A Single-Stage Deep Learning-based Approach for Real-Time License Plate Recognition in Smart Parking System

Lina YU[1]*, Shaokun LIU
Hebei College of Industry and Technology, Hebei Shijiazhuang, 050091, China

***Abstract*—License plate recognition in smart parking systems plays a crucial role in enhancing parking management efficiency and security. Traditional methods and deep learning-based approaches have been explored for license plate recognition. Deep learning methods have gained prominence due to their ability to extract meaningful features and achieve high accuracy rates. However, existing deep learning-based fire detection methods face challenges in terms of accuracy, real-time requirement, and computation cost, as evident from previous studies. To address these challenges, we propose a single-stage deep learning approach using YOLO (You Only Look Once) algorithm. Our method involves generating a custom dataset and conducting training, validation, and testing processes to train the YOLO-based model. Experimental results and performance evaluations demonstrate that our proposed method achieves high accuracy rates and satisfies real-time requirements, validating its effectiveness for license plate recognition in smart parking systems.**

Keywords*—*Smart parking; license plate recognition; deep learning; single-stage detector; Yolo

I. Introduction

Smart parking systems have emerged as a transformative solution to address the challenges of urban parking management, aiming to optimize parking resource utilization and enhance the overall parking experience for drivers [1, 2]. These systems leverage advanced technologies to provide real-time parking information, streamline parking processes, and contribute to efficient traffic management in smart cities [3].

License plate recognition (LPR) technology plays a pivotal role in smart parking systems by enabling automated vehicle identification, entry/exit control, and payment processes [4, 5]. LPR involves the detection, extraction, and recognition of license plate information from images or videos. By accurately capturing and processing license plate data, smart parking systems can provide seamless and convenient parking experiences to drivers, optimize resource allocation, and enhance operational efficiency.

Current technologies and recent advances in license plate recognition have significantly enhanced the capabilities of smart parking systems. Various approaches, including computer vision, machine learning, and deep learning, have been employed to improve the performance of LPR systems [6]. Among these technologies, vision-based methods have garnered significant attention from researchers due to their ability to handle variations in lighting conditions, vehicle types, and license plate designs.

Vision-based methods utilize computer vision techniques to extract and analyze license plate information from images or videos [7]. These methods have shown promising results in terms of accuracy and reliability, making them suitable for real-world deployment in smart cities. The significance of research in this area lies in developing more accurate and efficient LPR systems that can handle complex real-world scenarios and meet the real-time requirements of smart parking systems [8]. On the other hand, deep learning-based methods have emerged as state-of-the-art approaches for license plate recognition in smart parking systems. Deep learning models, particularly those utilizing convolutional neural networks (CNNs), have shown remarkable advancements in various computer vision tasks [9, 10]. However, there are still limitations and challenges in existing deep learning-based approaches for license plate recognition, particularly in achieving real-time processing and high accuracy rates.

To tackle these limitations, further research is needed to investigate alternative approaches that can achieve high accuracy rates in real-time. One such approach is the utilization of single-stage deep learning models, such as the YOLO (You Only Look Once) algorithm [11]. Single-stage models eliminate the need for time-consuming region proposal networks and achieve efficient end-to-end license plate detection and recognition [12]. By exploring single-stage and YOLO-based methods, the research aims to address the current limitations and challenges in deep learning-based license plate recognition for smart parking systems.

In this study, we propose a YOLO-based method with a custom dataset generation to tackle the identified research gap and meet the real-time requirements of license plate recognition in smart parking systems. The YOLO-based model is generated using a custom dataset, encompassing various challenging scenarios encountered in smart parking environments. The model is trained, validated, and tested to evaluate its performance in terms of accuracy, speed, and robustness.

The research contributions of this study lie in identifying the research gap in deep learning-based license plate recognition for smart parking systems and proposing a YOLO-based method to address this gap. Additionally, comprehensive experimental evaluations are conducted to validate the proposed method, assessing its performance in real-world scenarios. The contributions of this research are significant for the development of accurate and efficient license plate recognition systems, ultimately improving the efficiency and convenience of smart parking management in urban environments.

The main research contributions in this study are listed as follows:

- The study identifies the existing limitations in real-time license plate recognition for smart parking systems and highlights the need for more efficient and accurate methods to address these challenges.
- Generating an extensive custom dataset for car license plate recognition.
- The study proposes a YOLO-based approach with a custom dataset generation to improve license plate detection and recognition in smart parking systems, offering an effective solution for real-time processing and enhanced accuracy.
- The study conducts comprehensive experimental evaluations to validate the proposed method, assessing its performance in terms of accuracy, speed, and robustness. The results demonstrate the effectiveness and superiority of the YOLO-based approach in addressing the research gap and meeting the requirements of smart parking systems.

II. Previous Studies Review

The authors in [4] focuses on parking entrance control using license plate detection and recognition. The method employs computer vision techniques to detect and recognize license plates of vehicles entering a parking area. The advantages of this approach include enhanced security and improved efficiency in managing parking access. By automating the entrance control process, it reduces the need for manual intervention and enables real-time monitoring. However, potential drawbacks may include challenges in accurately detecting and recognizing license plates under varying lighting conditions and vehicle speeds. Overall, this research presents a practical solution for parking entrance control, offering benefits in terms of security and efficiency while acknowledging the need for robust license plate detection and recognition algorithms.

