© Universiti Tun Hussein Onn Malaysia Publisher's Office

JTS

http://penerbit.uthm.edu.my/ojs/index.php/jts
ISSN : 2229-8940 e-ISSN : 2600-7940

Journal of Techno-Social

AI-Based Analytics for Hawkers Identification in Video Surveillance for Smart Community

Noraini Azmi[1,3], Latifah Munirah Kamarudin[1,3*], Ammar Zakaria[3], Syed Muhammad Mamduh Syed Zakaria[1,3]

[1]Faculty of Electronic Engineering & Technology (FKTEN),
Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, MALAYSIA

[2] Faculty of Electrical Engineering & Technology (FKTE),
Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, MALAYSIA

[3]Centre of Excellence for Advanced Sensor Technology (CEASTech),
Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/jts.2023.15.02.008
Received 02 October 2023; Accepted 21 November 2023; Available online 21 December 2023

Abstract: Street hawking is a widespread phenomenon in urban areas globally, presenting challenges for local authorities such as traffic congestion, waste management, and negative impacts on the city's image. This research addresses key issues faced by authorities in managing hawkers, including the resistance to formalization, maintaining urban aesthetics, waste disposal, and understanding user preferences. The study investigates the performance of the You Only Look Once (YOLO) algorithm, utilizing Convolutional Neural Networks (CNN) for real-time object detection. To achieve this objective, the YOLOv5 algorithm is trained with a custom image dataset collected from the same camera along the street in the city area to detect five classes of objects, namely umbrella, table, stool, car, and people. Real images that were captured via camera and video surveillance were compiled as datasets which are then used to train and test the algorithm. The study aims to provide insights into the data collection process of hawkers along the street around the areas and the development of real-time hawker detection for the smart city application.

Keywords: Artificial intelligent, Hawker identification, video surveillance, smart community, real-time

1. Introduction

The growth of hawkers has contributed to various problems for local authorities and the public. AI-based analytics for Hawker refers to the use of artificial intelligence techniques such as machine learning, computer vision, and natural language processing to analyze data and detect street vendors or hawkers who operate without proper licenses or permits. This technology can be used to identify hawkers, as well as to predict and prevent trading activities. Besides that, pedestrians prefer to walk on the streets because of the obstacles that the hawkers have established. Since the hawkers use the space, there is a chance of an accident involving a moving car. Aside from customers stopping their vehicles by the road and blocking off one lane, the residents also complained about the lack of cleanliness and hawker vehicles blocking the access roads in their neighborhood. Hawker stalls that are not regulated can be an eyesore, contributing to hygiene problems in the area and disrupting pedestrian walkways or traffic. Besides that, there are numerous active hawkers, including those who operate along roadsides, on school grounds, and elsewhere. As a result, the object is usually invisible to the human eye or camera. This issue persists because it is difficult to detect hawkers in

*Corresponding author: latifahmunirah@unimap.edu.my

real-time, labor-intensive, and time-consuming since authorities need to patrol and monitor the situation throughout the whole state as the stall can be set up anywhere and anytime. Hence, makes it difficult for the local authorities to tackle the issue. To minimize the activities of unlawful hawkers, real-time detection of objects such as umbrellas, tables, stools, cars, and people is required.

This research aims to develop real-time object detection in smart city applications for hawker detection where the system can be further enhanced for the identification of illegal hawkers. Another advantage of AI-based analytics for hawkers is that it can be integrated into smart communities, which are communities that use technology to improve the quality of life for residents and visitors. Smart communities use a variety of technologies such as sensors, cameras, and data analytics to gather and analyze data about the community and use this data to improve services such as transportation, public safety, and environmental monitoring. One of the main advantages of AI-based analytics for hawkers' detection is that they can quickly and accurately identify and track hawkers, even in crowded and chaotic environments. This can help authorities to take action against illegal hawkers more efficiently and effectively. To better understand the properties of these foreign objects, some researchers have devised efficient ways to detect them by relying on computers to identify elements of interest automatically, and a Convolutional Neural Network (CNN) framework based on the YOLO series has good detection speed performance. It was never simple to deal with enormous amounts of image and video data, especially when they contained important information. However, YOLOv5 has achieved substantial progress in object detection through the introduction of YOLOv3 and YOLOv4. YOLOv5 is more robust and can recognize characteristics in photos even better than YOLOv3 and YOLOv4.

2. Literature Review

One of the main approaches used in the literature for object recognition algorithms is YOLO (You Only Look Once). Kristo, Ivasic-Kos, and Pobar (2020) used YOLO to detect objects in difficult weather conditions. Initially, the convolutional neural network (CNN) model was developed in RGB for detecting pictures to examine the challenge of automated human recognition in thermal images. After being retrained using a dataset of thermal images acquired from movies depicting illegal border crossings and travels through protected territories, faster R-CNN, SSD, Cascade R-CNN, and YOLOv3 perform. The films were taken at various distances and with various movement patterns during the night, in clear, rainy, and foggy conditions. Kristo, Ivasic-Kos, and Pobar (2020) aim to evaluate the performance of these models in detecting people in thermal images, which can be challenging due to the low resolution and poor quality of the images. The use of thermal images can be useful in scenarios where visibility is low, such as at night or in adverse weather conditions. Kristo, Ivasic-Kos, and Pobar (2020) conduct experiments on their dataset of thermal images and analyze the results and discuss any challenges or limitations encountered during the research. This study provides insight into the ability of cutting-edge object detectors to recognize individuals in thermal pictures and can be useful for other researchers and practitioners working on similar problems.

