Efficient Video Object Detection for EV Charging Plug Type Identification in Real-World Scenarios

Phasuwut Chunnapiya
Department of Computer and Information Science
Faculty of Applied Science
KMUTNB
Bangkok, Thailand
s6404062856048@email.kmutnb.ac.th

Porawat Visutsak
Department of Computer and Information Science
Faculty of Applied Science
KMUTNB
Bangkok, Thailand
https://orcid.org/0000-0003-2120-5585

***Abstract*—The rapid growth of electric vehicles (EVs) necessitates a robust charging infrastructure, but the diversity of charging plug types poses a challenge for both EV users and automated charging systems. This paper addresses the accurate identification of EV charging plug types in real-world scenarios using video object detection. We evaluated four state-of-the-art YOLO models (V5s, V6s, V7, and V8s) on a custom dataset. YOLOv8s demonstrates superior overall accuracy (mAP@0.5:0.95 of 0.95, precision 0.998, F1-score 0.997), while YOLOv7 excels in specific metrics (mAP@0.5 of 0.997, recall 0.997). YOLOv6s boasts the fastest training time. YOLOv5s has the lowest Gigaflops and Parameter. We bridge research and application with a user-friendly website for real-time EV socket detection, empowering users and paving the way for automated charging. This research contributes to a more accessible and efficient EV charging ecosystem, fostering sustainable transportation.**

Keywords— Electric Vehicle, Object Detection, YOLO, Charging Plug, EV Station

I. Introduction

The rapid growth of electric vehicles (EVs) necessitates a robust charging infrastructure. However, the diversity of charging plug types poses a challenge for EV users and automated charging systems. This paper addresses the problem of accurately identifying EV charging plug types in real-world scenarios using video object detection. We investigate the performance of four state-of-the-art YOLO models (V5s, V6s, V7, and V8s) on a custom dataset of EV charging sockets. Our findings reveal that YOLOv8s exhibits superior overall accuracy, while YOLOv7 demonstrates exceptional performance in specific metrics.

To bridge the gap between research and real-world application, we developed a user-friendly website that leverages the trained models for real-time EV socket detection. This system empowers EV users to effortlessly identify compatible charging stations, and it lays the groundwork for future advancements in automated charging technologies. By contributing to a more accessible and efficient EV charging ecosystem, this research supports the widespread adoption of sustainable transportation solutions.

The subsequent sections of this paper are organized as follows: The Literature Review provides a comprehensive overview of relevant concepts and prior research in video object detection. The Materials and Methods section details the research design, data collection instruments, and analysis techniques employed in the study. The Results and Discussion section presents the findings regarding the performance analysis of our method. Finally, the Conclusion summarizes the key findings, discusses their implications, and offers recommendations for future research and practice in the field of video object detection in the real world.

II. Literature Review

YOLO is a one-stage object detection algorithm developed by Joseph Redmon and his team. It is a one-stage detector capable of detecting multiple objects simultaneously. YOLO utilizes a single neural network to process input images and produce bounding box outputs. Each bounding box includes a confidence score for each class, allowing for rapid object detection. However, this speed comes at the expense of lower accuracy compared to two-stage object detection approaches[1], [2], [3].

In January 2024, YOLO will be divided into 8 versions, including YOLO, YOLO V2, YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8. Each version has been upgraded and features different architectures [4].

This research focuses on comparing the performance of YOLO V5s, YOLO V6s, YOLO V7, and YOLO V8s for the specific task of identifying electric vehicle charging plugs in Thailand. These versions represent a significant advancement in the YOLO architecture, offering improved speed and accuracy compared to earlier iterations.

2.1 YOLO V5

YOLO V5s, introduced in 2020, is an open-source project implemented in Python. It utilizes CSPDarknet53 as its backbone and incorporates the Feature Pyramid Network (FPN) and Pixel Aggregation Network (PAN) structures in its neck for enhanced feature fusion [5].

YOLO V5 offers a range of algorithms suitable for various use cases, from small to large models. Each model varies in speed and accuracy. The architecture of YOLO V5 consists of 3 main components: 1) Backbone, 2) Neck, and 3) Head.

