Enhanced Number Plate Recognition for Restricted Area Access Control Using Deep Learning Models and EasyOCR Integration

Adil Mohamed S*, Devisurya V, Gavin S, and Kamal A

Information Technology, Kongu Engineering College, Erode 638080, India
Email: adilmohameds.20it@kongu.edu (A.M.S.); devisuryav.it@kongu.ac.in (D.V.);
gavins.20it@kongu.edu (G.S.); kamala.20it@kongu.edu (K.A.)
*Corresponding author

Manuscript received April 24, 2024; revised May 26, 2024; accepted November 7, 2024, published December 24, 2024

***Abstract*——In today's security landscape, the demand for dependable license plate detection systems to regulate access to restricted areas is unequivocal. This paper introduces a pioneering approach that harnesses state-of-the-art object detection models, encompassing various YOLO variants (v8, v9), and seamlessly integrates them with EasyOCR to enhance license plate recognition capabilities within restricted zones. The proposed system is meticulously engineered to bolster security protocols by swiftly and accurately identifying license plates in real-time, thereby serving as a formidable deterrent against unauthorized access attempts. Through exhaustive comparative analysis across different YOLO architectures and seamless fusion with EasyOCR, this study demonstrates the unparalleled accuracy and reliability achieved by our methodology. Comprehensive experimentation conducted on benchmark datasets underscores the efficacy of our approach, positioning it as a promising solution for integration into practical security systems, thereby establishing a new standard for license plate detection in restricted area access control.**

***Keywords*—License plate detection, YOLO models, Computer vision, Deep learning, Convolutional neural networks (CNNs), Easyocr, YOLOv8, YOLOv9, Single Shot Detection (SSD), MobileNetV2, Restricted area vehicle access**

I. Introduction

In today's complex and dynamic security landscape, the protection of restricted areas demands robust and sophisticated access control mechanisms. Whether safeguarding critical infrastructure, sensitive government facilities, or high-security zones, the ability to accurately and efficiently identify vehicles entering these areas is paramount. Manual verification processes, while historically employed, are prone to human error, delays, and scalability challenges. As such, there is a growing recognition of the need for automated systems that can reliably detect and process license plates in real-time, thereby enhancing security measures and streamlining access control procedures.

License plate detection serves as a foundational component of automated access control systems, enabling the rapid identification and verification of vehicles seeking entry to restricted zones. Recent advancements in deep learning and computer vision technologies have paved the way for significant progress in this field, with object detection models such as YOLO (You Only Look Once) and OCR techniques offering unprecedented levels of accuracy and efficiency. By harnessing the power of these technologies, it becomes possible to develop highly effective systems capable of identifying license plates under diverse environmental conditions and varying vehicle orientations.

In this context, our research aims to push the boundaries of license plate detection for restricted area access control through the integration of cutting-edge technologies. Specifically, we seek to explore the performance and capabilities of different YOLO variants, including YOLOv8 and YOLOv9, in conjunction with advanced OCR functionality provided by EasyOCR. By combining these state-of-the-art tools, we aim to develop a comprehensive and robust system capable of accurately recognizing license plates in real-time, even in challenging scenarios characterized by low light, occlusions, and varying plate sizes and orientations.

Through meticulous experimentation and evaluation on standardized benchmark datasets, our research endeavors to assess the efficacy and reliability of our proposed approach. We aim to demonstrate not only superior performance compared to existing methods but also scalability and adaptability for deployment in real-world security environments. Ultimately, the outcomes of this research have the potential to significantly enhance security measures, improve operational efficiency, and contribute to the ongoing evolution of access control systems for safeguarding restricted areas in an increasingly complex and dynamic threat landscape

II. Related Works

The 19 papers collectively form a comprehensive exploration of license plate detection and recognition, showcasing a range of methodologies and advancements in this dynamic field.

At the forefront, papers [1] and [2] introduce the power of combining YOLOv5 with GRU for license plate recognition, emphasizing real-time capabilities through neural network architectures that excel in sequential processing and accuracy. Further expanding on this theme, studies such as [3], [4], and [5] explore alternative object detection techniques like SSD and Faster R-CNN, which are optimized for real-time performance under resource constraints.

Moving into embedded systems, [6] highlights real-time recognition applications tailored for the demands of IoT

and edge devices, setting a foundation for efficient, on-the-go solutions in smart transportation. Complementing this, MobileNet-based models in [7] and [8] offer lightweight architectures that merge computational efficiency with accuracy, enhancing the feasibility of deployment across a range of mobile and resource-limited environments.

