Revolutionizing Ripeness Detection with Roasted Shima Aji Fish and Deep Learning on Embedded Devices

Naphat Keawpibal[1], Ajaree Naco[2], Noppamas Pukkhem[3]

[1]Department of Computer and Information Technology
Faculty of Science, Thaksin University
Phatthalung, Thailand
naphat.k@tsu.ac.th
[2]Department of Computer and Information Technology
Faculty of Science, Thaksin University
Phatthalung, Thailand
ajaree@tsu.ac.th
[3]Department of Computer and Information Technology
Faculty of Science, Thaksin University
Phatthalung, Thailand
noppamas@tsu.ac.th

Abstract— Object detection is a fundamental concept in computer vision, which is widely recognized and has numerous applications in various domains, including agriculture, medicine, sports, and the food industry. The detection of ripeness in roasted fish is a particularly challenging task in the food processing industry. This study aims to compare the performance of two deep learning algorithms, YOLOv5s and MobileNetv2-SSD, for detecting the ripeness of roasted shima aji fish. The optimal model is selected for deployment on an embedded device. The dataset for this research comprises 689 images without data augmentation and 944 images with data augmentation of cut-open aji fish. The experimental results demonstrate that the YOLOv5s-DA model achieved the highest accuracy in detecting the ripeness of roasted shima aji fish, which is 98.30% of precision, and took approximately 30 minutes for model creation, outperforming MobileNetv2-SSD. Moreover, a prototype ripeness detection system for roasted shima aji fish is developed using a real-time camera and a Raspberry Pi 4, confirming the practical applicability of the proposed model.

Keywords-deep learning; ripeness detection; embedded device; roasted shima aji fish

I. Introduction

The seafood industry in the southern region of Thailand has encountered a range of obstacles resulting in a deceleration of production, notably escalating production costs and shortages of materials. However, processed frozen seafood products from Thailand are still in high demand, with many orders coming in from international markets. The southern region of Thailand has abundant marine resources, which means that there are relatively few problems with procuring raw materials for processing.

Shima aji fish, also known as *Selaroides leptolepis*, is a prevalent species in the Gulf of Thailand [1]. This type of fish is characterized by a long yellow stripe that runs from above the eye to the base of the tail. In the food processing industry, shima aji fish are commonly utilized for the production of dehydrated seafood, sweet fish, and roasted fish with sesame seeds, all of which offer high nutritional value and serve as an excellent source of calcium.

The traditional process for roasted shima aji fish involves using a conveyor system with grill machine, which is operated manually by three employees. The degree of ripeness of the fish depends on its size, the roasting temperature, and the length of time it spends on the machine. After the fish is roasted, the employees visually inspect the fish to grade its doneness. They do this by examining the color and burn marks on the fish. The fish is categorized into three groups based on its level of ripeness: unripe, ripe, and overripe. Fish that is properly ripe will be processed into further various products. If the fish is unripe, it will be returned to the machine and roasted for a shorter time by adjusting the speed of the conveyor belt while maintaining the same temperature. If the fish is burnt, it must be discarded. Despite the use of the machine, human labors are still necessary to visually grade the doneness of the fish. However, this can be problematic, as the color of ripe and burnt fish can be similar, making it easy to make mistakes during the sorting process.

The conventional roasting process heavily depends on manual labor, which is susceptible to errors during the sorting process, as discussed previously. As a solution, alternative approaches such as utilizing deep learning algorithms for ripeness detection could potentially offer a more efficient and precise classification of fish ripeness.

Several studies have explored object detection using deep learning techniques, which have been continuously evolving and applied in various domains [2-4]. To the best of our knowledge, no existing research has investigated the application of deep learning techniques for classifying the level of ripeness of roasted shima aji fish based on images.

In this paper, we propose an approach for ripeness detection of roasted shima aji fish using deep learning algorithms. The ripeness level of grilled shima aji fish can be classified into three categories, namely unripe, ripe, and overripe, which are visually determined by expert employees. Therefore, we compare and evaluate the performance of two individual object detectors, YOLOv5s and MobileNet-SSDv2, in terms of precision and execution time. The best-performing model is selected and implemented on an embedded device.

The structure of this paper consists of several sections that present the research findings in an organized manner. In Section 2, the related works in deep learning algorithms and research on object detection are discussed. Section 3 provides an overview of the architecture of the ripeness detection system for roasted shima aji fish, as well as the process steps of the deep learning algorithm. Section 4 presents the results of the evaluation of the deep learning algorithms and their deployment on an embedded board. Finally, in Section 5, the major findings are summarized.

