INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 16 No. 3 (2024) 212-222
https://publisher.uthm.edu.my/ojs/index.php/ijie

Image Recognition Using Yolov5 for Automotive Industry

Siti Zarina binti Mohd Muji[1]*, Tay Sui Tat[1], Fahmi Danial Haiqal Fazlin Hakimie[1], Abd Kadir Mahamad[1], Mohd Norzali Hj Mohd[1], Suhaila Sari[1], Chua King Lee[1], Muhammad Paend Bakht[2]

[1] *Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA*
[2] *Balochistan University of Information Technology Engineering and Management Sciences, PAKISTAN*

*Corresponding Author: szarina@uthm.edu.my
DOI: https://doi.org/10.30880/ijie.2024.16.03.019

Article Info

Received: 5 July 2024
Accepted: 27 October 2024
Available online: 9 November 2024

Keywords

Yolov5, object recognition, car seat, automotive industry

Abstract

The automotive industry greatly benefits from the implementation of a robust object recognition system, which enhances the efficiency of the inspection process. This paper introduces a GUI system designed to address issues related to operators mistakenly attaching incorrect barcodes to car seats, leading to misplacement in the production line. The study aims to achieve two objectives: firstly, to implement a deep learning framework based on YOLOv5 for car seat pattern recognition, and secondly, to develop a GUI interface that utilizes camera inputs to monitor car seat types and evaluate the performance of the YOLOv5 detection model. To achieve this, the project utilizes the YOLOv5 algorithm for car seat detection, employing a custom dataset comprising three types of car seats: Type A, Type B, and Type C. The dataset is labelled accordingly using the Roboflow website, and Google Colab is utilized for training the custom dataset, which is done over 90 epochs. Subsequently, the GUI system is developed using Tkinter, which enables car seat detection from both static and real-time images. The results obtained from the GUI system showcase the successful detection of car seats based on static and real-time inputs. The average confidence scores for Type A, Type B, and Type C, using the static images method, are found to be 0.53, 0.64, and 0.61, respectively. On the other hand, the average confidence scores using real-time images are slightly lower, with Type A at 0.30, Type B at 0.53, and Type C at 0.38. Upon comparing the results from static and real-time images, it becomes evident that the static image approach yields more accurate detections compared to the real-time image method.

1. Introduction

In the present day, the automotive industry is experiencing rapid growth and advancements. As reported by the Malaysia Automotive Association (MAA) through the automotive industry portal Marklines, car production for February has seen a significant increase of 13.5% compared to the previous year, reaching a total of 51,291 units [1]. This notable expansion indicates the rising significance of the automotive industry in the context of the Fourth Industrial Revolution (IR4.0). As a result, the quality of auto parts used in cars holds a critical influence on the overall performance of vehicles. With the automotive sector becoming a pivotal player in IR4.0, ensuring the

superior quality of auto parts is paramount to driving the industry's success and meeting the demands of an evolving market.

Car seats play a crucial role as auto parts, and ensuring a smooth production process along the conveyor line is of utmost importance, given the wide variety of car seat types manufactured by the company. Presently, the traditional method used to distinguish car seat types before transferring them to the warehouse relies on implementing barcodes. However, this approach has its drawbacks, as it depends heavily on the operator's physical abilities and working conditions to manually affix the appropriate barcode to each complete set of car seats. Unfortunately, errors may arise when operators inadvertently attach the wrong barcode to certain car seats. Such errors can have a significant impact on the overall system, leading to inefficiencies and disruptions in the production process [2]. Consequently, there is an urgent need for a more advanced and dependable solution to address these challenges and maintain a seamless car seat production line.

Within the automotive industry, implementing a top-notch object recognition system can greatly enhance the efficiency and effectiveness of the inspection process. Object recognition, a computer vision technique, plays a crucial role in identifying objects within images or videos, and it stands as a fundamental outcome of deep learning and machine learning algorithms [3]. By employing such a system, the industry can significantly improve inspection procedures, leading to smoother operations and more accurate results. In the realm of auto parts inspection, a vision system is commonly employed to detect defects in smaller components [4]. On the other hand, when dealing with larger components like car seats moving along a conveyor line, the identification and differentiation process typically involves the utilization of a barcode scanner and RFID technology [5]. In the domain of car seat recognition, this work proposes a significant advancement by applying the YOLOv5 model, a state-of-the-art deep learning method. The main contribution of this work is to enhance car seat recognition using advanced deep learning techniques. Unlike the existing industry practice that relies solely on the barcode system for car seat identification, which can occasionally lead to errors during the process, this research aims to introduce a second checking mechanism. By incorporating YOLOv5, the production process can be streamlined, ensuring a more reliable and error-free identification of car seats, thus contributing to smoother operations in the industry.

2. Related Work

Object detection is a computer vision technique focused on identifying objects within images or videos [6]. This technology has a rich history, marked by significant advancements over time. One of the earliest object detection methods emerged in the late 1960s, developed by Paul Viola and Michael Jones [7]. Their pioneering work led to the creation of the Viola-Jones algorithm, which centred around "Rapid Object Detection" using a Boosted Cascade of Simple Features. Initially designed for face detection, this framework proved adaptable to detecting various object classes as well. The field of object detection has continuously evolved, witnessing the development of increasingly sophisticated algorithms and technologies. Throughout its history, researchers and engineers have strived for enhancements, resulting in powerful and versatile solutions for object recognition and localization tasks.