In [13], license plate recognition algorithms are explored based on deep learning in complex environments. The proposed method utilizes deep learning techniques to address challenges such as varying lighting conditions, occlusions, and plate deformations. The advantages of this approach include achieving high accuracy rates and robustness in complex scenarios, thanks to the superior feature extraction and pattern recognition capabilities of deep learning algorithms. However, drawbacks include the need for large amounts of labeled data for training, computational complexity, and potential limitations in generalizing to different license plate formats and languages. Overall, this research highlights the effectiveness of deep learning in license plate recognition but emphasizes the importance of considering data requirements, computational complexity, and adaptability to practical implementation.

The authors in [14] proposes a license plate recognition system that utilizes YOLOv5 and CNN (Convolutional Neural Network) models. The method involves training the YOLOv5 model to detect license plates in images, followed by using a CNN model for character recognition. The advantages of this approach include the ability to accurately locate and recognize license plates in real-time, enabling efficient automation of tasks such as parking management and access control. The use of deep learning models like YOLOv5 and CNN allows for robust performance even in challenging scenarios. However, potential drawbacks may include the need for a large amount of labeled data for training the models and the computational complexity associated with deep learning algorithms. Overall, this research demonstrates the effectiveness of using YOLOv5 and CNN for license plate recognition, offering advantages in terms of accuracy and real-time processing while acknowledging the considerations related to data requirements and computational resources.

The authors in [15] developed a low-cost IoT-based Arabic license plate recognition model for smart parking systems. The method utilizes existing IoT infrastructure and image processing algorithms to automate the recognition of Arabic license plates, enhancing parking management efficiency. The advantages include its cost-effectiveness and focus on Arabic license plates, catering to regions with Arabic as the dominant language. However, limitations include reliance on IoT infrastructure and the restriction to Arabic license plates. Challenges such as connectivity issues and varying plate formats may affect accuracy.

The authors in [17] presented an all-encompassing automated license plate recognition (ALPR) system, which streamlines the entire process by employing the YOLO (You Only Look Once) algorithm for vehicle and license plate detection, coupled with vehicle classification. The proposed approach represents a comprehensive solution to license plate recognition, aiming to eliminate the need for multiple separate algorithms and enhance the automation of this crucial task in various applications. However, the study brings to light a significant concern regarding the system's low accuracy rate in certain scenarios. This limitation suggests that the system may encounter challenges in reliably recognizing license plates, particularly in complex or adverse conditions. Consequently, it underscores the necessity for further research and improvements in accuracy to ensure the system's robust performance across diverse real-world scenarios and applications.

The study [18] explored a convolutional neural network (CNN)-based approach for license plate detection. The proposed method in the study centers around the utilization of CNNs, a class of deep learning algorithms, to automatically detect license plates in images. The authors conduct a comprehensive investigation to assess the effectiveness of this approach. Their findings indicate that the CNN-based method demonstrates promise in the realm of license plate detection, providing notable advantages in terms of speed and efficiency compared to traditional methods. However, a critical limitation that emerges from their research is a relatively low accuracy rate. This limitation is particularly evident in challenging scenarios, such as low-light conditions or when license plates exhibit variations in size, angle, or perspective. The low accuracy rate underscores the need for further refinement and optimization of the CNN-based approach to improve its robustness and reliability, ensuring accurate license plate detection across diverse real-world situations and enhancing its potential applications in fields such as automated surveillance and transportation systems.

The study [19] presented a License Plate Recognition System (LPRS) utilizing an improved YOLOv5 (You Only Look Once) architecture in conjunction with a GRU (Gated Recurrent Unit) network. The proposed method aims to enhance the accuracy and efficiency of license plate recognition, a vital task with applications in security and traffic management. Through a comprehensive exploration of the proposed approach, the study reveals that the improved YOLOv5 and GRU-based LPRS system exhibits promising potential in recognizing license plates from images and videos. However, a significant limitation is identified, notably a lower accuracy rate in challenging conditions and scenarios, such as low lighting or extreme angles. This limitation underscores the need for further refinements and enhancements to address these challenges and improve the overall accuracy and robustness of the system, as accurate license plate recognition is crucial for various real-world applications, including surveillance and automated toll collection systems.

III. MATERIAL AND METHOD

A. Dataset Generation

In this study, a custom dataset is generated for license plate recognition. For this dataset, following steps are performed.

1) Collect images: In this study, for collecting license plate images from internet resources to generate a custom dataset, several steps are performed. Firstly, we identify reliable sources that provide license plate images, including open data repositories and publicly available datasets. Secondly, systematically search and retrieve license plate images from these sources using specific keywords or filters with image databases that contain labeled license plate images. Thirdly, verify the authenticity and quality of the collected images to ensure they are suitable for dataset creation for image resolution, clarity, and visibility of the license plates. Finally, we consider verifying the accuracy and reliability of the labeled license plate information.

