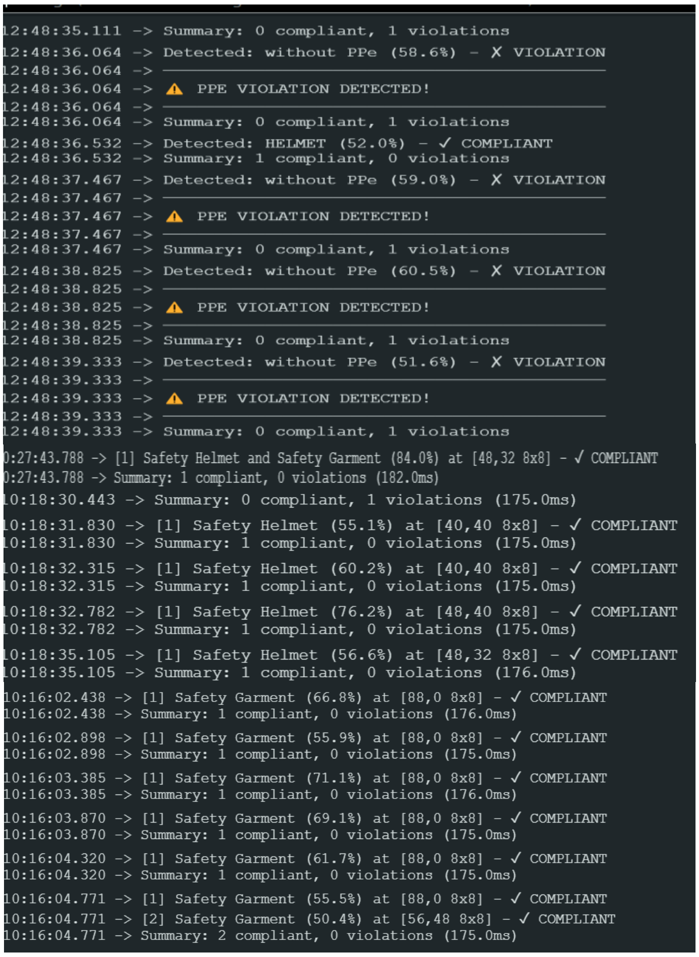

Figure 11 Display of object classification results using Arduino IDE Serial Monitor

Based on the testing and demonstrations above, the researcher might draw various conclusions about the relevance of this research to earlier research, namely:

1) The accuracy gained in this investigation was lower than in earlier studies. This is most likely due to discrepancies in sample datasets and their processing prior to training.
2) Although the accuracy is not exceptionally great, this study demonstrates that the ESP32-S3 and OV5640 CAM can be utilized as object detection/classification devices despite their relatively low specifications (512 kb of RAM).
3) The FOMO CNN architecture described by Edge Impulse Studio shows a visibly low resource consumption, which makes deploying and training a low-end microcontroller a straightforward process.
4) The Edge Impulse Studio AI platform has tools and libraries that ensure a simple development process for AI/ML projects.

C. Demonstration Via Web Server

The system demonstration made use of the ESP32-S3's capabilities to host a web server, which served as the primary interface for showing live camera footage and Edge Impulse AI model inference results. The web server was accessed via a local Wi-Fi network by entering the IP address displayed by the module after setup

Figure 12 Webserver Dashboard Display

The resulting web interface is designed to be simple and responsive on both mobile and desktop devices. The homepage displays two main components.

1) Live video feed
2) Real-time AI detection results

The system automatically highlights identified regions with colored bounding boxes and reports the violation in the online log. the demonstration procedure demonstrated that the webserver was able to function reliably at an average frame rate of 8–15 FPS, depending on the image resolution and AI model size. This speed is nevertheless suitable for occupational safety monitoring, as PPE infractions often occur within a timeframe detectable by the system.