In a different study by Algiriyage *et al.* (2021) the YOLO algorithm was used to predict traffic flow in real-time from low-quality CCTV video data using computer vision algorithms. The ability to estimate traffic flow is important for identifying road traffic patterns, which can contribute to traffic. The results of the study using low-frame-rate video demonstrate the performance of the custom-trained YOLOv4 (You Only Look Once version 4) model, with an F1-score greater than 0.95. This indicates that the model can accurately detect cars in the footage and can be used for estimating traffic flow in real-time. This study highlights the potential of using computer vision algorithms and custom-trained models for estimating traffic flow from low-quality CCTV video footage, which can be useful for traffic modeling and traffic management.

Apart from that YOLOv3 was used for electrical appliance status monitoring for home automation and human tracking using video surveillance (Sukkar, Kumar, & Sindha, 2021). YOLOv3 was used for detecting humans and the tracking algorithm (Deep SORT) allows the system to track the movement of humans and maintain consistency in the detection results. To enhance the reliability of the system, the authors implement an advanced tracking-by-detection technique called Deep SORT, which constantly tracks the movement of human beings. They also show a modification of the person re- identification model to improve the performance of the Deep SORT tracking. The re-identification model is used to identify and track individuals throughout the video, which is important for maintaining consistency in the tracking results. This study demonstrates the potential of using YOLOv3 in combination with advanced tracking techniques for human detection and identifying the status of appliances in real-world scenarios. Similarly, Sukkar, Kumar, and Sindha (2021) used YOLOv5 for real-time pedestrian detection. Sukkar, Kumar, and Sindha (2021) focus on the challenges of detecting pedestrians, which include variations in the background image, age, gender, clothes, lighting, and occlusion. Sukkar, Kumar, and Sindha (2021) focus on using YOLOv5 for detecting pedestrians, by training and fine-tuning the model on a dataset of pedestrian images. Similarly, Pawar (2019) used deep learning with CCTVs and drones for detecting crimes and assisting law enforcement agencies. Pawar (2019) proposed YOLO to reduce violence by using image-based extraction techniques with smart methods of deep learning and artificial intelligence to support security officials in detecting, monitoring, and tracking perpetrators.

The updated YOLOv5 algorithm was also used in the real-time detection of foreign objects by ground-penetrating radar (Qiu, Zhao, Chen, Zeng, Huang, & Xiang, 2022), and object detection on the roadside (Huang & Huang, 2022).

(Qiu, Zhao, Chen, Zeng, Huang, & Xiang, 2022) developed an algorithm for real-time ground- penetrating radar target identification utilizing the YOLO series neural network architecture (GPR) images. The YOLO series of neural networks, including YOLOv5, are known for their good detection speed performance, making them suitable for real-time applications. Qiu, Zhao, Chen, Zeng, Huang, and Xiang (2022) found that the YOLOv5 model has enhanced durability and the ability to detect objects in GPR pictures compared to other algorithms. To verify this, they compared the performance of other outstanding target detectors, such as Fast R-CNN, the upgraded YOLOv5 model, and use the SSD (Single Shot MultiBox Detector), under the same training scenario. The results of the comparison show the enhanced YOLOv5 model is more suited for the accurate detection of targets in GPR images. This is likely due to its ability to accurately detect and distinguish features in the images, as well as its fast detection speed. YOLOv5 and other algorithms based on YOLO series neural networks are suitable for real-time detection of targets in GPR images, and YOLOv5 has improved robustness and is better able to discern features in GPR pictures compared to other algorithms. Based on literature, YOLOv5 is found suitable for automatic detection and identification of hawkers in images and videos. Moreover, based on a study by Kristo, Ivasic-Kos, and Pobar (2020), the YOLO algorithm is also suitable for detecting hawkers in images and videos, even in crowded and chaotic environments. In another research, Liu and Liu (2021) used YOLO to map the street vendor in the city area. They successfully produced the kernel density map of the detected street vendor.

3. Object Detection Algorithms

Object detection algorithms have been commonly used for object detection tasks, including but not limited to face recognition, traffic monitoring, foot traffic analysis, inventory management, anomaly and defect detection, and parking occupancy. Convolutional Neural Networks (CNNs) is one of the early machine learning approaches that were used for object detection such as the detection of objects in vending machines (Wang, Shi, Wang, Nan, & Lian, 2019), and retail product detection (Wei, Tran, Xu, Kang, & Springer, 2020). R- CNN (Regions with CNN features) is a popular CNN-based approach for object detection. It uses selective search to identify regions of interest in an image and then passes these regions through CNN for object classification. R-CNN methods is composed of three main stages: region proposal, feature extraction, and classification. The region proposal stage generates a set of region proposals, the feature extraction stage extracts feature from each proposal, and the classification stage classifies each proposal into one of the predefined classes. R-CNN and its variants such as Fast R- CNN and Faster R-CNN, have been used in several studies for object identification (Charitidis *et al.*, 2023). For example, in one study, a Fast R-CNN model was trained on a dataset of images containing illegal hawkers and tested on a separate dataset of images. The results showed that the model was able to accurately detect illegal hawkers with a high level of precision. Another study used Faster R-CNN to detect illegal hawkers in real-time using a camera mounted on a drone. The results showed that the model was able to detect illegal hawkers with a high level of accuracy, even in challenging environments such as crowded markets. However, the R-CNN method has some limitations, such as being computationally intensive and slow. Nevertheless, the literature review shows that the R-CNN method has good performance for object detection tasks, including illegal hawkers' identification in smart communities.