2) Image annotation: In collected images, there are several images with no labeling which are required to label and annotate. Annotating unlabeled license plate images for our custom dataset involves manually adding annotations to identify and localize license plate regions. The annotator uses specialized software to draw bounding boxes around the license plates and assign labels representing alphanumeric characters. Accuracy and attention to detail are crucial in ensuring precise annotations. Consistent annotation guidelines and regular quality checks maintain dataset quality. Annotated license plate images enable the training and evaluation of accurate and robust license plate recognition models for real-world applications.

3) Data augmentation: To cover more variations in images and extending the dataset, we performed data augmentation. Augmenting license plate images is performed to expanding the custom dataset by applying various transformations and modifications to the existing images. This augmentation technique enhances the dataset's diversity and variability, making the trained model more robust and capable of handling different scenarios. Common augmentation techniques include rotation, translation, scaling, flipping, adding noise, changing lighting conditions, and introducing occlusions. By applying these transformations, the dataset can capture variations in license plate positions, angles, sizes, and image characteristics, such as brightness and noise levels. Augmentation helps to mitigate overfitting and improve the generalization ability of the license plate recognition model. It also enables the model to better handle real-world challenges, such as variations in weather, lighting conditions, and vehicle orientations. Therefore, we generated 8642 images with above steps in our custom dataset.

B. One-Stage Detector

One-stage detector algorithms are a type of object detection algorithm used in computer vision tasks. Unlike two-stage detectors that involve a separate region proposal network (RPN) and object detection stage, one-stage detectors perform both object localization and classification in a single pass. One popular example of a one-stage detector algorithm is the YOLO (You Only Look Once) algorithm. This algorithm enables to process the entire image in a single pass makes it efficient and suitable for real-time applications like license plate recognition in smart parking systems. Fig. 1 [16] shows the structure of one-stage detector.

In a one-stage detector algorithm, the input to the algorithm is an image that contains objects of interest, such as vehicles with license plates in the case of license plate recognition. The algorithm processes the entire image at once and produces a set of bounding boxes along with corresponding class predictions for each detected object. This is in contrast to two-stage detectors that first propose regions of interest and then classify those regions.

The backbone network is a crucial component of the one-stage detector algorithm. It is typically a deep convolutional neural network (CNN) that extracts hierarchical features from the input image. The backbone network can be a popular CNN architecture like ResNet, VGG, or Darknet, which has been pre-trained on a large-scale image classification task. The purpose of the backbone network is to capture high-level representations and semantic information from the image.

The neck network typically consists of additional convolutional layers that fuse and combine the multi-scale features from the backbone network. This fusion helps to enhance the representation power of the features and enables the algorithm to detect objects at different scales and resolutions.

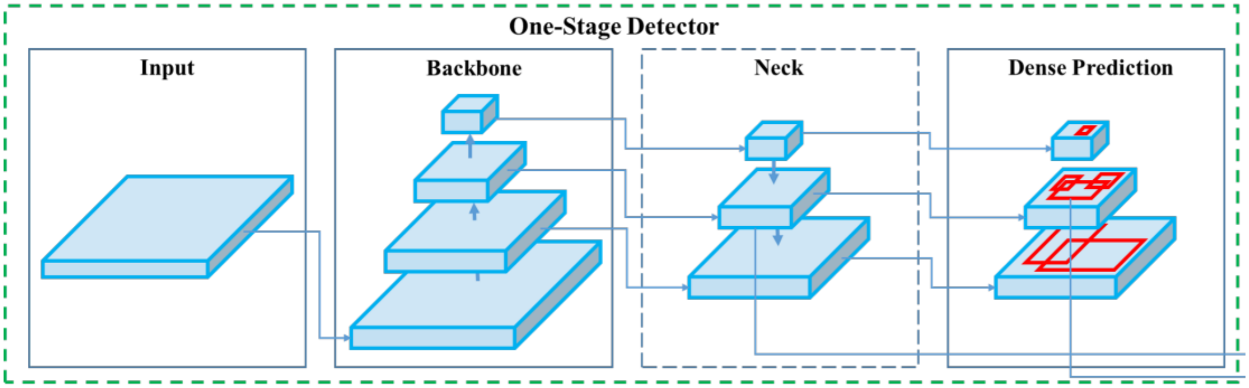

Fig. 1. Structure of one-stage detector.

The head or dense prediction part of the one-stage detector algorithm performs the final stage of object detection. It consists of convolutional layers followed by fully connected layers. The head network takes the fused features from the neck network as input and performs spatial and channel-wise convolutions to generate the final detection outputs. These outputs include the bounding box coordinates (x, y, width, height) and class probabilities for each detected object. The head network utilizes anchor boxes or default boxes to anchor the predicted bounding boxes to predefined scales and aspect ratios [16].

C. Model Generation

Generating a YOLOv5 model for license plate recognition on a custom dataset involves several steps, including dataset preparation, training, validation, and testing. Firstly, the custom dataset is divided into training, validation, and testing sets with the typical split of 70%, 20%, and 10%, respectively. The training set, comprising 70% of the dataset, is used to train the YOLOv5 model. The validation set, consisting of 20% of the dataset, is utilized for monitoring the model's performance during training and tuning hyperparameters. Lastly, the testing set, which makes up 10% of the dataset, is used to evaluate the final model's performance.