D. Discussion

If implemented in industry with the resolution of QVGA, or 320 x 240 pixels, the detection range can only reach 5-10 meters, stressing the need for multi-camera systems in detecting wide areas. Also, the detection accuracy can decrease by 20-40% in situations where the lighting conditions are poor, there's high glare, or in outdoor environments, and the use of infrared modules for 24/7 operations or hybrid automated manual methods should be considered for increased accuracy. Moreover, the system can degrade with the entry of more than 3-4 employees in the image, contrary to the requirements of assembly lines where the density of employees can reach 40%.

To ensure the efficiency of surveillance, the cameras have to be placed at strategic points where control is critical. These points include entry/exit points, boundaries of hazard zones, or high-risk workstations where employees move across the zones. This is known as the checkpoint model, in which 100% surveillance is ensured at critical points despite the limited range of individual sensors. For this purpose, for large storage facilities or construction areas, the ESP32 module needs to be employed on the basis of zoning, with each module focusing on an 8x8 meter boundary to ensure overlap zones for seamless transition.

IV. CONCLUSION

The results of this study suggest that using the ESP32-S3 and OV5640 CAM for picture classification and PPE detection via a web server is an effective and straightforward technique. Customers may do categorization at a cheap cost and without the need for advanced programming skills thanks to the ESP32-S3 and OV5640 CAM, which are very simple, affordable, and easy to program devices. Consequently, this application has the potential to be developed and implemented in a number of industries, such as Robotics, Manufacturing, food processing and agriculture.

V. REFERENCES

[1] A. Badri, B. Boudreau-trudel, and A. Saâdeddine, "Occupational health and safety in the industry 4 . 0 era : A cause for major concern ?," *Saf. Sci.*, vol. 109, no. August 2017, pp. 403–411, 2018, doi: 10.1016/j.ssci.2018.06.012.

[2] S. Al-Azani, H. Luqman, M. Alfarraj, A. A. I. Sidig, A. H. Khan, and D. Al-Hamed, "Real-Time Monitoring of Personal Protective Equipment Compliance in Surveillance Cameras," *IEEE Access*, vol. 12, no. September, pp. 121882–121895, 2024, doi: 10.1109/ACCESS.2024.3451117.

[3] N. Mitrovi and M. Đor, "Implementation of WebSockets in ESP32 based IoT Systems," pp. 261–264, 2021.

[4] V. Hyginus *et al.*, *Integrating IoT sensors and machine learning for sustainable precision agroecology : enhancing crop resilience and resource efficiency through data - driven strategies , challenges , and future prospects*. Springer International Publishing, 2025. doi: 10.1007/s44279-025-00247-y.

[5] M. Cabanillas-carbonell, J. S. Rivera, and J. S. Muñoz, "Artificial intelligence in video surveillance systems for suspicious activity detection and incident response : A systematic literature review," vol. 19, no. 3, pp. 389–405, 2025.

[6] S. Sathya and M. S. Swetha, "Real-Time Chat Application with Web Socket for Network Efficiency," 2025.

[7] S. Automatisés, D. Priyasta, and R. Septiawan, "Enabling EV Roaming Through Cascading WebSockets in OCPP 1 . 6," vol. 56, no. 3, pp. 437–449, 2023.

[8] Y. S. Baraskar, P. Gawande, and P. V. Ambhore, "EXPLORING THE CAPABILITIES OF ESP32-S3 : AN ADVANCED IOT DEVELOPMENT BOARD FOR NEXT-GENERATION EDGE COMPUTING," no. 11, pp. 5219–5227, 2025.

[9] L. Zhang, W. Guo, and C. Lv, "Science in One Health Modern technologies and solutions to enhance surveillance and response systems for emerging zoonotic diseases," *Sci. One Heal.*, vol. 3, no. August 2023, p. 100061, 2024, doi: 10.1016/j.soh.2023.100061.