The Viola-Jones algorithm employs four crucial stages for object detection. The first stage involves Haar-like features, which are adept at detecting specific patterns like edges, lines, and corners within images [7]. In the second stage, the integral image (also known as the summed area table) is utilized as a data structure to efficiently compute pixel value sums over rectangular regions in an image [7]. This integral image plays a crucial role in swiftly evaluating Haar-like features, enhancing the algorithm's speed. The third stage incorporates AdaBoost (Adaptive Boosting), a machine learning algorithm used for training classification models [7]. AdaBoost iteratively trains a series of weak classifiers, which are then combined to create a strong classifier capable of making accurate predictions on unseen data. Finally, the fourth stage introduces cascading classifiers, a type of classifier that combines multiple individual classifiers in a cascade to improve overall model performance [8]. The cascading classifier is trained to focus on specific subsets of the data, leading to increased accuracy in predictions [7]. This cascading approach further refines the object detection process, making it more efficient and reliable.

In the field of object detection, significant advancements have emerged over the years. One notable feature introduced during this period is the Histogram of Oriented Gradient (HOG), a feature descriptor widely used in computer vision algorithms for object detection [8]. HOG operates by dividing an image into smaller cells, and for each cell, it calculates a histogram of gradient orientations within that region [9]. This descriptor proves highly effective for object detection because it can capture the shape and structure of objects, even when they are partially viewed from different angles [10] [9].

After 2010, object detection faced limitations as the performance of hand-crafted features reached a saturation point [10]. However, a breakthrough occurred in 2012 when researchers introduced a new technique that utilized convolutional neural networks (CNNs) and deep learning to overcome this problem. The deep learning era characterizes the current state of machine learning, heavily influenced by the development and success of deep learning methods. Deep learning involves the use of artificial neural networks to learn from data and make predictions or decisions. This approach has proven remarkably effective in various applications,

Int. Journal of Integrated Engineering Vol. 16 No. 3 (2024) p. 212-222

including object detection algorithms like YOLO [10]. The adoption of deep learning has revolutionized the field, empowering object detection models to achieve higher accuracy, robustness, and versatility compared to the previous hand-crafted feature-based approaches.

Several factors have played pivotal roles in the continual improvement of object detection algorithms. Among these factors, the availability of increasingly large and diverse datasets for training object detection models stands out. The abundance of data enables algorithms to learn patterns more effectively, leading to enhanced results. Furthermore, the development of powerful computational hardware, such as Jetson Nano, Raspberry Pi, and Nvidia CUDA, has been instrumental in driving progress. These hardware platforms are designed for general-purpose parallel computing, making them well-suited for complex matrix calculations. These calculations can be divided into independent parts and processed simultaneously, significantly speeding up computations compared to single-processor approaches. With this advanced hardware, researchers can train larger and more complex algorithms, which in turn boosts the accuracy of object detection. The synergy of these factors has been the driving force behind the continuous advancement and refinement of object detection algorithms, making them more accurate, efficient, and capable of handling diverse real-world scenarios.

Yolov5 is built upon the PyTorch classifier, making it suitable for both training and detection purposes. The Yolo series initially started with the darknet technology, which evolved into Yolov2, Yolov3, and subsequently Yolov4. However, Yolov5 has emerged as a standout choice due to its leading performance in object detection [11]. One of the remarkable features of Yolov5 is its versatility in various scenarios, including Multiple Object Tracking (MOT) and Face Mask Wearing Recognition, among others. Its adaptability across different applications showcases its effectiveness and applicability. Moreover, Yolov5 has proven its capability to significantly improve accuracy without compromising real-time performance [12]. This balance between accuracy and real-time speed is a critical factor that has contributed to Yolov5's widespread adoption and its ability to excel in demanding real-world detection tasks.

Yolov5 has demonstrated remarkable success in various real-life applications, especially during the COVID-19 pandemic. One notable application is using the Yolov5 algorithm to train models for detecting whether individuals are wearing masks or not [13]. This capability has been invaluable in enforcing mask-wearing protocols and ensuring public safety. Researchers have harnessed the power of Yolov5 in numerous studies, including object tracking, as demonstrated by Tan et al [14]. They leveraged Yolov5 to achieve accurate and efficient tracking of objects in dynamic environments, contributing to advancements in computer vision research. Additionally, Qi et al [15] applied Yolov5 for object detection in a specific context, focusing on litchi (fruit) detection models. They utilized the Pyramid Scene Parsing Network (PSPNet) to crop and obtain segmentation images, enabling precise identification of litchi picking points. Such applications underscore the versatility and effectiveness of Yolov5 in addressing a wide range of real-world challenges across different domains.