II. LITERATURE REVIEW AND RELATED WORKS

In this section, two deep learning algorithms, YOLOv5 and MobileNet-SSDv2, are described and discuss related works on object detection using deep learning.

A. YOLOv5

YOLO stands for You Only Look Once, which is a state-of-the-art, real-time object detector and well-known algorithm that is used in various research domains [5]. YOLOv5 is the improvement version based on YOLOv1 – YOLOv4. One was the YOLOv4 developed by the conventional authors Joseph Redmon and Alexey Bochkovskiy [6], while other being the released YOLOv5 by Glenn Jocher [7]. The continuous improvements have made it the better performance on two well-known object detection datasets: Pascal VOC (visual object classes) [8] and Microsoft COCO (common objects in context) [9]. Moreover, one of the notable improvements in YOLOv5 is its adoption of the Python programming language, which simplifies the process of installing and integrating the algorithm with IoT devices. Additionally, PyTorch boasts a sizable and active community, which suggests that the platform will receive more contributions and experience continued growth in the future.

The YOLOv5 network is mainly composed of three main components: [2]

1. Backbone – Uses CSPDarknet that a CNN layer aggregate image features at different scales.
2. Neck – Uses PANet that a set of layers to combine image features and pass them to prediction.
3. Head – Takes features from the neck and performs localization and classification.

Fig. 1 shows an overview of YOLOv5 architecture. The common task of all object detection architecture involves extracting input image features in the backbone component and subsequently forwarding them to the object detector, which includes the detection neck and detection head. The detection neck, also referred to as neck, performs as a feature aggregation to combine and blends features to prepare for the forthcoming step. The detection head is responsible for detecting, classifying, and localizing for each bounding box.

Since this paper is focused on real-time detection, speed is a factor of utmost importance, hence the version has been chosen as the representative of the YOLOv5s family for its performance analysis.

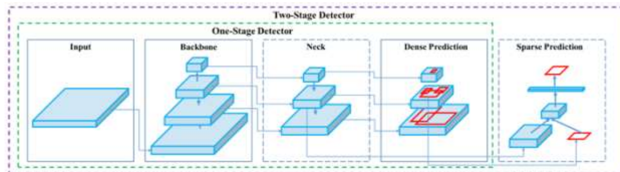

Figure 1. An Overview of YOLOv5 architecture [6].

B. MobileNet-SSDv2

MobileNet [10] is a computer vision model developed by Google, optimized for high speed, and designed for use in mobile and embedded applications. In particular, MobileNetv2 features an inverted residual structure, which offers improved modularity by removing non-linear layers and resulting in cutting-edge performance. This lightweight model is ideal for resource-constrained environments that require efficient processing. However, this lightweight design may come at a cost of lower accuracy predictions.

To address this issue, Google released Single Shot Detector (SSD), which is designed to achieve high accuracy in object detection tasks while maintaining fast processing speeds. SSD can detect multiple objects in an image using a single shot. By integrating SSD with MobileNet, the resulting model achieves superior performance and is referred to as MobileNet-SSD. This hybrid model takes advantage of MobileNet's speed and SSD's accuracy to provide a balanced solution for applications that require both speed and accuracy.

For the performance analysis of the application presented in this paper, MobileNetv2 integrated with SSD has been used.

C. Related Works

As previously mentioned, object detection has been extensively studied and applied in various domains. This section discusses related works on object detection using deep learning algorithms.

Many studies related to object detection in the field of agriculture have utilized image processing and deep learning techniques for object detection. Particularly, one study focused on developing a quality classification model for Chok-Anan mangoes using image processing technology with human vision [11]. The collected data was categorized into 4 different quality grades: Grade A, B, C, and D. Grade D represented rotten mangoes. The study demonstrated that the use of deep learning methods with convolutional neural network (CNN) algorithms achieved the highest accuracy of 99.79%.

The development of an automated system for analyzing sweetness and watermelon varieties by using deep learning techniques on smartphone was proposed [12]. This research used deep learning neural network via the Tensorflow library with Inceptionv3 and MobileNet algorithms to classify 4 varieties of watermelon with 100 images. The results showed that both algorithms have the same accuracy of 97.20%, but the model size obtained from MobileNet is smaller. Therefore, the Model form MobileNet was chosen to develop on Android device as a mobile application to classify watermelon varieties and identify the sweetness level.