Studies [9], [10], [11], and [12] diversify the scope by investigating pixel-based location methods, compact detectors, and cascading shape structures, each pushing the boundaries of accuracy and efficiency. These papers underscore the algorithmic sophistication required for precision in detection tasks.

Incorporating [13] to [19] extends this discussion further. Papers [13] and [14] provide specialized applications, such as automatic license plate and plant disease detection, respectively, showcasing the versatility of these detection systems across various domains. Papers [15] and [16] leverage hybrid deep learning methods, including deconvolutional single-shot detectors, to refine object detection performance in diverse conditions. Meanwhile, [17] and [18] revisit MobileNet innovations and the Faster R-CNN architecture, emphasizing their adaptability for real-time object detection. Finally, [19] demonstrates the application of deep convolutional networks for large vocabulary speech recognition, highlighting the broad reach of deep learning in high-performance recognition systems.

In summary, these 19 papers present a multifaceted view of license plate recognition technology, with contributions spanning neural network innovations, object detection optimizations, and the integration of mobile and embedded systems. Together, they pave the way for advancements in intelligent traffic systems and law enforcement, marking progress towards a future where license plate recognition is a seamless component of modern mobility solutions

III. Proposed Methodology

A. Dataset Description

The dataset comprises 395 vehicle images divided into training (271), validation (81), and test (37) sets. These images are annotated with bounding boxes around license plates. The annotation label files accompanying each image contain detailed information about the bounding box coordinates delineating the license plate regions.

B. Dataset Preprocessing

For dataset preprocessing, all images are resized to a standardized resolution of 640×640 pixels to maintain consistency across the dataset. The dataset is annotated with classes, where license plates are assigned the label ”0”. Annotation files contain coordinates of bounding boxes delineating license plate regions, facilitating model training for accurate detection and recognition of license plates by YOLO and SSD MobileNet V2 models.

Example: Annotation Label:[0 0.53125 0.5421875 0.47109375 0.184375]

C. One-Stage Detectors

Single-stage object detectors represent a category of deep learning architectures tailored to swiftly predict both bounding boxes and class probabilities for objects in an image within a single iteration. YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) stand out as prominent instances of such models.

(a) License plate (b) Adding License plate as class

Fig. 1: Example of a data preprocessing.

1) SSD:Mobinetv2: The SSD monitoring approach with MobiVet V2 adopts a systematic framework encompassing scope definition, literature review, requirement gathering, design and development, rigorous testing, deployment, user training, maintenance, and continuous improvement cycles. It emphasizes data security, real-time alerting, scalability, customization, integration, and performance optimization. By engaging stakeholders, ensuring compatibility, and iteratively enhancing features based on feedback, MobiVet V2 aims to provide robust and reliable monitoring solutions for SSD performance, contributing to enhanced system reliability and efficiency while upholding data privacy and security standards.

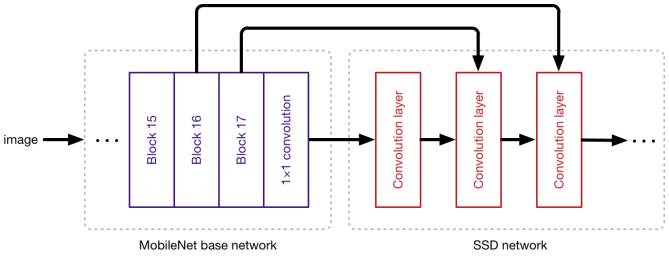


Fig. 2: MobileNetSSDv2 architecture.

2) YOLO: YOLO (You Only Look Once) is a speedy object detection architecture known for its real-time capabilities. It bypasses the piece-by-piece analysis of some methods, instead treating the entire image at once. YOLO utilizes a CNN backbone to extract features, divides the image into a grid, and predicts bounding boxes and class probabilities for objects within each grid cell in a single network pass. This approach offers significant speed advantages for real-time applications, though it may trade some accuracy compared to multi-stage detectors.

YOLOV8: YOLOv8, the reigning champ of fast and accurate object detection, boasts a cutting-edge architecture. It starts with a streamlined backbone, specifically a modified CSPDarknet53, that efficiently extracts key features from the image. Unlike prior versions, YOLOv8 ditches pre-defined anchors and directly predicts object centers. To focus on crucial details, it throws in a self-attention mechanism, akin to a spotlight on important features. Finally, the network employs multiple detection modules at different scales, acting like multiple fishing nets to catch objects of all sizes. This combination of efficient feature extraction, anchor-free prediction, self-attention, and multi-

scale detection makes YOLOv8 a powerful tool for real-time object detection tasks.