Yolov5 has been a popular choice for researchers in various studies, offering improvements in computational latency and performance on resource-constrained devices. Ravi and El-Sharkawy [16] focused on enhancing computational efficiency and found Yolov5n6 to outperform the Single Shot Multibox Detector (SSD) network. Guo et al [17] also concurred with Ravi and El-Sharkawy's findings, observing that Yolov5 exhibited superior speed and accuracy when compared to Yolov3 and R-CNN. Their research centred around detecting defects on steel surfaces in industrial settings. In the realm of license plate detection in unconstrained environments, Khan et al [18] demonstrated that Yolov5 facilitated faster and real-time detection processes, showcasing its suitability for practical applications. Moreover, Chiriboga et al [19] utilized Yolov5 to detect and classify rapid DNA origami nanostructures, while Zhao et al [20] employed Yolov5 for detecting particleboard surface defects. These diverse studies collectively highlight the widespread adoption of Yolov5 in different domains, attesting to its efficacy in various detection tasks.

3. Methodology

This project follows a specific workflow. Firstly, extensive research and exploration are conducted to gain a comprehensive understanding of Yolov5, and related research conducted by other researchers. Next, the dataset is labelled using Roboflow, and the labelled datasets undergo the annotation and augmentation processes. The annotated and augmented datasets are then trained using Google Collab. After the training process, the weight or model parameters obtained are imported into the system. The testing and validation of the datasets are performed using a laptop, leveraging the trained model. To facilitate control over all the processes from the laptop, a developed graphical user interface (GUI) is utilized. This GUI serves as a tool to manage and oversee the various stages of the project, ensuring seamless control and monitoring of the car seat detection system. In summary, the workflow of this project encompasses research and exploration, dataset labelling and augmentation, training using Google Colab, testing and validation using a laptop, and the utilization of a developed GUI for efficient control of the entire process.

The YOLOv5 algorithm has gained significant popularity in the field of object detection. It is widely utilized in various scenarios, including Multiple Object Tracking (MOT) and Face Mask wearing recognition. Google Collab,

developed by the Google Research team, offers a convenient platform for executing Python code directly through a web browser. It is particularly well-suited for machine learning, data analysis, and educational purposes. Google Collab provides a hosted Jupyter notebook service, granting users access to computing resources such as GPUs at no cost. Roboflow is a computer vision platform that enhances the process of building computer vision models. It offers improved techniques for data collection, preprocessing, and model training, resulting in faster and more accurate model development. Tkinter, the standard GUI library for Python, is a powerful tool for creating graphical user interfaces. Combining Python with Tkinter provides a straightforward and efficient approach for developing GUI applications. Tkinter utilizes an object-oriented interface to the Tk GUI toolkit. By leveraging these resources, the car seat GUI system can be successfully implemented in this project.

3.1 Dataset Preparation

The number of datasets used is 969 images for Type A car seats, 1218 images for Type B car seats and 491 images for Type C images. The image size is 640x 640 and for the dataset it is divided into 3 groups, training 60%, validation,30% and testing 10%. All datasets are taken in the same lighting at the storage car seat location at car seat factory in Selangor. Figure 1 shows the car seat type A, B and C which shows the difference in their appearance.

(a) **(b)** **(c)**

Fig. 1 *Car seat (a) Type A; (b) Type B; (c) Type C*

3.2 Annotation and Augmentation Process Using Roboflow

The annotation process involves labelling or marking the objects of interest within an image dataset. In the context of car seat detection, annotations are used to identify and outline the car seats present in the images. This process typically is by manually drawing the bounding boxes around the car seats in each image, indicating their location and extent using Roboflow software in Yolov5 format.

Augmentation is a technique used to enhance the training data by applying various transformations or modifications to the existing images. The goal of augmentation is to increase the diversity and variability of the dataset, helping the model generalize better to different real-world scenarios. Data augmentation techniques, including 90° rotations (clockwise, counter-clockwise) and upside-down manipulation, were applied to enhance the training data.

This section discusses the results obtained from the surface pressure measurement study. The effects of angle of attack, Reynolds number and leading edge bluntness are discussed in the next sub section.

3.3 Training in the Google Colab

Once the data already prepared from annotation and augmentation from previous process, the data now can be trained at Google Collab. Here, NVIDIA Tesla V100 SXM2 is used for training purpose. The environment needs to be set up by providing the necessary dependencies and libraries for training the car seat detection model, such as PyTorch and YOLOv5.

3.4 Image Sources for Testing Purpose

The car seat detection system is built around a laptop, which serves as the main controller for the recognition process. It is connected to a camera for capturing car seat images. The first method of the system involves using the camera to capture car seat images, which are then processed and displayed in the GUI system as shown in Fig 2. The results of the car seat detection process are shown directly in the GUI interface. In the second method, an Input file folder is created, and car seat pictures are placed inside this folder. By clicking the "Open camera" button, the pictures from the Input folder are sent to the backend for the detection process. The car seat pictures are analyzed and identified, and the results are saved in the Output file, specifically in the runs/detect/exp directory. In summary, the car seat detection system employs two methods: one using real-time camera capture and

Int. Journal of Integrated Engineering Vol. 16 No. 3 (2024) p. 212-222

immediate GUI display, and the other utilizing an Input folder for batch processing of car seat images, with the results being saved in the Output file directory.

Fig. 2 GUI system

4. Results and Discussion

4.1 Comparison of Training and Validation Result

The model's performance was evaluated using key metrics such as Precision, Recall, and mAP (mean Average Precision) at different intersection over union (IoU) thresholds, specifically 0.5 (50%) and 0.95 (95%). IoU measures the overlap between predicted and ground truth bounding boxes, providing insights into the model's localization accuracy. The performance of the YOLO model is influenced by three main factors: mAP, precision, and recall. mAP combines precision and recall to assess the overall accuracy of object detection. Precision measures the accuracy of object predictions, specifically how accurately the model identifies the positive class. Recall, on the other hand, quantifies how effectively the model detects objects or classes, indicating its ability to identify true positives.