Another research proposed a computer vision-based application using convolutional neural networks (CNNs) for the classification of mulberry fruit ripening stages [13]. This paper compared the accuracy performance of 5 deep learning algorithms, including DenseNet, Inception-v3, ResNet-18, ResNet-50, and AlexNet based on the black and white mulberries of 1,000 images. The results showed that the AlexNet and ResNet-18 networks achieved the best performance with 98.32% and 98.65% overall accuracy for classify the ripeness of white and black mulberries, respectively.

Deep learning algorithms, such as YOLOv5, have been increasingly utilized in a variety of domains, including safety detection [14], garbage classification [15], and mask detection [16]. With image, video, and real-time surveillance, classification and object detection have become critical research areas in recent years, where deep learning algorithms such as YOLOv3, YOLOv4, YOLOv5, and MobileNetv2-SSD have been employed [3, 16]. However, it should be noted that YOLOv4 model has slower execution times during the model building process, particularly when the number of training images is significantly increased [16]. Therefore, in this study, two deep learning algorithms, YOLOv5 and MobileNetv2-SSD, were selected to overcome this challenge.

III. The Proposed Method

This section describes an overview of the proposed method, AjiRipeDet, as shown in Fig. 2. The explanations consist of the processes of ripeness detection of shima aji fish from data collection to performance evaluation by comparing between YOLOv5s and MobileNetv2-SSD deep learning algorithms. Furthermore, the outperform model among them will be selected and implement on an embedded device to validate the performance in a practical environment.

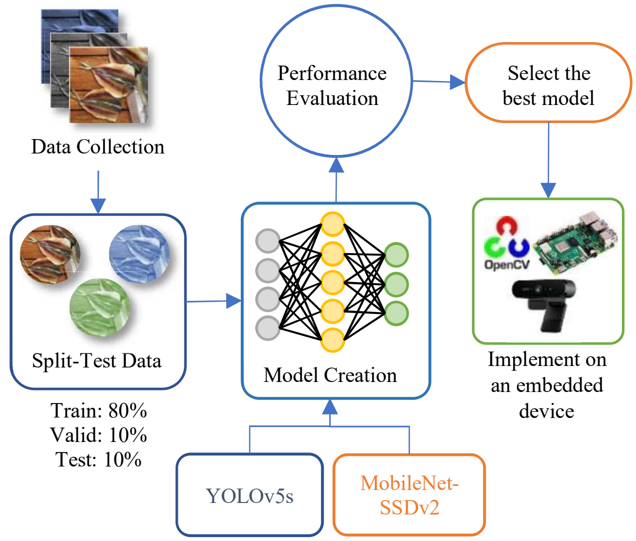

Figure 2. An Overview of the proposed method.

A. Data Collection and Preparation

The images of roasted shima aji fish were collected from PlaNeat Co., Ltd. in the Hat Yai District of Songkhla Province. A total of 689 images were gathered, which were divided into three categories based on their level of ripeness: unripe, ripe, and overripe. The ripeness level of each fish was determined by three experienced employees. The use of experienced employees to sort the images according to ripeness level also ensures the accuracy and reliability of the dataset. Specifically, the dataset included 393 unripe images, 281 ripe images, and 15 overripe images, as shown in Table 1.

Due to the limited number of overripe shima aji fish images in the original dataset, the authors applied data augmentation techniques to increase the number of images. Specifically, Roboflow website was used to perform operations such as image resizing, rotation, and blurring, resulting in 270 additional overripe images. The data augmentation process is a commonly used method to increase the size and diversity of image datasets, which can improve the robustness and generalization of deep learning models. The resulting dataset contained a total of 944 images, including 393 unripe images, 281 ripe images, and 270 overripe images generated through data augmentation. The allocation of images to each subset is presented in Table 2. The examples of roasted shima aji fish images and their class label are shown in Fig. 3.

The images dataset, which includes unripe, ripe, and overripe shima aji fish images, was divided into three subsets using the Split-Test technique. Specifically, 80% of the images were used for the training dataset, while 10% were allocated for both the validation and testing datasets. The use of the Split-Test technique is a standard practice in deep learning for evaluating model performance and preventing overfitting. The training dataset is used to optimize the model parameters, while the validation dataset is used to tune the model hyperparameters. Finally, the testing dataset is used to evaluate the model's generalization performance on unseen data. The allocation of images to each subset is presented in Table 1 and Table 2 for before data augmentation and after data augmentation, respectively.