2.1.1 Backbone

In the backbone, CSPDarknet53 is used as Backbone. CSPDarknet53 consists of CBS modules, C3 modules, SPPF module. CBS modules will help the C3 module with feature extraction and include Conv + BatchNorm + SiLU. SPPF modules which improve the performance of feature extraction networks through various convolution and pooling operations of different sizes.

979-8-3315-0991-0/24/$31.00 ©2024 IEEE

2.1.2 Neck

In the neck, YOLO V5 utilizes the FPN structure and the PAN structure. FPN propagates semantic information from top to bottom through upsampling operations, while PAN propagates location information from bottom to top through downsampling operations. By merging these two structures, the network can incorporate more details, forming a feature fusion module that operates across various scales to preserve both large and small target feature details.

2.1.3 Head

In the head, The bounding box utilizes the CIOU Loss function [6], and Non-Maximum Suppression (NMS) is utilized to filter out multiple object boxes and produce the predicted image. Moreover, the head in YOLO V5 resembles that of YOLO V4 and YOLO V3, generating three different outputs of feature maps to enable multi-scale prediction [7].

2.2 YOLO V6

YOLO V6, also introduced in 2022, is designed for industrial applications, prioritizing efficiency across diverse hardware platforms. It employs EfficientRep as its backbone and Rep-PAN as its neck, offering a balance between speed and accuracy.

There are several versions of YOLO V6, starting with YOLO V6-nano which is the fastest. Uses minimal parameters, all the way to YOLO V6-large with high accuracy in exchange for speed [8].

The architecture of YOLO V6 consists of three components: 1) Backbone, 2) Neck, and 3) Head [9], [10].

2.2.1 Backbone

In the backbone, We will use EfficientRep as the Backbone. EfficientRep consists of three parts: a) RepBlock, b) RepConvBlock, and c) CPSStackRep Block

a) Rep Block is designed for small-sized models and consists of RepVGG and ReLU activations during training.

b) RepConvBlock is a RepVGG block converted to RepConv during inference.

c) CPSStackRep Block is designed for medium and large models.

2.2.2 Neck

In the neck, Rep-PAN used as the Neck, which is an improved version of the PAN topology used in YOLO V4 and YOLO V5. Additionally, it replaces the CSPBlock used in YOLO V5 with either RepBlock (for small models) or CSPStackRep Block (for large models).

2.2.3 Head

In the head, It will be used similarly to YOLO V5 for both regression and classification.

2.3 YOLO V7

YOLO V7 introduced in June 2022, builds upon YOLO V4, Scaled YOLO V4, and YOLO-R. It boasts improved real-time object detection accuracy without increasing inference costs. Its architecture includes an Efficient Layer Aggregation Network (ELAN) and Maximum Pooling Convolution (MPConv) in its backbone, and RepConv in its neck.

The architecture of YOLO V7 consists of three components: 1) Backbone, 2) Neck, and 3) Head [7], [8], [9], [11].

2.3.1 Backbone

In the backbone, Backbone consists of multiple convolutions involving an Efficient Layer Aggregation Network (ELAN) module and a Maximum Pooling Convolution (MPConv) module. In the MPConv module, the MaxPool operation expands the receptive field of the current feature layer. Subsequently, it integrates this information with the feature data obtained after the regular convolution process, thereby enhancing the network's generalization capabilities. In the SPPCSPC module incorporates multiple MaxPool operations in parallel with convolution to mitigate distortion induced by image manipulation processes. Initially, the input image undergoes feature extraction within the Backbone, yielding three suitable feature layers for subsequent network assembly.

2.3.2 Neck

In the neck, RepConv is used to design a heavily parameterized convolutional architecture that provides more gradient diversity for feature maps at different scales.

2.3.3 Head

In the head, In the head, It will be used similarly to YOLO V5 for both regression and classification.

2.4 YOLO V8

YOLO V8 released in January 2023, supports various vision tasks, and features an architecture like YOLO V5. Notable changes include replacing the CSPLayer with the C2f module, transitioning to Anchor-Free detection, and reducing the convolutional layer's kernel size.

The architecture of YOLO V8 closely resembles that of YOLO V5, with changes in modules, layers, and other components, including: 1) Replacing the CSPLayer with the C2f module. 2) Transitioning from Anchor-Based to Anchor-Free detection. 3) Reducing the convolutional layer's kernel size from 6x6 to 3x3.