In summary, YOLOv5s was employed to train the model using annotated data in YOLO format. Data augmentation techniques were applied, and the model's performance was evaluated using metrics such as mAP, precision, and recall assessing its object detection accuracy. Figure 3 illustrates the metrics curves during the training process, showing the progression of the model's performance. Following evaluation, the YOLOv5 model achieved a validation precision score of 0.99325, recall score of 0.97988, and mAP scores of 0.97818 (@0.5IOU) and 0.78885 (@0.95IOU). These results demonstrate the effectiveness of the YOLOv5 method in object detection. Comparing Figures 3 and 4, it is evident that the mAP steadily increased throughout the range of 0-90 epochs. As the model approached the final epoch, the mAP score reached approximately 0.9786 or 97.86%. This indicates that the value of the epoch parameter influences the result, with an increasing number of epochs leading to improved precision within a certain limit. However, it is important to note that an excessive number of epochs can potentially cause the model to over-fit the training data. In general, the training process using the YOLOv5s architecture, with an appropriate number of epochs, yielded promising results, as evidenced by the steadily increasing mAP and high precision and recall scores achieved during evaluation.

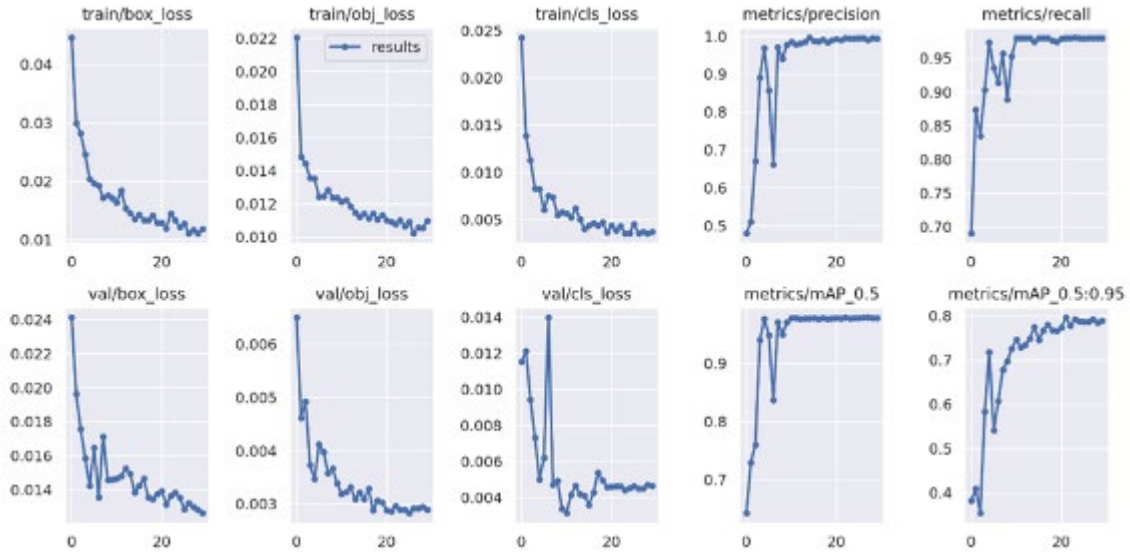

Fig. 3 *Results from training using 30 Epochs*

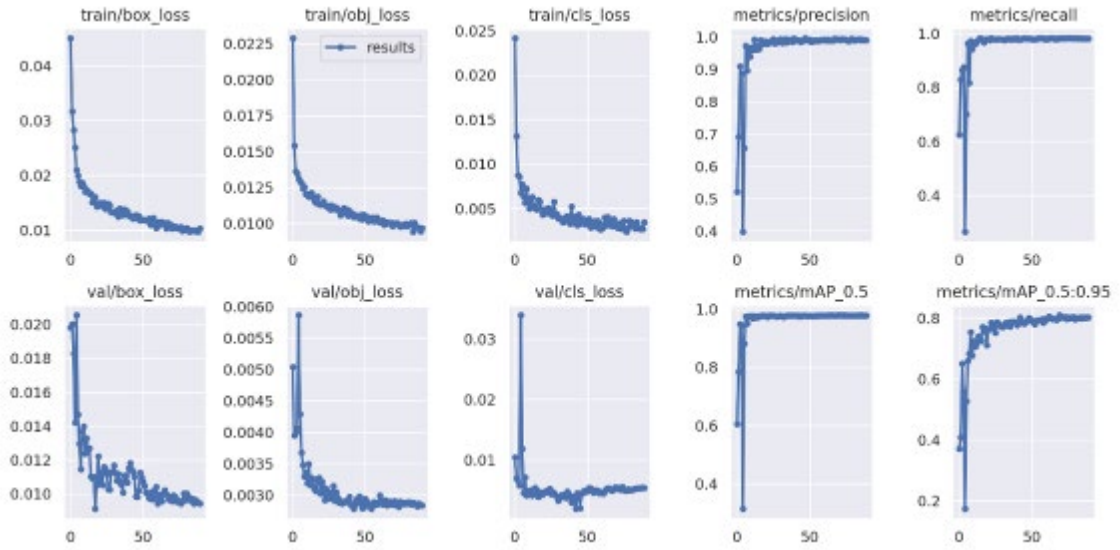

Fig. 4 *Results from training results using 90 epochs*

4.2 Comparison of Confusion Matrix

Authors including an appendix section should do so before the References section. Multiple appendices should all have headings in the style used above. They will automatically be ordered A, B, C etc. A confusion matrix serves as a concise representation of predicted results in a classification problem. It summarizes the count of correct and incorrect predictions for each class, enabling a comprehensive analysis of the model's performance. By examining the confusion matrix, one can identify