1) Training: In the training module, the YOLOv5 model is trained on the training set using the labeled license plate images. This is typically achieved through an iterative process using techniques such as stochastic gradient descent (SGD) or adaptive optimization algorithms. During training, the YOLOv5 model learns to detect and localize license plates in images and predict the corresponding alphanumeric characters. In training a YOLOv5 model for license plate recognition, several loss curves are commonly monitored during the training process: train/box_loss, train/obj_loss, and train/cls_loss. Fig. 2 shows training loss curves.

As shown in Fig. 2, the train/box_loss curve represents the loss associated with the bounding box predictions. The model aims to accurately predict the coordinates and dimensions of the bounding boxes around license plates. During training, the box loss gradually decreases as the model learns to better localize and fit the bounding boxes around the license plates. A lower box loss indicates improved precision and accuracy in the model's ability to detect and locate license plates.

The train/obj_loss curve captures the loss related to objectness prediction. In YOLOv5, each grid cell predicts whether an object is present within it or not. The objectness loss calculates the difference between the predicted objectness scores and the ground truth labels. This loss helps the model to discriminate between objects and background regions. As the training progresses, the model learns to assign higher objectness scores to grid cells containing license plates and lower scores to empty cells, resulting in a decrease in the objectness loss. A decreasing obj_loss curve indicates the model's improved capability to identify relevant regions for license plate recognition.

Fig. 2. Training loss curves.

The train/cls_loss curve represents the loss associated with the classification of license plate characters. In license plate recognition, the model needs to classify the alphanumeric characters present on the license plates. The cls_loss measures the dissimilarity between the predicted class probabilities and the ground truth character labels. The model aims to accurately recognize and classify the characters. As the training advances, the cls_loss decreases, indicating that the model becomes more proficient at correctly identifying the characters on the license plates. A diminishing cls_loss curve reflects the model's improved ability to perform accurate character classification.

By monitoring the train/box_loss, train/obj_loss, and train/cls_loss curves during training, one can gain insights into the model's learning progress and performance in different aspects of license plate recognition. The decreasing trends in these loss curves indicate the model's improvement in bounding box prediction, objectness estimation, and character classification, respectively.

2) Validation: In the validation module, the YOLOv5 model is evaluated on the validation set to assess its performance and fine-tune its hyperparameters. Based on the validation results, adjustments can be made to the model architecture, training parameters, or data augmentation techniques to improve its performance. During the validation process of a YOLOv5 model for license plate recognition, the loss curves serve as valuable metrics to assess the model's performance. Fig. 3 shows validation loss curves.

As shown in Fig. 3, the val/obj_loss curve corresponds to the loss related to objectness prediction during validation. It evaluates the difference between the predicted objectness scores and the ground truth labels for license plates in the validation images. As the model learns to distinguish between objects and background regions, the val/obj_loss decreases. A decreasing obj_loss curve indicates that the model is becoming more proficient at identifying relevant regions for license plate recognition during validation. This demonstrates the model's improved capability to assign higher objectness scores to grid cells containing license plates and lower scores to empty cells, enhancing its ability to discriminate between objects and background regions.

Fig. 3. Validation loss curves.

The val/cls_loss curve represents the loss associated with the classification of license plate characters during validation. The model aims to accurately recognize and classify the alphanumeric characters. As the model improves its ability to correctly identify characters on license plates during validation, the val/cls_loss, decreases. A diminishing cls_loss curve indicates that the model becomes more proficient at accurately classifying characters on license plates. This signifies the model's enhanced ability to perform accurate character recognition and classification during validation.

By monitoring the val/box_loss, val/obj_loss, and val/cls_loss curves during the validation process, one can evaluate the model's performance in bounding box prediction, objectness estimation, and character classification for license plate recognition tasks. Decreasing trends in these loss curves indicate the model's improvement in accurately localizing license plates, discriminating between objects and background regions, and correctly identifying characters.

IV. Experimental Results

This section presents experimental results and performance evaluation. For these results, the trained YOLOv5 model is applied to the unseen license plate images in the testing set. The model's predictions are compared against the ground truth annotations to evaluate its performance on new and unseen data. These metrics are precision, recall and F1-score are computed to assess the model's effectiveness in license plate recognition. We use the testing sets for this evaluation. The testing module provides insights into the model's generalization ability and its performance in real-world scenarios. By following this three-module approach, the YOLOv5 model is trained on the labeled license plate images from the training set, validated on the validation set to fine-tune its performance, and then tested on the unseen license plate images from the testing set to evaluate its effectiveness in license plate recognition tasks. This process ensures the development of a robust and accurate YOLOv5 model for license plate recognition on the custom dataset.

A. Confusion Matrix

Confusion matrix as popular tool for evaluating the performance of deep learning-based models is utilized in this experiment. It serves as a visual representation and quantitative assessment of the license plate recognition YOLOv5 model's predictive performance, allowing to evaluate, analyze, and improve the model's accuracy and effectiveness. Additionally, the confusion matrix enables the calculation of class-specific performance metrics, providing a more detailed understanding of the model's strengths and weaknesses across different license plate character classes.