YOLO (You Only Look Once) is a real-time object detection model proposed by Joseph Redmon of the University of Washington in 2016. It is based on the GoogLeNet model, a deep CNN trained for image classification where it uses a single CNN to predict bounding boxes and class probabilities for multiple objects within an image. YOLO provides information about the input image and all the objects within it, which enables it to detect objects in real time with high precision. It does this by dividing the image into a grid of cells, for each cell, it predicts the bounding boxes. One of the advantages of YOLO is its simplicity, it can be implemented in less than 30 lines of Python code. The architecture of YOLO is designed to be efficient and fast, allowing the model to predict bounding boxes and class probabilities for multiple objects in an image in real time. It is efficient by using a single network for all object detections, rather than multiple networks for each object. This makes YOLO suitable for real-time object detection tasks because it uses a grid-based approach to divide an image into smaller regions and then uses a CNN to predict the class probabilities for each region. Each grid cell is responsible for detecting the objects. YOLO also uses anchor boxes, which are pre-defined bounding boxes that provide the model with additional information about the size and shape of objects in the image, helping to improve the accuracy of object detection. According to the author's knowledge, there are yet studies conducted to detect hawkers using YOLO algorithms, especially in the context of the Malaysian environment and dataset.

In this paper, the focus is on the datasets and algorithms used for object detection, specifically for the task of hawkers' identification in smart communities. The most common algorithms that are used in this context include Region-based Convolutional Neural Networks (R-CNN), Fast Region-based Convolutional Neural Networks (Fast RCNN), and You Only Look Once (YOLO). These algorithms were tested using the Pascal Visual Object Classification dataset. The Pascal VOC dataset is a popular dataset used for object detection tasks in computer vision. It includes images of various objects and scenes, with annotations of the bounding boxes and class labels of the objects within the images. This dataset has been used as a benchmark for several object detection algorithms, such as R-CNN, Fast RCNN, and YOLO. The R-CNN algorithm uses selective search to identify regions of interest in an image and then classifies each region using a CNN. The Fast R-CNN algorithm is an improvement over R-CNN, which uses a shared convolutional layer to extract features from the entire image, rather than from individual regions. The YOLO algorithm, on the other hand, uses a single CNN to predict the bounding boxes and class probabilities for all objects in an image. YOLO is known for its speed and efficiency, making it suitable for real-time object detection tasks. By testing these algorithms on the Pascal VOC dataset, the performance of each algorithm can be compared in terms of accuracy, efficiency, and other metrics. This comparison can give insight into which algorithm is the most suitable for the task of illegal hawkers' identification in smart communities.

4. Hawkers Detection Algorithm

One of the approved cutting-edge models, YOLOv5, has outstanding support and is simpler to utilize in production. The fact that YOLOv5 is natively implemented in PyTorch is the finest aspect. YOLOv5 is one of the cutting-edge models for object detection that has gained popularity in recent years. One of the major advantages of YOLOv5 is its ease of use and support. The model is natively implemented in PyTorch, which is a popular deep learning framework known for its ease of use and flexibility. This means that it can be easily integrated into existing PyTorch-based projects and is simple to use in production. Additionally, YOLOv5 has many pre-trained weights available, which can be fine-tuned to adapt to specific tasks and datasets. YOLOv5 also has a lightweight architecture, which makes it suitable for real-time object detection tasks and can run on a variety of devices, including edge devices.
Overall, YOLOv5 was selected for the identification of hawkers in this study due to its ease of use, support, and real-time capabilities. This study investigates AI-based analytics for hawkers' identification in a smart community. This project aims to investigate the performance of You Only Look Once (YOLO) in identifying hawkers in the Malaysian environment. Figure 1 and Figure 2 show the overall architecture of the hawkers' detection system and the process of the hawkers' detection, respectively.