areas of confusion in the classification process. The confusion matrix comprises four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These components provide a detailed breakdown of the model's predictions, surpassing the limitations imposed by relying solely on classification accuracy. In essence, the confusion matrix offers a more nuanced perspective on the classification outcomes, allowing for a thorough evaluation of the model's strengths and weaknesses. By dissecting the results into TP, TN, FP, and FN, it becomes possible to gain deeper insights into the model's performance beyond a simple accuracy measure.

Fig 5 presents the confusion matrix for Type A, Type B, and Type C, illustrating True Positive (TP) instances where the model successfully makes accurate predictions for these classes. However, it also shows False Positive (FP) instances, indicating that the model mistakenly detects the background as one of the target classes. On the other hand, Figure 6 demonstrates that there are no False Positive instances related to the background class. This

Int. Journal of Integrated Engineering Vol. 16 No. 3 (2024) p. 212-222

indicates that the model successfully avoids detecting the background as one of the target classes. In summary, Figure 5 highlights the presence of False Positive cases where the model incorrectly identifies the background as one of the classes. However, Figure 6 demonstrates an improvement in the model's performance, with no instances of False Positive predictions related to the background class.

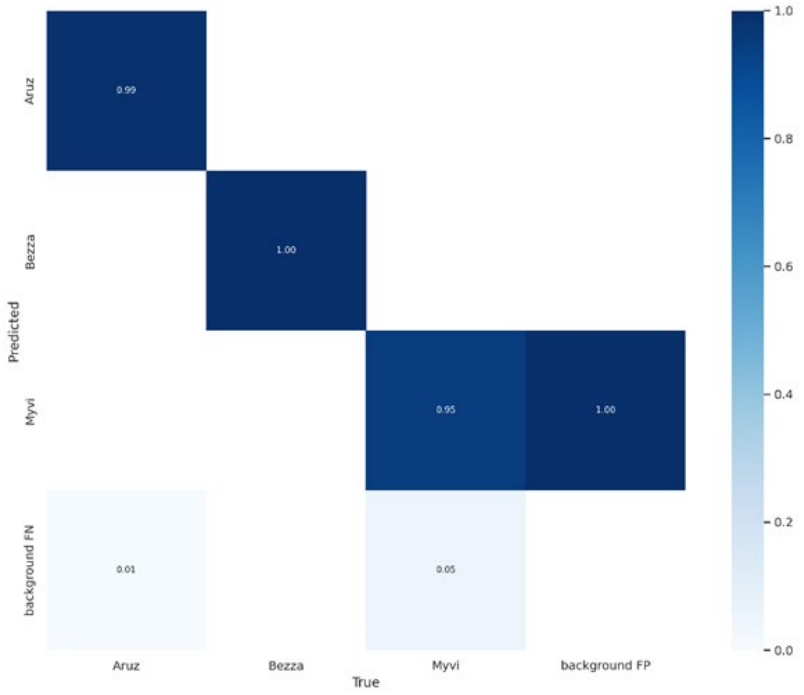

Fig. 5 *Confusion matrix after 30 epochs*

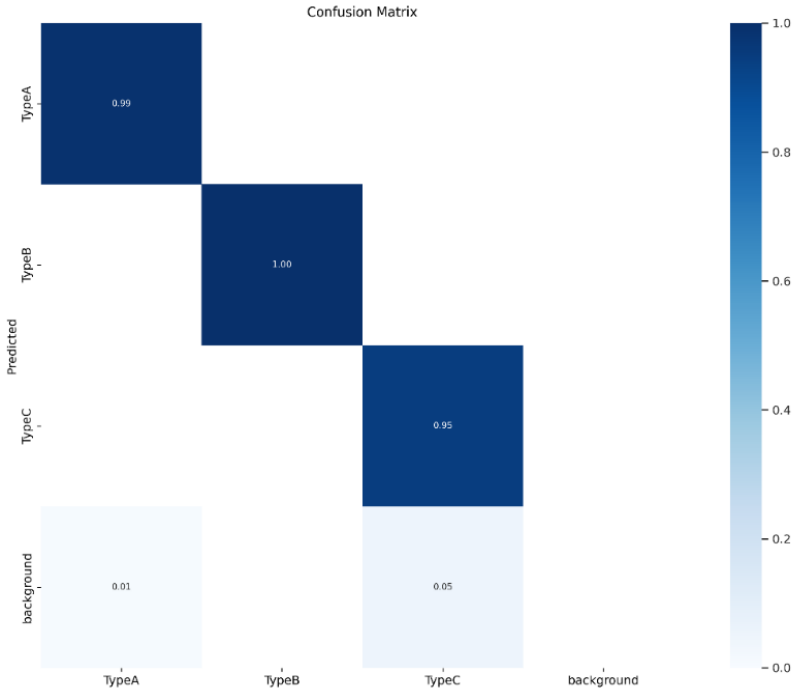

Fig. 6 *Confusion matrix after 90 epochs*

4.3 Car Seat Detection Using the Static Image

The detection process remains consistent with the first method. However, this method introduces a new step. Firstly, an Input folder file, as depicted in Figure 7, is created. Following that, static images are placed into the Input folder, as illustrated in Figure 8. Subsequently, another python file called "apphc.py" is executed in the