As shown in Fig. 4, the confusion matrix is organized as a grid, with the Y-axis representing the predicted classifications and the X-axis representing the true classifications. The main diagonal of the confusion matrix corresponds to the correctly classified instances, where the predicted class matches the true class. Off-diagonal cells represent the misclassified instances, where the predicted class differs from the true class.

B. Performance Measurement

By analyzing the confusion matrix, performance measurements can be performed. Popular metrics such as precision, recall, F1-score and mAP are calculated from the values in the confusion matrix.

1) Precision-confidence curve: The precision-confidence curve is a visual representation of the relationship between precision and confidence threshold in a license plate recognition YOLOv5 model. The precision-confidence curve plots the precision value at different confidence thresholds. Starting from a high confidence threshold, the precision will be relatively high because only confident predictions are considered valid. Fig. 5 demonstrates the precision-confidence curve.

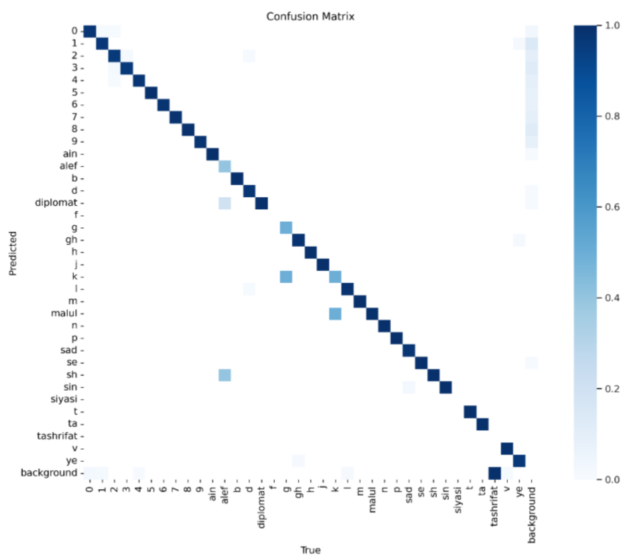

Fig. 4. Results of confusion matrix.

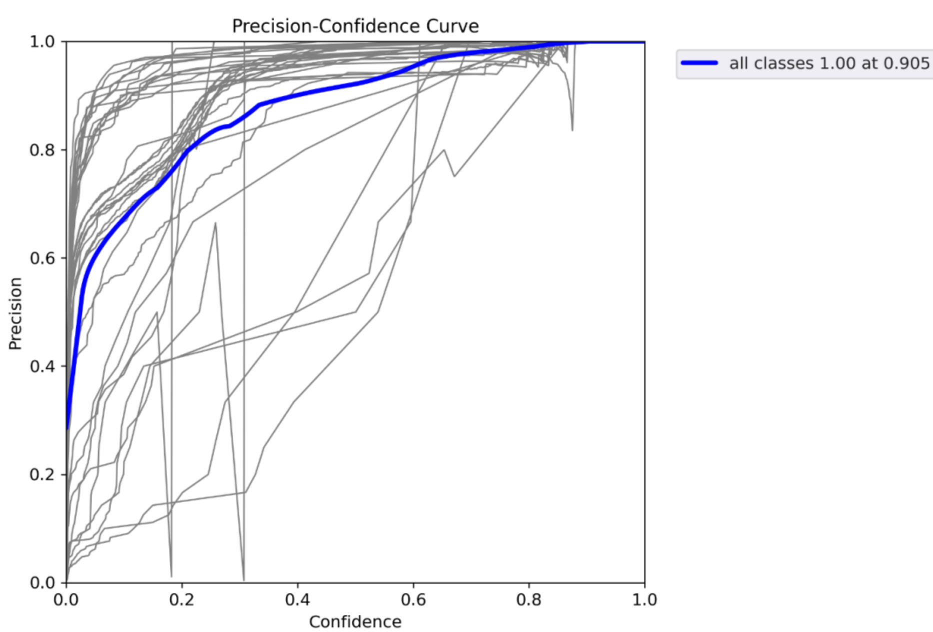

Fig. 5. The precision-confidence curve.

As shown in Fig. 5, The X-axis represents the confidence threshold, while the Y-axis represents the precision. The curve demonstrates how precision changes as the confidence threshold is adjusted. Initially, with a high confidence threshold, the precision is high because only confident predictions are considered valid. As the confidence threshold decreases, more predictions are included, potentially leading to false positives and a decrease in precision. Higher precision implies a lower false positive rate, indicating fewer incorrect predictions. However, setting a high confidence threshold may result in missing true positive instances, reducing efficiency. By analyzing the curve, it is possible to identify the optimal confidence threshold that strikes a balance between precision and efficiency. This information enables to assess the model's performance and adjust the confidence threshold to achieve the desired accuracy and recognition efficiency in the license plate recognition model.

2) Recall-confidence curve: The recall-confidence curve illustrates how the recall metric changes as the confidence threshold varies. As the confidence threshold decreases, more predictions are considered valid, including both true positives and potential false positives. Fig. 6 shows the recall-confidence curve.