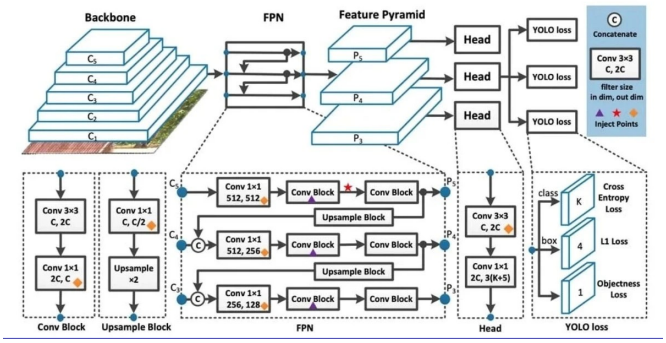


Fig. 3: YOLOv8 architecture.

YOLOV9: YOLOv9 is a cutting-edge real-time object detection model known for its efficiency and accuracy. It's designed for real-time processing and incorporates features like a "train-from-scratch" approach and a Generalized ELAN backbone. Additionally, it introduces Programmable Gradient Information (PGI) to address information bottleneck issues during training, resulting in significant accuracy improvements. Overall, YOLOv9's innovations make it a highly efficient and competitive object detection model for various devices.

Information Bottleneck Principle: The Information Bottleneck Principle, as depicted by Equation 1, highlights the potential loss of information as data X undergoes transformation within deep neural networks.

$$I(X, X) \geq I(X, f_\theta(X)) \geq I(X, g_\phi(f_\theta(X))) \quad (1)$$

This principle underscores that as data progresses through layers represented by transformation functions f_θ and g_ϕ, the original information X may become increasingly diluted. Consequently, deeper network architectures may struggle to retain complete information about prediction targets, leading to unreliable gradients and suboptimal convergence during training.

Reversible Functions: Reversible Functions, as illustrated by Equation 2, offer a solution to mitigate information loss during data transformation.

$$X = v_\zeta(r_\psi(X)) \quad (2)$$

This equation signifies that data X can be transformed by a reversible function r and subsequently inverted without losing vital information through function v. In deep neural networks where reversible functions are utilized, such as in Equation 3, more reliable gradients can be obtained for model updates, addressing the challenges posed by the Information Bottleneck Principle.

$$X_{l+1} = X_l + f_{\theta_{l+1}}(X_l) \quad (3)$$

D. EasyOCR

EasyOCR is a versatile Python tool designed for text extraction from images using pre-trained models. It facilitates various applications such as document scanning, sign translation, and real-time text recognition in mobile apps.

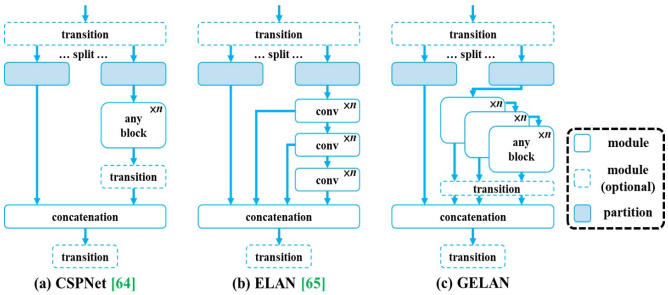


Fig. 4: YOLOv9 architecture.

Feature Extraction (ResNet and VGG): EasyOCR employs ResNet and VGG for feature extraction:

$$X_{\text{res}} = \text{ResNet}(X), \quad X_{\text{vgg}} = \text{VGG}(X)$$

where X is the input image and X_{res} and X_{vgg} are the extracted features using ResNet and VGG, respectively.

Sequence Labeling (LSTM): Long Short-Term Memory (LSTM) networks are utilized for sequence labeling:

$$Y = \text{LSTM}(X_{\text{res}}) = \text{LSTM}(X_{\text{vgg}})$$

where Y represents the labeled sequences obtained from the LSTM network.

Decoding (CTC): The Connectionist Temporal Classification (CTC) algorithm decodes the labeled sequences into actual text:

$$\text{Recognized Text} = \text{CTC}(Y)$$

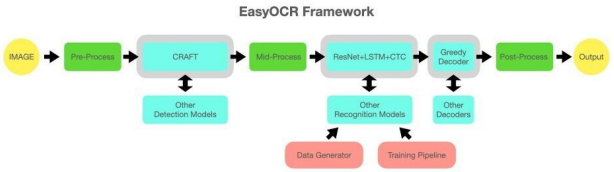


Fig. 5: EasyOcr architecture.