Anaconda prompt. This file reads all the images in the Input folder and performs the detection process. The resulting detections are saved in the runs/detect/exp folder, and prior to each new detection process, all previous results are removed to ensure a clean slate. Table 1 presents the testing results for three types of car seats, namely Type A, Type B, and Type C, using static images. The average confidence score for Type A is 0.53, for Type B is 0.64, and for Type C is 0.61.

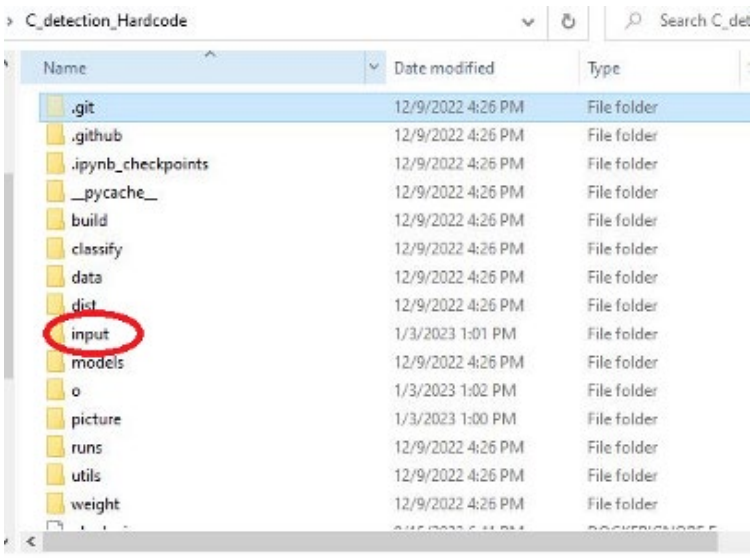

Fig. 7 *Create input folder*

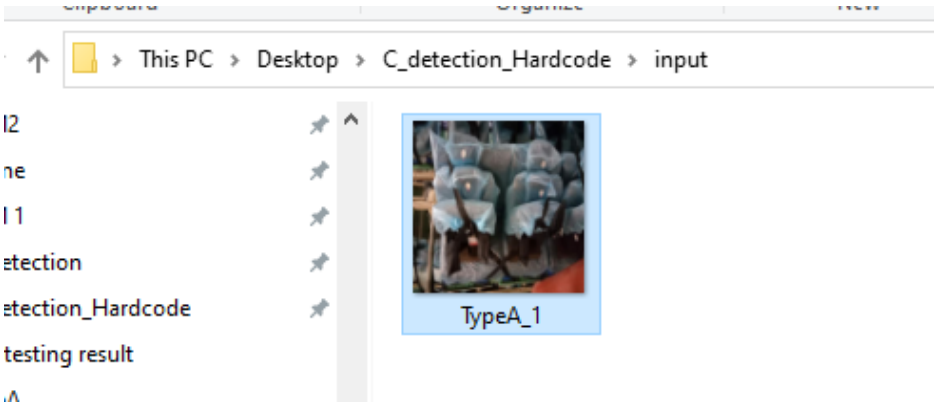

Fig. 8 *Place the images data into Input file*

Table 1 *The confidence score of car seat detection by using the static image*

Type A	Score	Type B	Score	Type C	Score
Type A 1	0.62	Type B 1	0.67	Type C 1	0.68
Type A 2	0.34	Type B 2	0.66	Type C 2	0.72
Type A 3	0.68	Type B 3	0.76	Type C 3	0.72
Type A 4	0.54	Type B 4	0.39	Type C 4	0.52
Type A 5	0.65	Type B 5	0.33	Type C 5	0.66
Type A 6	0.78	Type B 6	0.30	Type C 6	0.68
Type A 7	0.72	Type B 7	0.74	Type C 7	0.80
Type A 8	0.73	Type B 8	0.72	Type C 8	0.81
Type A 9	0.68	Type B 9	0.72	Type C 9	0.80
Type A 10	0.64	Type B 10	0.71	Type C 10	0.65
Average	0.68	Average	0.6	Average	0.7

Int. Journal of Integrated Engineering Vol. 16 No. 3 (2024) p. 212-222

4.4 Car Seat Detection Using the Real Time Image

The first method involves using real-time images for car seat detection. To begin, the smartphone needs to be positioned in front of the laptop webcam, displaying the dataset. In the graphical user interface (GUI), there is an "Open Camera" button shown in Figure 9, which activates the webcam. When pressed, the camera captures an image, and the GUI system proceeds to identify the type of car seat present. If the car seat is successfully detected, the GUI displays a "Pass" message, as depicted in Figure 9. However, if the car seat cannot be identified, a "Fail" statement is displayed instead.