Architecture of YOLO V8 consists of 3 main parts: 1) Backbone, 2) Neck, and 3) Head [8], [9], [12].

2.4.1 Backbone

In the backbone, CSPDarknet53 have used as the Backbone, with a change in the module. The CSPLayer will be replaced with the C2f module, where C2f stands for “cross-stage partial bottleneck with two convolutions”. The role of the C2f module is to integrate high-level features with contextual information to enhance detection accuracy.

2.4.2 Neck

In the neck, it will be similar to YOLO V5, but with the mentioned modifications in the Neck architecture.

2.4.3 Head

In the head, it will be similar to YOLO V5, but with the mentioned modifications in the Head architecture.

2.5 Our preliminary work

In [13], our previous work used YOLO V7 for video object detection specifically designed for indoor furniture and home appliances. We also introduce a novel frame sampling technique to optimize processing time. The proposed method was evaluated in a real-world environment and showed significant improvements compared to the standard YOLOv7 baseline. It achieved a 24.73% increase in mean average precision (mAP50-95), a key metric for object detection accuracy, and an 89.77% reduction in processing time. These results highlight the effectiveness of the proposed approach for efficient and accurate video object detection in the context of indoor furniture and home appliances, which can be valuable for applications like room layout planning and inventory management.

III. Materials and Methods

3.1 Dataset

This study focuses on video object detection of Socket EV. We aim to train YOLO models (YOLO v5s, YOLO v6s, YOLO v7, and YOLO v8s) with augmentation techniques such Flipped, rotating, scaling, cropping, ShiftScaleRotate, ElasticTransform, OpticalDistortion, GridDistortion, zooming, rotating, and shearing, blur processing, noise processing, brightened augmentation, Gaussian blurred, and motion blurred, to increase the number of images in the dataset. Our dataset consists of five classes of Socket EV, totaling 109,902 images. In Augmentation, we use the albumentations library for Augmentation. We specifically searched the Internet for image data from publicly available sources. Our search included sites such as Roboflow [14], Google, Youtube, online marketplaces, and others.[15] The five classes of Socket EV used in this study include: CHAdeMO; AC type 1; AC type 2; DC CCS type 1; DC CCS type 2; Each image in the dataset has been manually labeled and classified into its corresponding socket type. The dataset is divided into training, validation, and testing sets in an 8:1:1 ratio, respectively. Figure 1 shows some examples of our dataset.

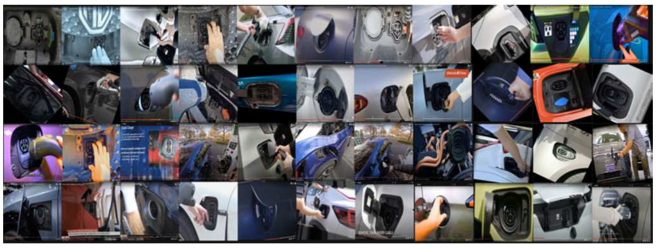

Fig. 1. Socket EV dataset [15].

3.2 Real-world test video

In this experiment, we recorded video clips at different charging stations to study real-world conditions. We aimed to capture various situations, such as rain, bright sunlight, nighttime parking lots, and dark environments. We filmed at two major events in Thailand: the Bangkok International Motor Show 2024 at Impact Muang Thong Thani on April 1, 2024, and the Bangkok EV Expo at the Queen Sirikit National Convention Center on February 11, 2024. The goal was to test how well the models work in different conditions. Figure 2 shows video clips from the charging stations, while Figure 3 displays footage from the Bangkok International Motor Show 2024. Figure 2 A shows scenario outdoor parking , Figure 2 B shows scenario nighttime parking, and Figure 2 C shows scenario rain.

Fig. 2. Video clips taken at actual locations at various charging stations.

Fig 3. Video clips taken at Thailand at the Bangkok International Motor Show 2024 at Impact Muang Thong Thani on April 1, 2024.

3.3 Experimental setting

We used YOLO V5s, YOLO V6s, YOLO V7, and YOLO V8s in the experiments. We used the same parameters and environmental data in all experiments. We describe the environment in Section 3.3.1, and the parameters used in Section 3.3.2

3.3.1 *Experimental Environment and Language Used in the Experiment*

Our experiment utilized three environments: Environment A, B and C. Environment A was used for training, validation, and prediction, while Environment B and C was dedicated solely to prediction.