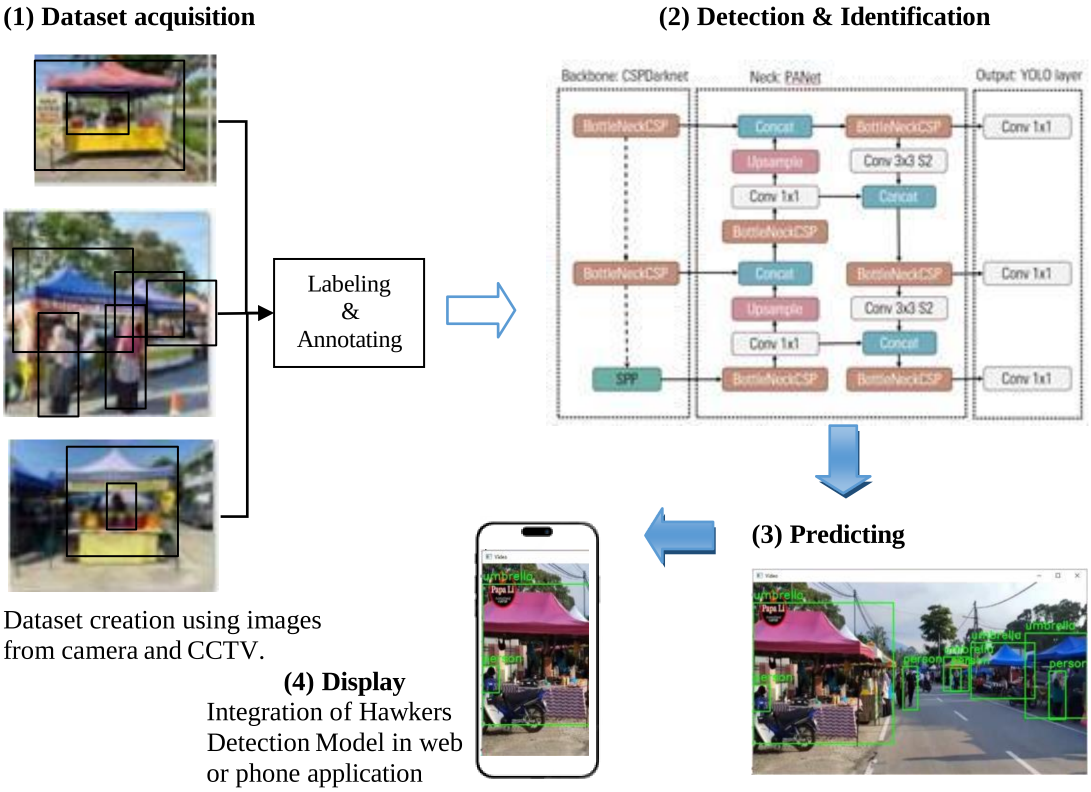


Fig. 1 - System architecture for The Hawkers' detection system

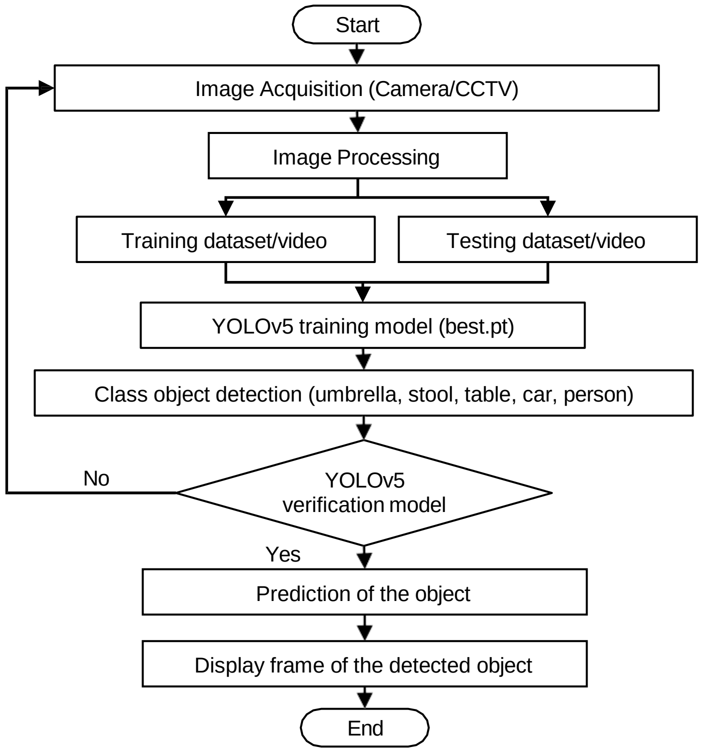


Fig. 2 - Flowchart of The Hawker detection process

The initial stage involves dataset creation which requires capturing images using cameras and CCTV. Next, the dataset was labeled and annotated to identify five classes of objects in the image namely umbrella, table, stool, car, and people. The YOLOv5 algorithm is trained with the custom image dataset collected from the same camera along the street in the city area to detect five classes of objects namely umbrella, table, stool, car, and people. The Hawkers Detection Model was then obtained and integrated into the web page or mobile phone application for a better viewing experience and flexibility. The details of each process are as follows:

- **Image acquisition**: the initial step involves obtaining a real-time video stream from CCTV, which entails capturing images or videos using a camera or CCTV.
- **Image pre-processing**: the steps taken to format images before they are used by model training and inference.
- **Training dataset/video**: which signifies the utilization of a dataset for training purposes.
- **Testing dataset/video**: where the dataset is evaluated to assess the system's performance.
- **YOLOv5 training model (best.pt):** the YOLOv5 training model is responsible for generating the best.pt file, which represents the optimal trained model achieved during the training process.
- **Class objects detection:** Class object detection involves identifying various objects, including umbrellas, tables, stools, cars, and people.
- **YOLOv5 verification model:** the system verifies the correctness of the object class. If it is incorrect, the process repeats itself by returning to image acquisition. On the other hand, if the object class is determined to be correct, the process will proceed.
- **Prediction of the object:** the identity of the object will be shown.
- **Display frame of the object detection:** this process operates in real-time, guaranteeing seamless repetition to enable continuous monitoring and tracking of hawkers while achieving optimal accuracy of the results.

4.1 Dataset Creation

The data is gathered and compiled from various sources to create a comprehensive dataset specific to object detection. The collection process involves manual data collection via capturing photos using cameras, extraction of images from surveillance cameras, gathering the necessary information, and identifying relevant images from the internet. To obtain a comprehensive dataset for hawker detection, a total of 400 images have been collected through a combination of manual data collection, video recording, online sources, city council records, social media platforms, and various other relevant websites. This extensive and diverse collection approach was undertaken to ensure the availability of a sufficiently large and varied dataset, enabling accurate and effective detection of hawkers. The dataset images include various objects typically found in hawker settings, including umbrellas, tables, stools, cars, and people. In this project, these objects have been categorized into 5 distinct classes for classification purposes: 0: Umbrella; 1: Table; 2: Stool; 3: Car; 4: People.