For example, if the camera successfully detects a Type B car seat, the detection status in the GUI will be marked as "Pass". Conversely, if the camera fails to detect any car seat, the status will be indicated as "Fail". The results of the detection process are saved in the runs/detect/exp folder, as illustrated in Figure 10. Table 2 displays the testing results for the same three types of car seats, but using real-time images. In this case, the average confidence score for Type A is 0.30, for Type B is 0.53, and for Type C is 0.38.

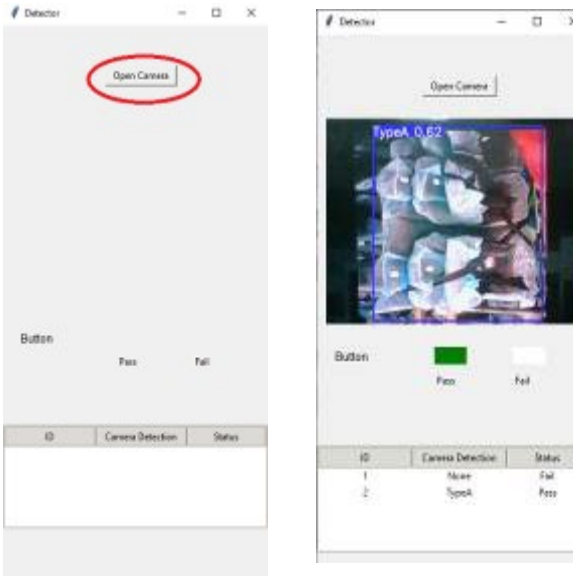

Fig. 9 *Result from real time image*

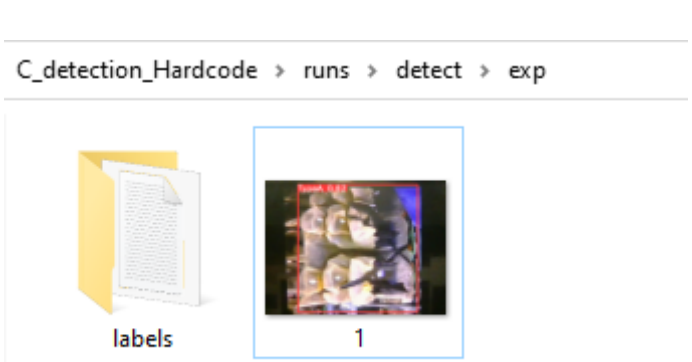

Fig. 10 *Result saved in runs/ detect/exp folder*

Table 2 *The confidence score of car seat detection by using real time image*

Type A	Score	Type B	Score	Type C	Score
Type A 1	0.27	Type B 1	0.64	Type C 1	0.39
Type A 2	0.34	Type B 2	0.59	Type C 2	0.27
Type A 3	0.75	Type B 3	0.71	Type C 3	0.68
Type A 4	0.38	Type B 4	0.48	Type C 4	0.29
Type A 5	0.51	Type B 5	0.38	Type C 5	0.43
Type A 6	0.61	Type B 6	0.37	Type C 6	0.37
Type A 7	0.49	Type B 7	0.44	Type C 7	0.34
Type A 8	0.50	Type B 8	0.31	Type C 8	0.52
Type A 9	0.57	Type B 9	0.58	Type C 9	0.48
Type A 10	0.45	Type B 10	0.33	Type C 10	0.41
Average	0.49	Average	0.48	Average	0.42

4.5 Analysis Between Static and Real Time Image

The comparison of these two methods demonstrates the significance of the image capture environment in relation to the car seats GUI system. In essence, the results clearly indicate that the static image method outperforms the real-time image method. When comparing the results obtained from real-time image usage, the confidence scores for car seat detection are lower compared to the confidence scores achieved through the utilization of static

images. It can be summarized that, Yolov5 is perform well with static image as this is the same lighting for the dataset taken at the factory, while for real time image, many factors could affect its result specifically the different lighting and the camera resolution where the camera used is from laptop camera. Therefore, to enhance the result for real time, the same lightning process and camera should be used.

5. Conclusion

In summary, the project has successfully delved into the theoretical study of deep learning in object detection, particularly focusing on previous research on Yolov5. The dataset labelling process for three types of car seats, along with the training of the YOLOv5 model using Google Collab and subsequent testing of the car seat dataset using the Anaconda prompt on a laptop, has been completed. The GUI design for the project has also been finalized, although it may not be as advanced as the latest version available. Furthermore, it is important to note that the accuracy of the car seat detection system relies heavily on the number of datasets utilized. In this project, the performance of static image for three types of car seat is much better compared to real time image where for static image it gives accuracy average value 0.68, 0.6 and 0.7 while for real time image it gives lower than that which is 0.49, 0.48 and 0.42 for type A, B and C respectively. In this project, the number of datasets is relatively small, which could impact the overall performance. Additionally, the simplicity of the tested scenes in comparison to the complex background of car seats in a production line poses a challenge in extending the trained model to real-life production line scenarios for car seat detection. Therefore, an increase in the dataset size is recommended to enhance the effectiveness of the system. Moreover, there is an opportunity to expand the functionality of the GUI system by incorporating features such as a database and an alarm system. These additions would enhance the overall utility and practicality of the project.