Details of Environment A include Python 3.10.12 and PyTorch 2.10+cu118, featuring an NVIDIA RTX A5000 GPU with 24,564 MB of CUDA memory, and an AMD EPYC 7302P CPU with 16 cores and 31.3 GB of RAM. In contrast, Environment B also uses the same versions of Python and PyTorch but runs on a MacBook Pro Max 2022 with an Apple M2 Max GPU featuring 38 cores and an Apple M2 Max CPU with 12 cores, along with 32 GB of RAM. And Environment C use iPhone 12 Promax including IOS 18.0.

3.3.2 *Parameters*

In the training and validation of our models, we used the same parameters for all four models (YOLO V6s, and YOLO V8s). We set the number of epochs to 100, with a batch size of 32 and an image size of 640. The training was conducted in Environment A using the Dataset Socket EV, and we utilized 8 workers for data loading. The initial learning rate (lr0) was set to 0.01, with a learning rate factor (lrf) of 0.01 for YOLO V5s, V6s, and V8s, and the same for YOLO V7. The optimizer used was SGD, with a momentum of 0.937 and a weight decay of 0.0005. Table 1 shows values Gigaflops and Parameters.

TABLE 1. GFLOPS and Parameters

Model	Gigaflops	Parameters
YOLO V5s	16.0 Gigaflops	7,033,114
YOLO V6s	44.9 Gigaflops	16,452,192
YOLO V7	105.2 Gigaflops	37,218,132
YOLO V8s	28.7 Gigaflops	11,137,535

3.4 Website development

We have developed a comprehensive website using Next.js as the framework, integrating the "@tensorflow/tfjs" library to facilitate predictions. By implementing a rendering approach known as CSR (Client-Side Rendering), we enable prediction processing to occur directly on the client side. This method effectively utilizes the user's device resources, significantly reducing the burden on the server while providing a smoother and more responsive user experience.

To enhance the functionality of our application, we specifically chose the YOLO V6s and YOLO V8s models. These models are designed to deliver accurate and efficient results, allowing us to effectively meet the diverse needs of our users. The ability to perform precise detections is crucial, especially in applications involving real-time analysis and decision-making.

The "@tensorflow/tfjs" library, developed by Google, serves as a powerful tool for implementing machine learning capabilities in JavaScript. Its compatibility with CSR makes it an ideal choice for our project. In our setup, the server is responsible for sending the necessary models and libraries to the client. Once received, the client can process and render web pages independently. This architecture facilitates complete management of content and presentation, allowing for dynamic data changes without the need to reload the entire page. As a result, users experience a more seamless and efficient interaction with the application.

For deployment, we utilized specific environments tailored for optimal performance. The website runs on Node.js version 18.17, with Next.js version 14.2.7 and @tensorflow/tfjs version 4.21.0. Each component was chosen to ensure compatibility and enhance the overall efficiency of the application.

For those interested in exploring our work further, the complete codebase can be found in our GitHub repository titled _Thesis_Detecting-the-type-of-socket-electric-vehicle-charger-in-Thailand [16]. This repository contains all the necessary resources and documentation for understanding and replicating our project.

Figures 4 and 5 show the interface of the developed website. Image 4 A) is the homepage, where users can choose between two models for prediction: YOLO V6 and YOLO V8. After selecting a model, it appears in Image 4 B), which has four buttons: “Change Model,” “Open Image,” “Open Video,” and “Open Webcam.”

Clicking these buttons takes users to different options. The “Open Image” button lets users choose an image for prediction, the “Open Video” button allows them to select a video, and the “Open Webcam” button opens the webcam for live predictions. The results from these actions are shown in Figure 5 A, B, and C, making the website easy and friendly to use.

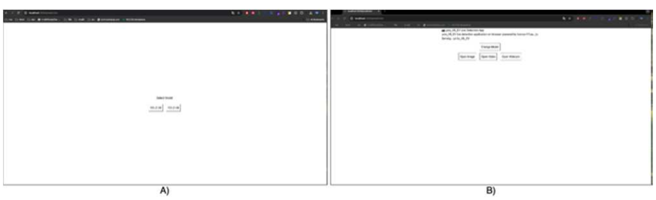

Figure 4. The developed website capture image.