By assigning unique numerical labels to each class, the classification model developed as part of this project can accurately identify and categorize objects observed in hawker environments based on their corresponding class numbers. Figure 3 shows some of the images in the dataset.

Fig. 3 - Dataset for Hawker detection

This pivotal step involves accurately assigning labels to objects such as umbrellas, tables, stools, cars, and people within the images present in the dataset. By assigning these labels, the objects can be precisely identified and analyzed based on their respective classes, enhancing the accuracy of computer vision tasks, particularly object recognition. Accurate labeling plays a vital role in the development of models that can effectively classify objects in diverse images, thus contributing to the advancement of computer vision technology. These processes of labeling provide specific details about the position and dimensions of each object in the image, labeled with their respective class object IDs. Such accurate labeling plays a vital role in various computer vision applications, including object detection, tracking, and classification. These labels contain essential details regarding the class of each hawker object,

Labeled datasets play a critical role in supervised learning as they serve as the foundation for training machine learning algorithms to discern patterns and generate accurate predictions. In the case of hawker object detection, the labeled dataset enables the algorithm to effectively identify and classify various hawker objects, including umbrellas, tables, stools, cars, and people, based on the provided labeling process. Figure 4 shows the labeling process. The process of labeling involves saving a text file with the name "Hawker.txt," as shown in Figure 5, matching the corresponding image. The text file should include the following content: ID: Unique identification number for the object, x-center: X-coordinate of the object's center, y-center: Y-coordinate of the object's center, width: width of the object and height: height of the object.

By adhering to this labeling guide, each object within the image can be accurately annotated with the necessary information, facilitating subsequent analysis and model training. Figure 6 shows the hawker image with its corresponding label in the text file. A total of 200 objects are labeled in this homemade dataset to train the mapping transformation neural network. From these objects, 700 data points are used as the training set, and 100 data points are used as the test set.

Fig. 4 - Dataset labeling process

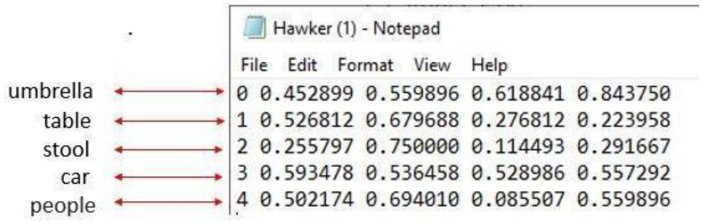


Fig. 5 - Example of the labeled dataset

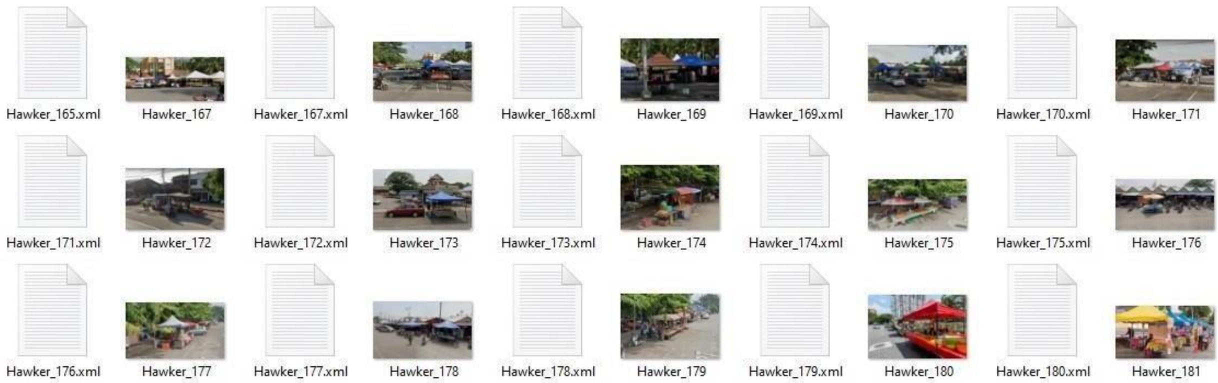
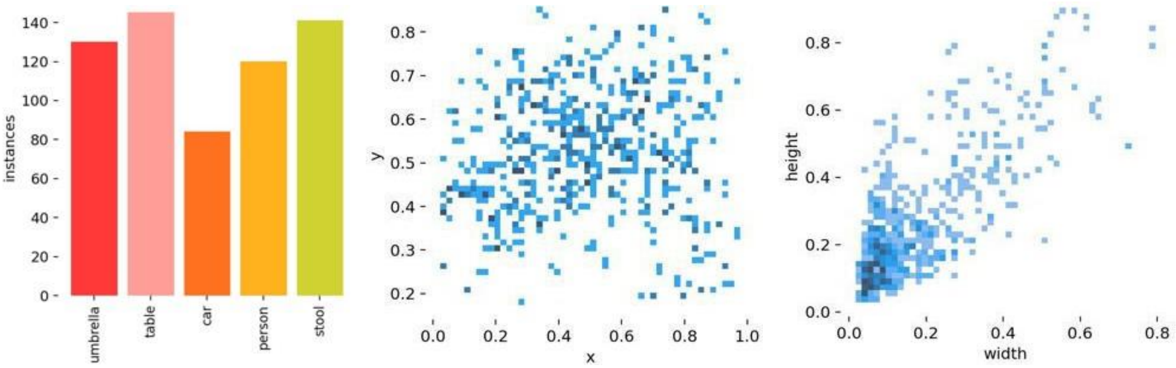