[10] N. Abekiri, A. Rachdy, M. Ajaamoum, B. Nassiri, L. Elmahni, and Y. Oubail, "Platform for hands-on remote labs based on the ESP32 and," *Sci. African*, vol. 19, p. e01502, 2023, doi:

10.1016/j.sciaf.2022.e01502.
[11] J. C. Palomares-salas, J. Maria, J. C. Palomares-salas, J. Maria, P. Remigio-carmona, and J. Jos, "ScienceDirect n n Characterization Characterization and and Performance Performance Evaluation Evaluation of of ESP32 ESP32 for for Synchronized Sensor Networks Real-time Synchronized Sensor Networks," vol. 00, no. 2022, 2024, doi: 10.1016/j.procs.2024.05.104.
[12] J. Paul *et al.*, "ScienceDirect Extraction of Measurement Device Information on an ESP32 Extraction of Measurement Device Information on an ESP32 Microcontroller : TinyML for Image Processing Microcontroller : TinyML for Image Processing," 2024, doi: 10.1016/j.procs.2024.09.670.
[13] M. Novak, P. Doležal, O. Budík, L. Ptáček, J. Geyer, and R. G. Blue, "Engineering Science and Technology , an International Journal Intelligent inspection probe for monitoring bark beetle activities using embedded IoT real-time object detection," *Eng. Sci. Technol. an Int. J.*, vol. 51, no. May 2023, p. 101637, 2024, doi: 10.1016/j.jestch.2024.101637.
[14] D. Hercog, T. Lerher, and M. Trunti ̌, "Design and Implementation of ESP32-Based IoT Devices," no. 1, 2023.
[15] P. Dani, P. Adi, and Y. Wahyu, "Performance evaluation of ESP32 Camera Face Recognition for various projects," vol. 02, pp. 10–21, 2022, doi: 10.31763/iota.v2i1.512.
[16] S. M. Kulkarni, S. Umadi, R. M. Amogh, S. S. Airani, and M. M. Raikar, "ScienceDirect Grading and Classification of Tomatoes Using ESP32-CAM and Impulse Cloud based IoT Platform for Sustainability," *Procedia Comput. Sci.*, vol. 260, pp. 938–946, 2025, doi: 10.1016/j.procs.2025.03.277.
[17] G. R. Pinaso, L. M. Figueiredo, O. R. Júnior, and M. Rogério, "s emina," pp. 1–9, 2024, doi: 10.5433/1679-0375.2024.v45.49197.
[18] I. O. P. C. Series and M. Science, "Weather prediction system and recommendation of plant varieties as an effort to minimize harvest failure with android-based Backpropagation Artificial Neural Networks Weather prediction system and recommendation of plant varieties as an effort to minimize harvest failure with android- based Backpropagation Artificial Neural Networks," 2021, doi: 10.1088/1757-899X/1098/3/032027.
[19] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Artificial Intelligence for the Classification of Plastic Waste Utilizing TinyML on Low-Cost Embedded Systems," vol. 13, no. 6, pp. 2328–2337, 2023.
[20] N. Li, L. Cheng, C. Ji, S. Dongye, and M. Li, "An Improved Framework for Airport Detection under the Complex and Wide Background," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 9545–9555, 2022, doi: 10.1109/JSTARS.2022.3217040.
[21] M. Rina, A. Azullya, and L. Nurliyana, "Deep learning mango fruit recognition based on TensorFlow lite," vol. 9, no. 3, pp. 565–576, 2023.
[22] G. George, S. Adeshina, and M. M. Boukar, "INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Development of Android Application for Facial Age Group Classification Using TensorFlow Lite," pp. 0–3, 2023.
[23] K. Filus and J. Doma, "Computers & Security Software vulnerabilities in TensorFlow-based deep learning applications," vol. 124, 2023, doi: 10.1016/j.cose.2022.102948.
[24] N. Chieng and A. Sallang, "A CNN-Based Smart Waste Management System Using TensorFlow Lite and LoRa-GPS Shield in Internet of Things Environment," vol. 9, 2021.
[25] Y. Adhimanata, I. Sulistiyowati, A. Wisaksono, S. D. Ayuni, U. M. Sidoarjo, and J. Timur, "Overload Monitoring and Warning System for 3-Phase Electric Motorcycle based on IoT," vol. 7, no. 1, pp. 30–41, 2025, doi: 10.12928/biste.v7i1.12262.