Acknowledgement

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1(vot Q354).

Conflict of Interest

There is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Siti Zarina Mohd Muji, Tay Su Tat, Fahmi Daial Haiqal Fazlin;* ***data collection:*** *Abd Kadir Mahamd;* ***analysis and interpretation of results:*** *Mohd Norzali Hj Mohd;* ***draft manuscript preparation:*** *Suhaila Sari, Chua King Lee, Muhammad Paend Bakht. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Malaysia Automotive Production volume, 2022, Retrieved April 22, 2022, from https://www.marklines.com/en/statistics/flash_prod/automotive-production-in-malaysia-by-month
[2] Li Xiaoguang, Zhu Juan, Wang Chao (2018) Real-time detection system for automobile car seat based on vision. *Proceedings of the 2018 3rd International Workshop on Materials Engineering and ComputerSciences (IWMECS 2018)* https://doi.org/10.2991/iwmecs-18.2018.118
[3] Sandhya Devi, M. R. S., Vijay Kumar V. R. and P. Sivakumar (2021) A Review of image Classification and Object Detection on Machine learning and Deep Learning Techniques, *5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India : 1-8. doi: 10.1109/ICECA52323.2021.9676141
[4] Keyence (2019) Unleash the Potential of your Vision System, All in One Vision System Applications.
[5] Avery Dennison (2021) Automotive Applications Bonding, labelling, fastening and tracking solutions for the automotive seat manufacturer.
[6] Zhao, Z. -Q., Zheng, P., Xu, S. -T. and Wu, X. (2019) Object Detection With Deep Learning: A Review, *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212-3232, https://doi.org/10.1109/TNNLS.2018.2876865
[7] Viola, P., and Jones, M. J. (2004) Robust Real-Time Face Detection, *International Journal of Computer Vision 57*(2), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
[8] Zhengxia, Z., Zhenwei, S., Yuhong, G., and Jieping,Y. (2023) Object Detection in 20 Years: A Survey, *Proceedings of the IEEE 111(*3), 257-276. https://doi.org/10.48550/arXiv.1905.05055
[9] Dalal, N., and Triggs, B. (2005) Histogram of Oriented[Gradients for Human Detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 886-893. https://doi.org/10.1109/CVPR.2005.177.

[10] Chinmoy Borah (2020). Evolution of Object Detection. Retrieved on December 31,2022, from https://medium.com/analytics-vidhya/evolution-of-object-detection-582259d2aa9b
[11] Tan, L., Dong, X., Ma Y. and Yu, C. (2018) A Multiple Object Tracking Algorithm Based on YOLO Detection," *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI*), Beijing, China, 1-5. https://doi.org/10.1109/CISP-BMEI.2018.8633009
[12] Juan R. T., Diana, M. C.-E. (2023) A Comprehensive Review of Yolo: From Yolov1 to Yolov8 and Beyond. *ACM Computing Surveys*. https://doi.org/10.48550/arXiv.2304.00501
[13] Yang, G., Feng, W., Jin, J., Lei, Q, Li, X, Gui, G., and Wang, W. (2020) Face Mask Recognition System with YOLOV5 Based on Image Recognition, *IEEE 6th International Conference on Computer and Communications (ICCC),* 1398-1404. https://doi.org/10.1109/ICCC51575.2020.9345042.
[14] Tan, L., Dong, X., Ma, Y. and Yu, C. A Multiple Object Tracking Algorithm Based on YOLO Detection (*2018) 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) Beijing, China*, https://doi.org/10.1109/CISP-BMEI.2018.8633009.
[15] Qi, X., Dong, J., Lan, Y., & Zhu, H. (2022) Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet, *Remote Sensing* 14(9), http://dx.doi.org/10.3390/rs14092004
[16] Ravi, N., and El-Sharkawy, M. (2022) Real-Time Embedded Implementation of Improved Object Detector for Resource-Constrained Devices, *Journal of Low Power Electronics and Applications, 12*(2), http://dx.doi.org/10.3390/jlpea12020021
[17] Guo, Z., Wang, C., Yang, G., Huang, Z., and Li, G. (2022) MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface, *Sensors 22*(9), 3467. https://doi.org/10.3390/s22093467
[18] Khan, I. R., Ali, S. T. A., Siddiq, A., Khan, M. M., Ilyas, M. U., Alshomrani, S., and Rahardja, S. (2022) Automatic License Plate Recognition in Real-World Traffic Videos Captured in Unconstrained Environment by a Mobile Camera, *Electronics 11*(9),1408, http://dx.doi.org/10.3390/electronics11091408
[19] Chiriboga, M., Green, C.M., Hastman, D.A., Mathur, D., Wei, Q., Diaz, S. A., Medintz, I.L., Veneziano, R. (2022) Rapid DNA origami nanostructure detection and classification using the YOLOv5 deep convolutional neural network, *Sci Rep 12*, 3871 https://doi.org/10.1038/s41598-022-07759-3
[20] Zhao, Z., Yang, X., Zhou, Y., Sun, Q., Ge, Z. (2021) Real-time detection of particleboard surface defects based on improved YOLOV5 target detection, *Sci Rep 11*, 21777, https://doi.org/10.1038/s41598-021-01084-x