Fig. 6 - Dataset (image with the corresponding labeling)

4.2 Model Development and Training

The development, training, and deployment of the YOLOv5 Hawker Model was carried out effectively using Google Colab, resulting in accurate and efficient hawker object detection. The training was run from epoch 0 to epoch 99, with a total of 100 epochs. The purpose of specifying the number of epochs is to determine how many times the model will iterate over the entire training dataset. Increasing the number of epochs allows the model to learn from the data more extensively, potentially improving its performance. The training process took approximately 0.268 hours to complete, running for a total of 100 epochs. Once the training was completed, the trained model was obtained and saved. Based on the training result, the optimal setting for the YOLOv5 network is by setting the epochs to 80, the learning rate of 0.05, and 0.605 momentum. The dataset is trained by the improved YOLOv5, it can be observed that the improved YOLOv5 can converge quicker, and the convergence value is lower. Figure 7 (a) represents the number of instances detected for different object classes. The Y-axis represents the number of instances detected, ranging from 0 to 140, while the X-axis represents the five object classes: umbrella, table, car, person, and stool.

According to the graph, there were 130 instances of umbrellas detected, 150 instances of tables, 80 instances of cars, 120 instances of people, and 140 instances of stools. The graph allows us to compare the detection frequency across different object classes and gain insights into the distribution of detected instances among the classes. Likewise, Figure 7 (b) and (c) show the category distribution of detected objects. It can be seen that this dataset was almost equally distributed except for car since it is not usually present in the images, unlike person, umbrella, table, and stool. Table and stool account for most of the detected objects. The main reason is that the data source of this dataset is the environment of the hawker, with fewer cars, so most of the objects are umbrellas, stools, and people. Moreover, this dataset contains a large number of large-scale target detections, especially in the case of dense and overlapping targets, which makes it easier for the detection model to accurately identify objects. Finally, the number of samples in this dataset is relatively equal for each category, so the object detection network sufficiently learns the feature information of these samples, resulting in high sensitivity and recognition accuracy for these samples.



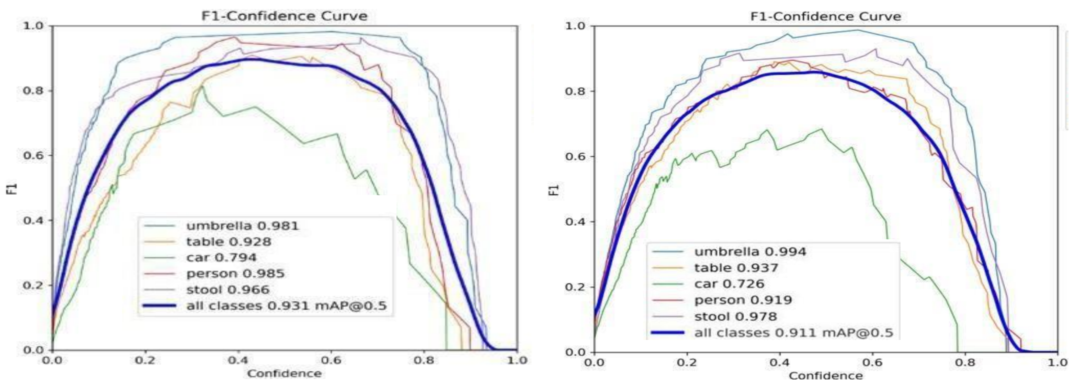
(a) Object category (b) Objects centre (c) Objects size

Fig. 7 - The visualization of the analysis of the hawker's dataset. (a) The distribution of detection object category and label. (b) the distribution of detection object center locations. (c) the distribution of detection object sizes

Figure 8 (a) and Figure 8 (b) show the detection accuracy of each class for 70% training (30% testing) and 80% training (20% testing), respectively. For the 70% training (30% testing), the precision-recall curve for the ‘umbrella’ class achieved a high value of 0.981, indicating excellent performance in accurately detecting ‘umbrella’ instances. The ‘table’ class also showed good performance with a precision-recall curve value of 0.928. The ‘car’ class had a relatively lower precision-recall curve value of 0.794. The ‘person’ class demonstrated exceptional performance with a precision-recall curve value of 0.985, indicating high precision and recall in detecting ‘person’ instances. Similarly, the ‘stool’ class performs well with a precision-recall curve value of 0.966. The overall mean average precision at Intersection over the Union (IoU) threshold of 0.5 (mAP@0.5) was 0.931, indicating satisfactory object detection performance across all classes.

Similarly, For the 80% training (20% testing), the precision-recall curve for the ‘umbrella’ class improved further to 0.994, indicating even higher precision-recall for ‘umbrella’ instances compared to the 70% training and 30% testing. The ‘table’ class also showed improvement with a precision-recall curve value of 0.937. The ‘car’ class had a lower precision-recall curve value of 0.726 compared to the 70% training and 30% testing. The ‘person’ class achieved a precision-recall curve value of 0.919, indicating slightly lower performance compared to the 70% training and 30% testing. The overall mAP@0.5 decreased to 0.911, indicating a slightly lower overall performance compared to the previous.