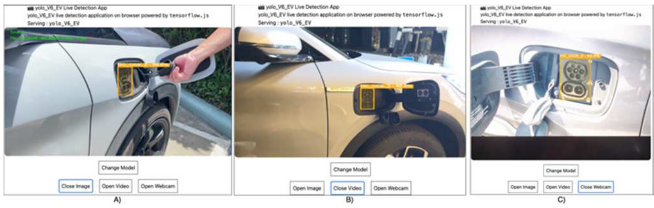

Figure 5. The capture website of the detection.

IV. RESULTS AND DISCUSSION

4.1 Detection results

This paper presents a comparative analysis of four object detection algorithms—YOLO V5s, YOLO V6s, YOLO V7, and YOLO V8s—to evaluate their efficacy in detecting EV charging sockets. The models underwent rigorous training, and their performance was assessed using metrics such as accuracy, recall, mAP50, mAP50-95, and Gigaflops. The results of this evaluation are detailed in Table 2. Furthermore, we conducted a comparative analysis with the work presented in [17], where a YOLO V8s model was proposed for video object detection tasks.

TABLE 2. Detection results

Metric	YOLO V8s in our proposed method	YOLO V8s model employed by [17]
mAP50	0.995	0.928
mAP50-95	0.95	0.745
Precision	0.998	0.947
Recall	0.996	0.889
Inference time on the training dataset	N/A	100.6 ms
Detection rate	N/A	97.23%

The results presented in Table 2 highlight the superior performance of our YOLO models, particularly YOLO V8s, in comparison to the YOLO V8s model employed by [17]. Our models consistently demonstrate higher precision, recall, and mAP values, indicating greater accuracy in detecting EV charging sockets.

Specifically, YOLO V8s achieves a mAP@.5 of 0.995 and a mAP@.5-.95 of 0.95, significantly surpassing the 0.928 and 0.745 reported by [17]. This suggests that our model not only excels at detecting objects with a standard IoU threshold of 0.5 but also maintains high accuracy across a range of IoU thresholds, implying superior object localization capabilities.

Furthermore, our models exhibit consistently higher precision and recall values, with YOLO V8s achieving a precision of 0.998 and a recall of 0.996, compared to 0.947 and 0.889, respectively, in [17]. This translates to fewer false positives and false negatives, crucial for real-world EV charging applications.

While a direct comparison of inference time and detection rate is not feasible due to the absence of this data for our models, the overall performance advantage in terms of mAP, precision, and recall strongly suggests that our YOLO models, especially YOLO V8s, offer a more accurate and effective solution for detecting EV charging sockets.

YOLOv5s has the lowest Gigaflops and Parameters value among all YOLO. Gigaflops refers to the number of mathematical operations required for the model to complete one iteration. These metrics increase, resulting in higher inference speed: the lower the inference speed, the faster the results [18].

TABLE 3. Experimental results for all four videos

Video	All video situations	YOLO V6s			YOLO V8s		
		AVG of Ground Truth	**Number of wrong guesses**	**Number of correct guesses**	**AVG of Ground Truth**	**Number of wrong guesses**	**Number of correct guesses**
1	- Parking at night - Light rain	84.61111111	0	10	89.22	0	10
2	- Parking at night	84.96	0	11	89.56	0	11
3	- Parking at night	88.308	2	25	86.85	2	25
4	- Outdoor parking	87.67	1	27	85.82	0	28
5	- Outdoor parking	89.60	1	23	89.7	0	24
6	- Heavy rain - plug far away	75.35	4	5	65.8	4	5

4.2 Detection EV Socket on website

To evaluate the real-world performance of the YOLO V6s and YOLO V8s models trained for EV socket detection, we captured videos featuring EV sockets as detection targets. The objective was to assess the effectiveness of both models and establish consistent confidence thresholds.

We conducted experiments using six video clips. Videos 1-5 were recorded at various charging stations, while Video 6 was sourced from YouTube. The experimental setup is illustrated in Figures 6 to 11, and the results of all six videos are shown in Table 3. In Table 3, we summarize the detection results, including the average detection rate, the number of incorrect predictions, and the number of correct predictions. All results can be found in the file [19].