TABLE I. NUMBER OF ROASTED FISH USED WITHOUT DATA AUGMENTATION

Ripeness level	#Training dataset	#Validation dataset	#Testing dataset	#Total data
Unripe	315	39	39	393
Ripe	225	28	28	281
Overripe	11	2	2	15
Total	551	69	69	689

TABLE II. NUMBER OF ROASTED FISH USED IN WITH DATA AUGMENTATION

Ripeness level	#Training dataset	#Validation dataset	#Testing dataset	#Total data
Unripe	315	39	39	393
Ripe	225	28	28	281
Overripe	216	27	27	270
Total	756	94	94	944

Figure 3. The example of original images of roasted shima aji fish with the class label.

B. Model Creation

To create model, the training of the YOLOv5s and MobileNet-SSDv2 models was performed using a Google Cloud platform instance equipped with an Nvidia A100-SXM4 GPU with 40 GB of RAM. Four training epochs were considered, including 200, 300, 400, and 500 epochs. Due to the dataset's size, which contained almost 1,000 images, a batch size of 16 was utilized during the training process.

As previously mentioned, a split-test approach was employed to divide the dataset into training, validation, and testing subsets. Specifically, 80% of the images were assigned to the training dataset, while 10% were assigned to both the validation and testing datasets. Therefore, 94 images were used for validation, and an additional 94 images were allocated for testing purposes. Fig. 4 illustrates the process steps for model creation and measurement metrics to evaluate model performance.

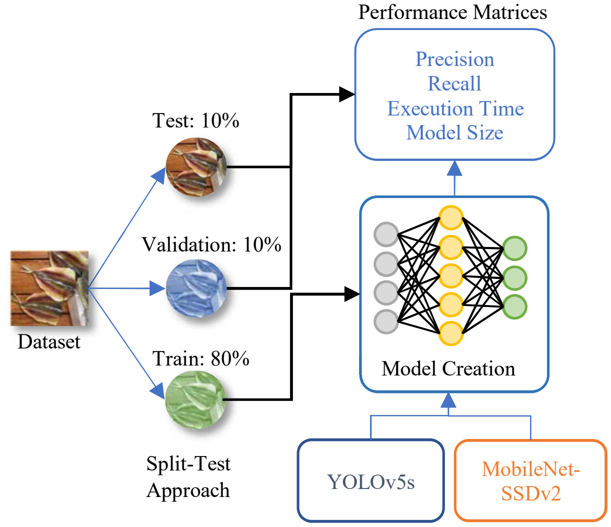

Figure 4. The process steps for model creation.

C. Performance Evaluation and Model Selection

After training the deep learning models, their performance was evaluated with different epoch values by comparing two algorithms, YOLOv5 and MobileNet-SSDv2, with data augmentation and without data augmentation. Several performance metrics were measured including mean average precision, recall, execution time, and model size. The goal was to select the most efficient model to further develop on an embedded device.

The comparison of the performance metrics for each algorithm, along with different epoch values, enabled the identification of the optimal configuration with the highest accuracy and the lowest execution time and model size. This is a crucial step for deploying the model on an embedded device with limited computational resources. Therefore, the model with the best trade-off between accuracy, execution time, and model size was selected as the final candidate for deployment on the embedded device.

D. Implementation on An Embedded Device

Following the selection of the most efficient model, the chosen model was further developed for deployment on an embedded device. For this work, the Raspberry Pi 4 board was selected as the device, and the OpenCV library was used to enable the real-time detection of the ripeness level of roasted shima aji fish through the camera.

IV. EXPERIMENTAL RESULTS

A comparison of two models was carried out after the completion of the training process with different number of epochs. Those models were trained on Google Cloud platform with Nvidia A100-SXM4 GPU with 40 GB of RAM and the batch size of 16 is set. The performance of the trained models is presented in Table 3 and Table 4 for the model trained without data augmentation and for the model trained with data augmentation, respectively.