Increasing the training data from 70% to 80% resulted in improved performance for the ‘umbrella’ and ‘table’ classes, but a slight decrease in performance for the ‘car’ and ‘person’ classes. The ‘stool’ class maintained consistent performance across both. Overall, the model showed promising object detection capabilities, with high precision and recall values for most classes. However further analysis and optimization are required, particularly for the ‘car’ class, to enhance the model’s ability to accurately detect instances of that specific object. IoU is a common evaluation metric used in object detection tasks. It measures the overlap between the predicted bounding box and the ground truth bounding box. IoU is calculated as the ratio of the intersection area to the union area of the two bounding boxes. The IoU value ranges from 0 to 1, where 0 indicates no overlap and 1 indicates a perfect match between the predicted and ground truth bounding boxes.



(a)Accuracy for 70% training and 30% testing. (b) Accuracy for 80% training and 20% testing

Fig. 8 - Dataset (image with the corresponding labeling)

4.3 Real-time Hawkers Detection

Fig. 9 - Detection probability of each class

YOLO is a fast, accurate object detection, making it ideal for computer vision applications. The YOLO model can be integrated into a webcam/ CCTV/phone for real-time detection. Including fetching images from the display, the resulting system is interactive and engaging. While YOLO processes images individually, when attached to a webcam or CCTV it functions like a tracking system, detecting objects as they move around and changing in appearance. Figure 9 shows the detection and the detection probability of each class in each image.

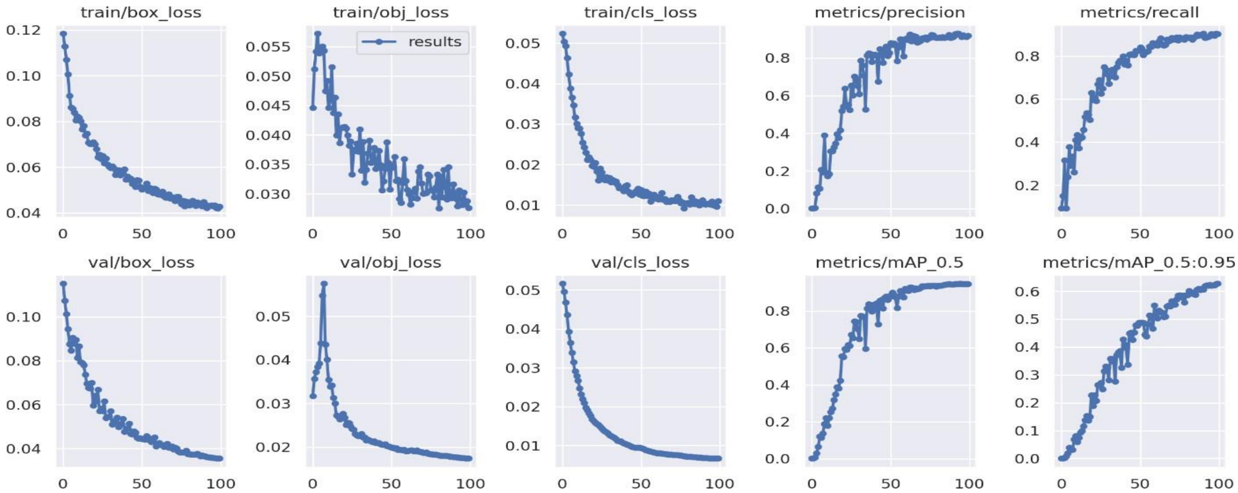


Fig. 10 - Result of the model

Figure 10 presents a comprehensive overview of various metrics and loss values obtained during the training process using Google Colab. The graph displays the train/box_loss, train/obj_loss, train/cls_loss, metrics/precision, metrics/recall, val/box_loss, val/obj_loss, val/cls_loss, metrics/mAP_0.5, and metrics/mAP_0.5:0.95. These metrics and loss values are crucial indicators of the model's performance and its ability to accurately detect and classify objects. By analyzing these results, we can gain insights into the model's training progress and assess its effectiveness in hawker object detection. There are three different types of loss shown in Figure 10 namely train/box_loss, train/obj_loss, and train/cls_loss. The train/box_loss (a.k.a. box loss during training) and val/box_loss (a.k.a. box loss during validation) represents how well the algorithm can locate the center of an object and how well the predicted bounding box covers an object. The train/obj_loss (a.k.a. objectiveness loss during training) and val/obj_loss (a.k.a. objectiveness loss during validation) are a measure of the probability that an object exists in a proposed region of interest. A high objectivity value suggests that the image window is likely to contain an object. Similarly, the train/cls_loss (a.k.a. classification loss during training) and val/cls_loss (a.k.a. classification loss during testing) represents how well the algorithm can predict the correct class of a given object. The model improved swiftly in terms of precision, recall, and mean average precision before plateauing after about 80 epochs. The box, objectness, and classification losses of the validation data also showed a rapid decline until around epoch 100. In addition, early stopping was used in order to select the best weights.

5. Conclusion

In this paper, the YOLOv5 machine learning algorithm was trained and provided real-time object detection in various applications such as detecting umbrellas, stools, tables, cars, and people classes. A custom dataset was gathered and augmented to obtain accuracy for each object taken. The model was trained using the YOLOv5 pre-trained weights for 100 epochs. Overall, all the objectives have been met, and the YOLOv5 series algorithm has proven effective in achieving real-time object detection across various applications, including the identification of specific objects related to hawker activities. The achievement of these objectives indicates the potential of the YOLOv5 algorithm for practical and real-world applications requiring fast and accurate object detection.