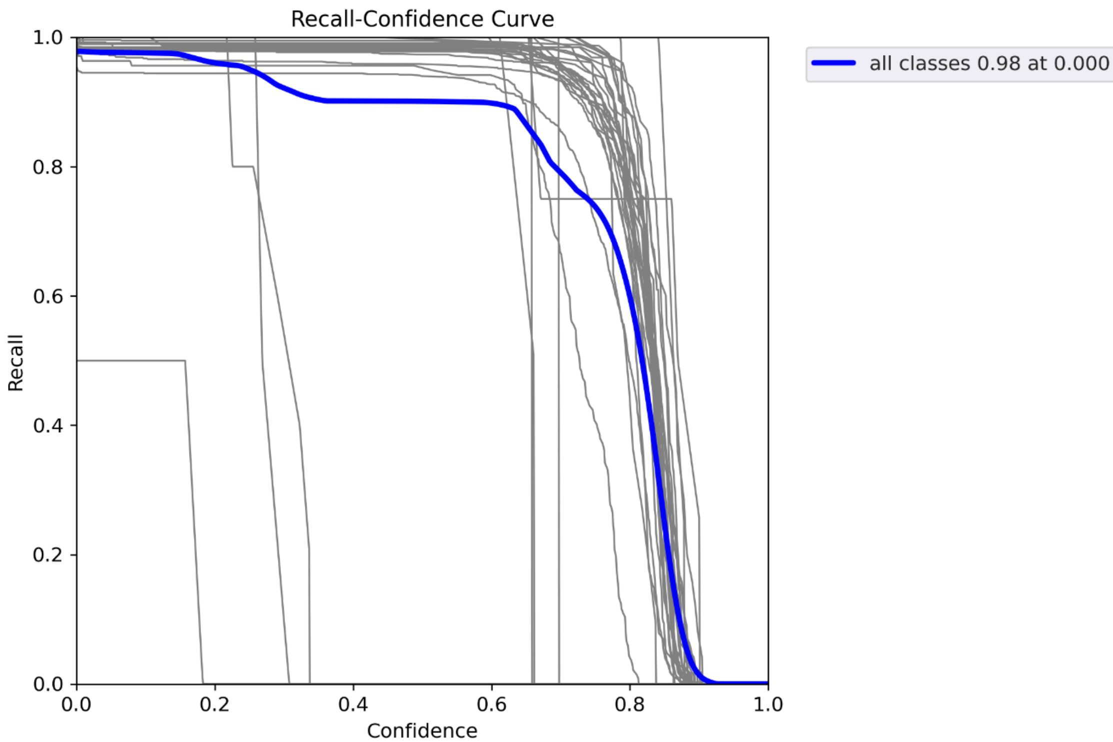

Fig. 6. The recall-confidence curve.

A shown in Fig. 6, the X-axis of the curve represents the confidence threshold, which determines the minimum confidence score required for a prediction to be considered valid. The Y-axis represents the recall, also known as sensitivity or true positive rate. Recall measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives plus false negatives). Higher recall indicates that the model is better at capturing and recognizing actual positive instances, minimizing false negatives. However, a lower confidence threshold that maximizes recall may also introduce more false positives, reducing the model's efficiency. It becomes a trade-off between correctly identifying more positive instances and the potential inclusion of false positives.

3) Precision-Recall curve: The precision-recall curve is another visual representation that provides insights into the performance and efficiency of a license plate recognition YOLOv5 model. The curve showcases the relationship between precision and recall, two important metrics used to evaluate the model's predictive accuracy. Fig. 7 shows the precision-recall curve.

The X-axis of the precision-recall curve represents the recall, also known as sensitivity or true positive rate. The Y-axis represents precision, which indicates the proportion of correctly predicted positive instances out of all instances predicted as positive (true positives plus false positives). The precision-recall curve illustrates how the precision metric change as the recall varies. A higher recall indicates that the model is capturing a larger number of positive instances, minimizing false negatives. As the recall increases, the model is becoming more sensitive and successfully identifying more actual positive instances. However, as the recall increases, it becomes more challenging to maintain a high precision. The inclusion of more positive instances may introduce false positives, impacting the precision of the model.