TABLE III. A COMPARATIVE PERFORMANCE WITHOUT DATA AUGMENTATION

Model	Epoch	mAP (%)	Recall (%)	Execution time (min.)	Model size (MB)
YOLOv5s	200	98.20	100.00	14.02	14.9
	300	96.90	99.50	20.43	
	400	**98.50**	**100.00**	**27.10**	
	500	97.60	100.00	34.00	
MobileNet-SSDv2	200	61.40	67.40	41.27	6.45
	300	89.78	91.50	61.20	
	400	**92.04**	**93.58**	**80.85**	
	500	89.47	90.38	100.71	

TABLE IV. A COMPARATIVE PERFORMANCE WITH DATA AUGMENTATION

Model	Epoch	mAP (%)	Recall (%)	Execution time (min.)	Model size (MB)
YOLOv5s-DA	200	96.30	98.50	22.38	14.9
	300	**98.30**	**98.80**	**33.36**	
	400	98.40	98.80	44.46	
	500	96.00	100.00	55.32	
MobileNet-SSDv2-DA	200	91.32	93.37	58.25	6.45
	300	**91.59**	**93.91**	**86.97**	
	400	89.99	92.78	114.78	
	500	91.56	93.81	143.15	

The study findings suggest that the precision of ripeness detection is not significantly improved by increasing the number of epochs, particularly for both YOLOv5s and YOLOv5s-DA models. The optimal models for precision, recall, and execution time were YOLOv5s and YOLOv5s-DA, as demonstrated in the tables. Despite the smaller model size of both MobileNet-SSDv2 and MobileNet-SSDv2-DA, their prediction accuracy fell short and necessitated a longer model building time. Furthermore, the implementation of data augmentation resulted in a more effective model with respect to the overall performance metrics, thereby mitigating the overfitting issue.

As depicted in Figure 5, a performance comparison was conducted between YOLOv5-DA and MobileNet-SSDv2-DA models. The results confirmed that the model generated by YOLOv5-DA exhibited greater accuracy and faster processing speed than that generated by MobileNet-SSDv2-DA. In real-time applications, speed is a crucial factor, but the model's accuracy also plays a vital role in ensuring successful functioning. Therefore, to achieve optimal performance, the YOLOv5s-DA model is an ideal choice for prediction, despite requiring computation-intensive hardware and memory. The selected model was tailored to the hardware specifications of the Raspberry Pi 4 board, necessitating code optimization for the ARM architecture, and ensuring the model's operability within the device's memory constraints.

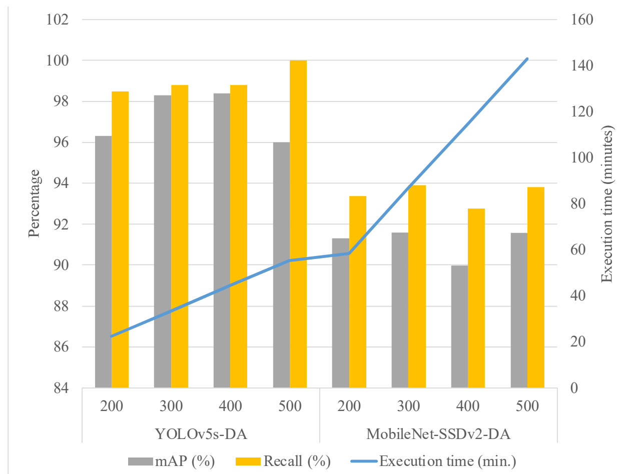

Figure 5. The comparison performance between YOLOv5s-DA and MobileNet-SSDv2-DA

Following the development and experimentation of a model for classifying the level of ripeness of roasted shima aji fish using images, the YOLOv5s model was selected for ripeness detection. The model was then developed on the Raspberry Pi 4 board, utilizing a camera and OpenCV library. Fig. 6 depicts the connection between the Raspberry Pi board and the camera through the USB port, enabling the real-time detection of the ripeness level of the roasted fish.

Figure 6. The design of connecting Raspberry Pi 4 to the camera.

In this study, the efficacy of utilizing the OpenCV library on the Raspberry Pi 4 board for the ripeness detection of roasted shima aji fish was investigated, and the results are presented in Fig. 7. The findings suggest that the system can effectively classify the ripeness level of the roasted fish. Nonetheless, it is important to note that some inaccuracies were observed in detecting the ripeness level of the roasted shima aji fish. This may be attributed to the darkening of the roasted fish sample's color, which occurred several hours after its removal from the machine. To address this issue, the collected and prepared training data should include various time periods to capture the changing color of the roasted fish sample.

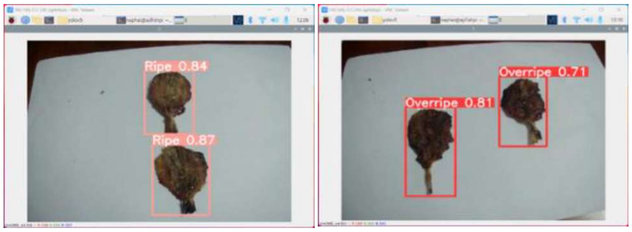

Figure 7. The result of ripeness detection on Raspberry Pi 4 board.