In conclusion, the objectives of this project have been successfully achieved. We have performed data collection of hawkers along the street in specific areas, allowing for a comprehensive understanding of hawker activities in urban environments. Additionally, we have developed a real-time object detection system specifically designed for hawker detection in smart city applications. This system enables efficient and accurate identification of hawkers, contributing to effective management and monitoring of urban street vendors. Overall, the achievement of these objectives enhances our understanding of hawker activities, facilitates more efficient urban management, and paves the way for further advancements in AI-based object detection systems. The outcomes of this project have significant implications for the development of smarter cities and contribute to the improvement of hawker identification and management processes. The strengths of this project lie in its ability to accurately detect and identify hawkers, leveraging advanced AI algorithms and techniques. This empowers smart community initiatives by enhancing safety, efficiency, and overall urban management. However, it is important to acknowledge the weaknesses and areas for improvement. One potential challenge is the complexity of the hawker environment, which may include crowded spaces, occlusions, or diverse object appearances. Ensuring robust detection and accurate classification under such conditions requires continuous refinement and optimization of AI models.

To further enhance this project, future work can explore incorporating additional contextual information, such as crowd density or hawker behavior analysis, to improve the overall system's capabilities. Additionally, considering the scalability and deployment challenges in real-world settings is crucial for achieving practical implementation and widespread adoption. By addressing these strengths and weaknesses, it can maximize the potential of AI-based analytics for hawker identification in smart community projects, contributing to safer, more efficient, and sustainable.

Acknowledgement

This research work was funded by the Ministry of Higher Education (MOHE) Malaysia under the Malaysian Technical University Network (MTUN) (Grant No.: 9028-00002/9002-00094) titled “Development and Deployment of Smart Community and Innovation Centre for Smart City 4.0” and Prototype Development Research Grant Scheme (Grant No.: PRGS/ 1/ 2022/ ICT04/ UNIMAP/ 02/ 1) titled “Upscaling and Field-testing of Integrated Flood Monitoring System with Visual Intelligent Dashboard and Data Analytics”.

References

Algiriyage, N., Prasanna, R., Doyle, E. E., Stock, K., Johnston, D., Punchihewa, M., & Jayawardhana, S. (2021, May). Towards Real-time Traffic Flow Estimation using YOLO and SORT from Surveillance Video Footage. In *Proceedings of the 18th International Conference on Information Systems for Crisis Response and Management, Blacksburg, VA, USA* (pp. 23-26).

Charitidis, P., Moschos, S., Pipertzis, A., Theologou, I. J., Michailidis, M., Doropoulos, S., ... & Vologiannidis, S. (2022). StreetScouting: A Deep Learning Platform for Automatic Detection and Geotagging of Urban Features from Street-Level Images. *Applied Sciences*, *13*(1), 266.

Gupta, A., Gupta, N., & Gupta, J. (2022). Machine Learning Based Prototype Working Model to Determine Category of Garbage Waste Using Raspberry Pi. *Res. Sq.*, (pp. 1–9)

Gupta, P. K., Verma, P. A., & Chauhan, P. (2019). Machine Learning Based Crop Cycle Mapping Using Multi Temporal Space Datasets. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, 557-561.

Huang, L., & Huang, W. (2022). Rd-yolo: An effective and efficient object detector for roadside perception system. *Sensors*, *22*(21), 8097.

Krišto, M., Ivasic-Kos, M., & Pobar, M. (2020). Thermal object detection in difficult weather conditions using YOLO. *IEEE access*, *8*, 125459-125476.

Liu, Y., & Liu, Y. (2022). Detecting the city-scale spatial pattern of the urban informal sector by using the street view images: A street vendor massive investigation case. *Cities*, *131*, 103959.

Martinez-Alpiste, I., Golcarenarenji, G., Wang, Q., & Alcaraz-Calero, J. M. (2022). Smartphone-based real-time object recognition architecture for portable and constrained systems. *Journal of Real-Time Image Processing*, *19*(1), 103-115.

Nkalubo, L. B., & Nakibuule, R. (2022). Review on Real-Time Object Detection Models using Deep Neural Networks. *Available at SSRN 4331602*.

Pawar, M., Dhanki, M., Parkar, S., Dandekar, C., & Gupta, B. (2019). A novel approach to detect crimes and assist law enforcement agency using deep learning with cctvs and drones. *International Journal for Research in Applied Science and Engineering Technology*, *7*, 653-662.

Qiu, Z., Zhao, Z., Chen, S., Zeng, J., Huang, Y., & Xiang, B. (2022). Application of an improved YOLOv5 algorithm in real-time detection of foreign objects by ground penetrating radar. *Remote Sensing*, *14*(8), 1895.

Sukkar, M., Kumar, D., & Sindha, J. (2021, July). Real-time pedestrians' detection by YOLOv5. In *2021 12th international conference on computing communication and networking technologies (ICCCNT)* (pp. 01-06). IEEE.

Wahab, F., Ullah, I., Shah, A., Khan, R. A., Choi, A., & Anwar, M. S. (2022). Design and implementation of real-time object detection system based on single-shoot detector and OpenCV. *Frontiers in Psychology*, *13*, 1039645.

Wang, K., Shi, F., Wang, W., Nan, Y., & Lian, S. (2019). Synthetic data generation and adaption for object detection in smart vending machines. *arXiv preprint arXiv:1904.12294*.

Wei, Y., Tran, S., Xu, S., Kang, B., & Springer, M. (2020). Deep learning for retail product recognition: Challenges and techniques. *Computational intelligence and neuroscience*, *2020*.