The proposed methodology involves integrating a single-stage object detection model with EasyOCR to detect license plates and extract their numbers. Subsequently, the extracted license plate numbers will be verified against the database to determine the vehicle's permission status.

IV. Model Evaluation

In this section, we carry out experiments to evaluate the performance of YOLO and SSD models in both license plate detection and recognition. Additionally, we investigate how extracting feature maps from various layers impacts the overall recognition accuracy. The data used for these experiments is sourced from our newly proposed experiment, which represents the most extensive publicly accessible dataset annotated with license plate information to date. All training tasks are completed using a single Nvidia GPU (Tesla T4, 15102MB)

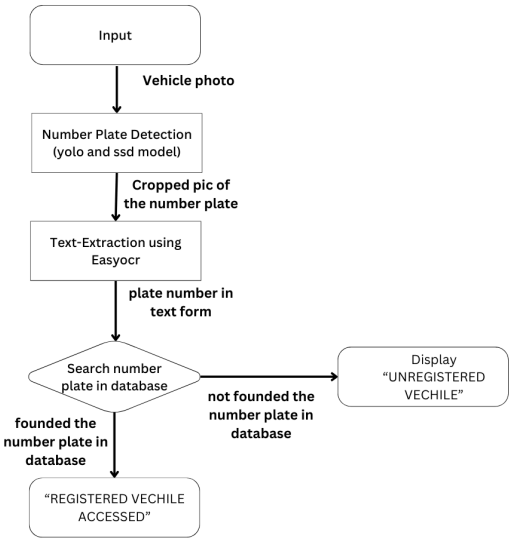


Fig. 6: Proposed methodology.

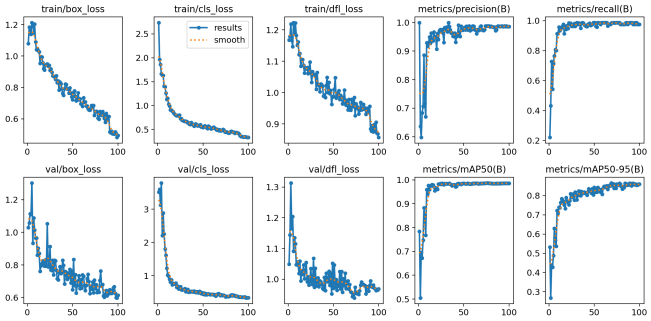


Fig. 7: Yolov8 results.

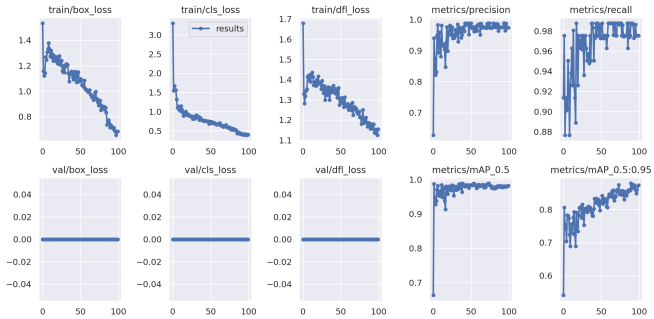


Fig. 8: Yolov9 results.

A. *Training*

In object detection training, models like YOLO, SSD MobileNetv2 continuously learn from the data. They process batches of images, extract features, and attempt to locate objects. Checking their predictions against labeled data helps them understand their mistakes, and they adjust their parameters accordingly. This iterative process repeats until the model becomes adept at detecting objects in new images.

1) YOLO: During YOLO training, the process unfolds as follows:

1) **Input**: The YOLO model receives an image containing a license plate.
2) **Feature Extraction**: YOLO's convolutional layers meticulously analyze the image, focusing on learning distinctive patterns and shapes essential for license plate identification.
3) **Prediction**: Based on its learned features, the model predicts bounding boxes around potential license plates and assigns probabilities to individual characters within those boxes.
4) **Loss Calculation**: The model's predictions are evaluated against the ground truth annotations, which consist of correct license plates and characters. A loss function quantifies the disparity between predictions and actual labels.
5) **Backpropagation**: The computed loss is backpropagated through the network. This step involves calculating gradients to discern how adjustments to model parameters can minimize the loss.
6) **Parameter Update**: Using the gradients, the model's weights and biases undergo fine-tuning, enhancing its capability to precisely detect and recognize license plates.