V. CONCLUSION

A deep learning algorithm was utilized to develop a ripeness detection model for roasted shima aji fish, which was then deployed on an embedded device. Data of roasted shima aji fish was collected and prepared from PlaNeat Co., Ltd. resulting in a dataset of 689 labeled images without data augmentation and 944 labeled images with data augmentation. The ripeness level of the fish was classified into three categories: unripe, ripe, and overripe. In our experiment, two deep learning algorithms, YOLOv5-based and MobileNet-SSDv2-based, were compared to classify the ripeness level of the roasted fish images. The dataset was divided into training, validation, and testing subsets. The results showed that YOLOv5-based outperformed MobileNet-SSDv2-based in accuracy and computation time but had a larger model size. The model of YOLOv5-based with data augmentation, YOLOv5-DA, was selected as the optimal model for real-time usage and evaluated accordingly.

A prototype system for ripeness detection of roasted shima aji fish on an embedded device was developed using Python on the Raspberry Pi 4 board and a real-time camera. The system successfully classified and detected the ripeness level of roasted shima aji fish, with some errors in detecting the level due to the color of the roasted fish sample changing over time. Future work will involve testing and evaluating the prototype system in a real environment, with the aim of integrating it into the industry to reduce the potential for human error.

ACKNOWLEDGEMENT

This work was financially supported by Thaksin University Research Fund. The authors would like to thank PlaNeat Co., Ltd. for helping the location of the image collection and testing the prototype of the ripeness detection model for the roasted fish.

REFERENCES

[1] Bunluedaj C., Noranarttragoon P., Boonjorn N, et.al, Stock Assessment of Yellow stripe scad (*Selaroides leptolepis* (Cuvier, 1833)) in the Gulf of Thailand, Technical Report, 2016.

[2] Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G., & Wang, W. (2020, December). Face Mask Recognition System with YOLOV5 Based on Image Recognition. In 2020 IEEE 6th International Conference on Computer and Communications (ICCC) (pp. 1398-1404). IEEE.

[3] Iyer, R., Ringe, P. S., & Bhensdadiya, K. P. (2021). Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for real-time mask detection. Artic. Int. J. Res. Eng. Technol, 8, 1156-1160.

[4] Karthi, M., Muthulakshmi, V., Priscilla, R., Praveen, P., & Vanisri, K. (2021, September). Evolution of yolo-v5 algorithm for object detection: automated detection of library books and performace validation of dataset. In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-6). IEEE.

[5] Redmon, J., and Ali F. (2018). "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767.

[6] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[7] Jocher, G., Stoken, A., Borovec, J., Changyu, L., & Hogan, A. (2020). ultralytics/yolov5: v3. 0. Zenodo.

[8] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111, 98-136.

[9] Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference on Computer Cision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.

[10] Chiu, Yu-Chen, et al. "Mobilenet-SSDv2: An improved object detection model for embedded systems." 2020 International Conference on System Science and Engineering (ICSSE). IEEE, 2020.

[11] Pattansarn, N., & Sriwiboon, N. (2020). Image processing for classifying the quality of the Chok-Anan mango by simulating the human vision using deep learning. Journal of Information Science and Technology, 10(1), 24-29.

[12] Hongboonmee, N., & Jantawong, N. (2020). Apply of Deep Learning Techniques to Measure the Sweetness Level of Watermelon via Smartphone. Journal of Information Science and Technology, 10(2), 59-69.

[13] Ashtiani, S. H. M., Javanmardi, S., Jahanbanifard, M., Martynenko, A., & Verbeek, F. J. (2021). Detection of mulberry ripeness stages using deep learning models. IEEE Access, 9, 100380-100394.

[14] M. Zhou, J. Zhu and X. Li, "Safety helmet detection system of smart construction site based on YOLOv5S," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 2022, pp. 1223-1228, doi: 10.1109/ICSP54964.2022.9778524.

[15] Z. Wu et al., "Using YOLOv5 for Garbage Classification," 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Yibin, China, 2021, pp. 35-38, doi: 10.1109/PRAI53619.2021.9550790.

[16] F. Muding, A. Moolman and N. Keawpibal. Real-time Wearing Face Mask Detection with Deep Learning Algorithm. In: The 13th National Science Research Conference. Phatthalung, Thailand, 12 – 13 May 2022, pp. 847-856.