From the comparison between the predictions of YOLO V8s and YOLO V6s in Environment B and C, the results shown in Figures 12 and 13, as well as Table 4, indicate that YOLO V8s performs better than YOLO V6s, with a difference in accuracy of 3.49% for Environment B and 3.31% for Environment C. Additionally, the comparison of prediction differences between the two environments shows a variation of 0.54% for Environment B and 0.37% for Environment C, demonstrating the model's stability across different environments.

Figure 6. The results of detection EV Socket of video 1.

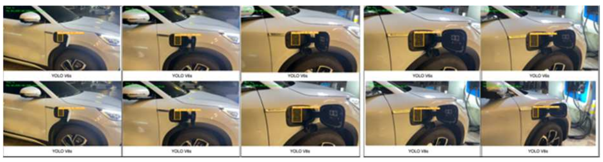

Figure 7. The results of detection EV Socket of video 2.

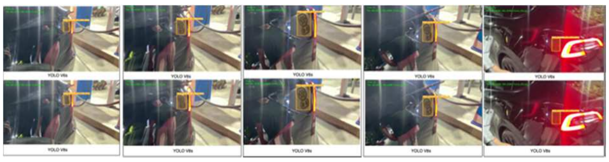

Figure 8. The results of detection EV Socket of video 3.

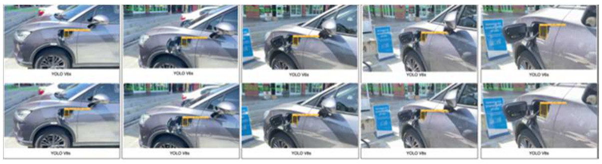

Figure 9. The results of detection EV Socket of video 4.

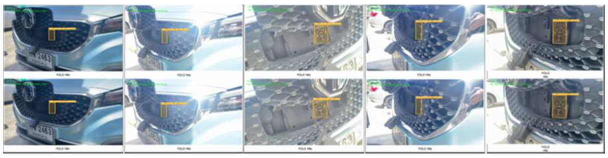

Figure 10. The results of detection EV Socket of video 5.

Figure 11. The results of detection EV Socket of video 6.

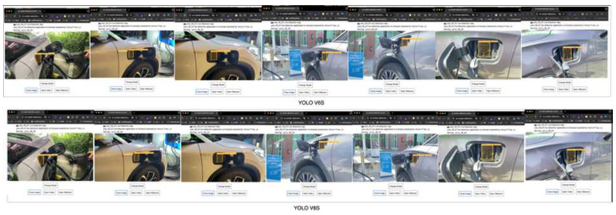

Figure 12. The results of detection EV Socket of Environment B.

Figure 13. The results of detection EV Socket of Environment C.

TABLE 4. Detection results of Environment B and Environment C

File name	Second	Environment B		Environment C	
		YOLO V6s	YOLO V8s	YOLO V6s	YOLO V8s
IMG_6279	0	85.1	90.1	86.1	90.9
IMG_6283	7	85.8	90.6	86.1	91
IMG_6283	8	86.7	91.8	87.2	91.6
IMG_6292	21	86.9	90.3	87.7	90.9
IMG_6292	20	87.1	89.7	88	90
IMG_6292	35	90.8	93.1	90.9	93.7
IMG_6292	33	92.1	93.3	92.3	93.4
AVG		87.79	91.27	88.33	91.64

V. CONCLUSION

In this study, we investigate YOLO-based video object detection models for extracting information from Real-world test video to assist with Robot charger automatic. Our findings suggest that YOLOv8s might be the most accurate model overall, achieving the highest mAP@0.5:0.95 (0.95), indicating excellent precision in bounding box predictions across various overlap thresholds, achieving the highest Precisions (0.998) and F1 score (0.997) However, YOLOv7 remains a strong contender due to its impressive mAP@0.5 (0.997).

YOLOv5s has the lowest Gigaflops value, along with Precision (0.996), Recall (0.996), mAP@0.5 (0.995), mAP@0.95 (0.912), and F1 Score (0.996), while YOLOv7 has the highest Gigaflops and parameters. Comparing YOLOv5s with YOLOv7, YOLOv7 has a higher mAP@0.95 value than YOLOv5s by 0.038 and a higher Gigaflops value than YOLOv5s by 6.575 times. The Precision, Recall, mAP@0.5, and F1 Score of YOLOv5s are also close to YOLOv7. Therefore, YOLOv5s is suitable for use in resource-constrained devices as it has lower Gigaflops than YOLOv7 while still having similar Precision, Recall, mAP@0.5, and F1 Score.