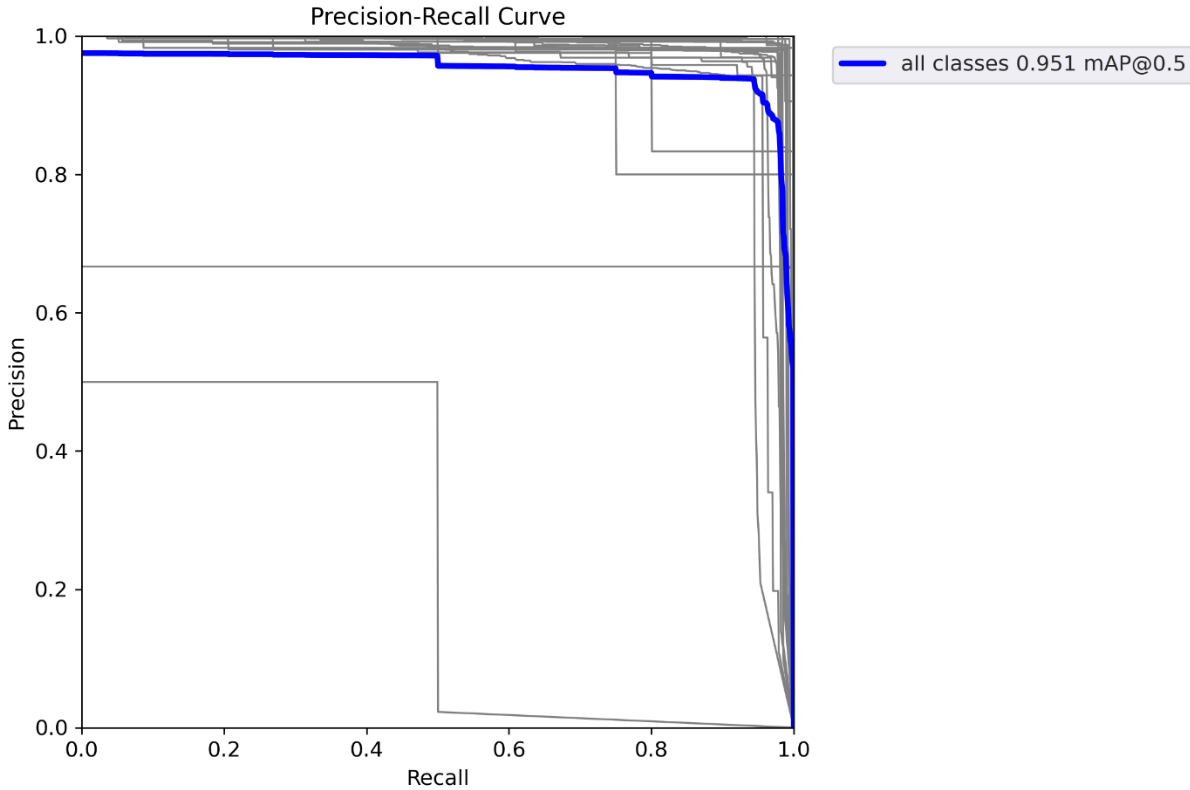

Fig. 7. The precision-recall curve.

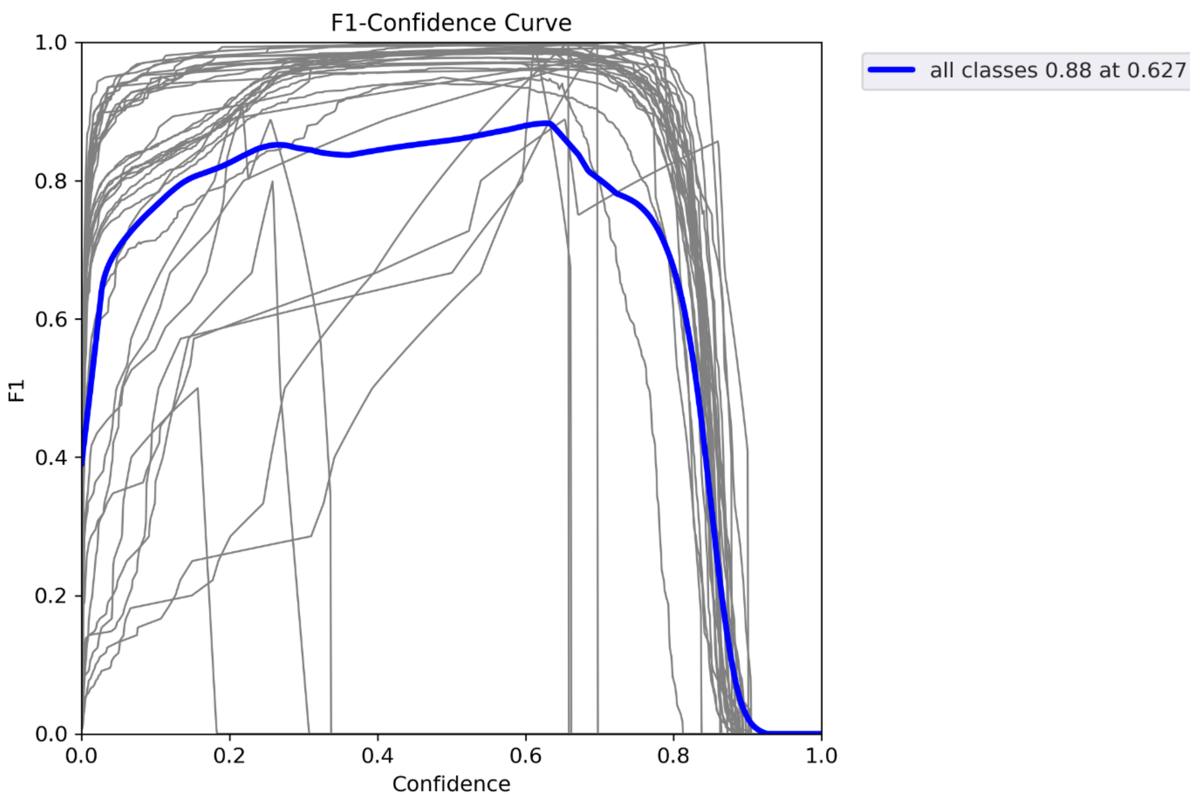

Fig. 8. F1-score curve.