In our real-world test, we capture a video clip at a real location to check our model's performance, as shown in Figures 6 to 11. We also created a website that makes predictions using the computing power of different devices. The results show that the system can detect EV sockets in real time, with an average accuracy of 85.08% based on the ground truth.

The code and dataset used in this research are available on the author's GitHub repository: "phasuwut/_Thesis_ Detecting-the-type-of-socket-electric-vehicle-charger-in-Thailand".

REFERENCES

[1] F.-F. Li, A. Karpathy, and J. Johnson, "Spatial Localization and Detection," 2016.

[2] F.-F. Li, Y. Li, and R. Gao, "Lecture 6: CNN Architectures".

[3] N. Chalitapanukul and W. Wongseri, "Automated steel surface visual inspection system," KMUTNB, Bangkok, 2022.

[4] A. Aggarwal, V. Kumar, and R. Gupta, "Object Detection Based Approaches in Image Classification: A Brief Overview," in 2023 IEEE Guwahati Subsection Conference (GCON), Guwahati, India: IEEE, Jun. 2023, pp. 1–6. doi: 10.1109/GCON58516.2023.10183609.

[5] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022. Accessed: May 28, 2024. [Online]. Available: http://arxiv.org/abs/2209.02976

[6] R. Arifando, S. Eto, and C. Wada, "Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses," Appl. Sci., vol. 13, no. 9, p. 5802, May 2023, doi: 10.3390/app13095802.

[7] O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, "A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms," J. Comput. Soc. Inform., vol. 2, no. 1, pp. 1–12, Feb. 2023, doi: 10.33736/jcsi.5070.2023.

[8] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," Machines, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.

[9] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.

[10] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022. Accessed: Jul. 12, 2024. [Online]. Available: http://arxiv.org/abs/2209.02976

[11] Y. Sun, Y. Li, S. Li, Z. Duan, H. Ning, and Y. Zhang, "PBA-YOLOv7: An Object Detection Method Based on an Improved YOLOv7 Network," Appl. Sci., vol. 13, no. 18, p. 10436, Sep. 2023, doi: 10.3390/app131810436.

[12] S. Li, H. Huang, X. Meng, M. Wang, Y. Li, and L. Xie, "A Glove-wearing Detection Algorithm Based on Improved YOLOv8." Nov. 02, 2023. doi: 10.20944/preprints202311.0111.v1.

[13] F. Pensiri, P. Chunnapiya, W. Khamprapai, and P. Visutsak, 'Efficient Video Object Detection of Indoor Furniture and Home Appliances', in 2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2024, pp. 336–339, doi: 10.1109/BlackSeaCom61746.2024.10646241.

[14] "Roboflow: Computer vision tools for developers and enterises Accessed: Jul. 12, 2024. [Online]. Available: https://roboflow.com/

[15] https://github.com/phasuwut/_Thesis_Detecting-the-type-of-socket-electric-vehicle-charger-in-Thailand/blob/main/File_reference/%5BDataset%5D%20List%20source.xlsx [Online]

[16] https://github.com/phasuwut/_Thesis_Detecting-the-type-of-socket-electric-vehicle-charger-in-Thailand [Online]

[17] V. Tadic, A. Odry, Z. Vizvari, Z. Kiraly, I. Felde, and P. Odry, "Electric Vehicle Charging Socket Detection using YOLOv8s Model," *Acta Polytech. Hung.*, vol. 21, no. 10, pp. 121–139, 2024, doi: 10.12700/APH.21.10.2024.10.8.

[18] V. C. Mahaadevan, R. Narayanamoorthi, R. Gono, and P. Moldrik, 'Automatic Identifier of Socket for Electrical Vehicles Using SWIN-Transformer and SimAM Attention Mechanism-Based EVS YOLO', IEEE Access, vol. 11, pp. 111238–111254, 2023.

[19] https://github.com/phasuwut/_Thesis_Detecting-the-type-of-socket-electric-vehicle-charger-in-Thailand/blob/226dd1baa9109e23265208f98c884e6bd4e4ed66/File_reference/(thesis)%20Prediction%20Score.xlsx [Online]