4) F1-score: The F1-score curve demonstrates how the F1-score change as the threshold varies. The F1-score is important for measuring the efficiency of a license plate recognition YOLOv5 model because it combines precision and recall into a single value. It provides a comprehensive assessment of the model's ability to accurately identify positive instances while avoiding false positives.

As shown in Fig. 8, the curve illustrates the relationship between the F1-score and a varying threshold used for predictions. The X-axis of the F1-score curve represents the threshold, which is the minimum confidence score required for a prediction to be considered valid. The Y-axis represents the F1-score, which is a metric that combines both precision and recall into a single value. The F1-score provides a balanced evaluation of the model's performance, taking into account both the ability to correctly identify positive instances (precision) and the ability to capture all actual positive instances (recall). As the threshold decreases, more predictions are considered valid, potentially increasing the recall of the model.

V. Conclusion

In this study, we introduce a YOLO-based method complemented by a custom dataset tailored for license plate recognition within smart parking systems. Our research effectively addresses the primary research gap, delivering real-time processing capabilities and enhanced accuracy. The extensive custom dataset empowers the model to handle even the most challenging scenarios frequently encountered in smart parking environments. Our comprehensive experimental evaluations underscore the method's superior performance across accuracy, processing speed, and robustness. The significant contributions of this research encompass the identification of limitations, the creation of a specialized dataset, the proposal of the YOLO-based approach, and the thoroughness of our evaluations. These findings collectively advance the development of precise and efficient license plate recognition systems, thereby augmenting the management of smart parking in urban areas. It's worth noting that our custom dataset specifically targets the license plate types prevalent in the studied region or environment. Nevertheless, variations in license plate designs, formats, fonts, sizes, colors, and layouts exist across different countries, regions, or vehicle types. Future endeavors can prioritize the development of methods capable of generalizing and adapting to this diversity, potentially incorporating transfer learning, data augmentation techniques, or domain adaptation strategies to enhance the model's proficiency in accurately recognizing and interpreting various license plate variations.

References

[1] T. Perković, P. Šolić, H. Zargariasl, D. Čoko, and J. J. Rodrigues, "Smart parking sensors: State of the art and performance evaluation," Journal of Cleaner Production, vol. 262, p. 121181, 2020.

[2] C. Biyik et al., "Smart parking systems: Reviewing the literature, architecture and ways forward," Smart Cities, vol. 4, no. 2, pp. 623-642, 2021.

[3] R. Nawaratne, S. Kahawala, S. Nguyen, and D. De Silva, "A generative latent space approach for real-time road surveillance in smart cities," IEEE Transactions on Industrial Informatics, vol. 17, no. 7, pp. 4872-4881, 2020.

[4] M. S. Farag, M. M. El Din, and H. El Shenbary, "Parking entrance control using license plate detection and recognition," Indonesian Journal of Electrical Engineering and Computer Science, vol. 15, no. 1, pp. 476-483, 2019.

[5] S. M. Mutua, "An automatic number plate recognition system for car park management," Strathmore University, 2016.

[6] A. Fahim, M. Hasan, and M. A. Chowdhury, "Smart parking systems: comprehensive review based on various aspects," Heliyon, vol. 7, no. 5, p. e07050, 2021.

[7] G. T. Sutar, A. M. Lohar, and P. M. Jadhav, Number plate recognition using an improved segmentation. Citeseer, 2019.

[8] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," Journal of Visual Communication and Image Representation, vol. 71, p. 102773, 2020.

[9] A. Farley and H. Ham, "Real time IP camera parking occupancy detection using deep learning," Procedia Computer Science, vol. 179, pp. 606-614, 2021.

[10] R. N. Babu, V. Sowmya, and K. Soman, "Indian car number plate recognition using deep learning," in 2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICICT), 2019, vol. 1: IEEE, pp. 1269-1272.

[11] J. Du, "Understanding of object detection based on CNN family and YOLO," in Journal of Physics: Conference Series, 2018, vol. 1004: IOP Publishing, p. 012029.

[12] S. Wu, X. Li, and X. Wang, "IoU-aware single-stage object detector for accurate localization," Image and Vision Computing, vol. 97, p. 103911, 2020.

[13] W. Weihong and T. Jiaoyang, "Research on license plate recognition algorithms based on deep learning in complex environment," IEEE Access, vol. 8, pp. 91661-91675, 2020.

[14] S. Raj, Y. Gupta, and R. Malhotra, "License plate recognition system using yolov5 and cnn," in 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, vol. 1: IEEE, pp. 372-377.

[15] M. M. Abdellatif, N. H. Elshabasy, A. E. Elashmawy, and M. AbdelRaheem, "A low cost IoT-based Arabic license plate recognition model for smart parking systems," Ain Shams Engineering Journal, vol. 14, no. 6, p. 102178, 2023.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[17] Al-Batat, Reda, et al. "An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification." Sensors 22.23 (2022): 9477.

[18] Cao, Yong. "Investigation of a convolutional neural network-based approach for license plate detection." Journal of Optics (2023): 1-7.

[19] Shi, Hengliang, and Dongnan Zhao. "License Plate Recognition System Based on Improved YOLOv5 and GRU." IEEE Access 11 (2023): 10429-10439.