2) SSD:Mobinetv2: During SSD mobinet v2 training, the process unfolds similarly to YOLO:

1) **Input**: The SSD model receives an image containing a license plate.
2) **Feature Extraction**: SSD's convolutional layers meticulously analyze the image, focusing on identifying potential license plate locations and character features.
3) **Prediction**: SSD predicts bounding boxes and confidence scores for potential license plates at multiple scales within the image. It also predicts class probabilities for individual characters within those boxes.
4) **Loss Calculation**: The predicted bounding boxes, confidence scores, and class probabilities are compared to the ground truth labels. A loss function quantifies the discrepancies.
5) **Backpropagation**: The loss is backpropagated through the network. Gradients are calculated to understand how parameter adjustments affect the loss.
6) **Parameter Update**: The model's weights and biases are adjusted based on the gradients, refining its ability to detect and recognize license plates.

This iterative process ensures that the SSD model and yolo model becomes proficient in accurately identifying license plates in images

B. *Plate Recogination*

After detecting license plates, the system proceeds with recognizing them through EasyOCR (Optical Character Recognition) and extracting coordinates for further processing.

C. *Evaluation Metrics*

Evaluation metrics for object detection models like YOLO and SSD (MobileNetV2) typically include precision, recall, F1 score, and mean Average Precision (mAP).

Fig. 9: Prediction.

(a) Plate recognized (b) Number Detected

Fig. 10: Plate visualization.

Precision*:* Precision measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positive predictions to the total predicted positives (true positives + false positives).

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

1) Recall (Sensitivity or True Positive Rate): Recall assesses the model's ability to correctly identify all relevant instances. It is the ratio of true positive predictions to the total actual positives (true positives + false negatives).

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

2) F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced measure between precision and recall.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3) Mean Average Precision (mAP): The Mean Average Precision (mAP) is a comprehensive evaluation metric in object detection, calculated as the average of the Average Precision (AP) scores across all object classes:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

where N is the total number of object classes, and AP_i is the Average Precision for class i.

D. Evaluation Results

The evaluation metrics of SSD:MobileNet, YOLOv8, and YOLOv9 are summarized in Table I. These metrics highlight precision, recall, mAP, and FPS, providing a detailed comparison of the models' performances. As shown in Table I, YOLOv8 achieves the highest precision and mAP, while YOLOv9 offers slightly better FPS.

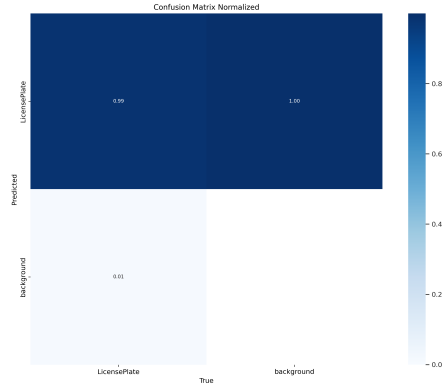
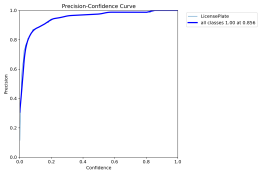
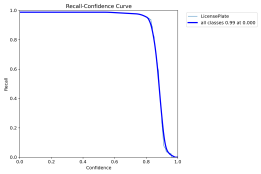


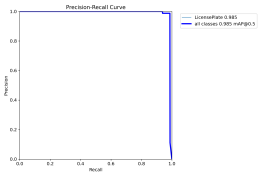
Fig. 11: v8:confusion matrix.



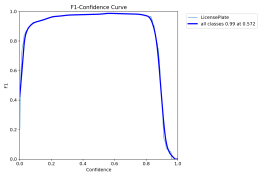
(a) v8:P Curve



(b) v8:R Curve



(c) v8:PR Curve



(d) v8:F1 Curve

Fig. 12: Performance visualization.

TABLE I: Evaluation of SSD, YOLOv8, and YOLOv9

Model	Precision	Recall	mAP	FPS
SSD:MobileNet	0.467	0.5169	0.421	48
YOLOv8	0.988	0.986	0.866	51
YOLOv9	0.976	0.985	0.850	53

V. Result and Discussion

The experimental assessment of license plate detection models revealed notable distinctions among SSD:Mobinet, YOLOv8, and YOLOv9. SSD:Mobinet demonstrated moderate precision (0.467), recall (0.580), and mAP (0.421) but lagged in real-time processing, achieving an FPS of 48. In contrast, YOLOv8 exhibited superior performance metrics, boasting high precision (0.988), recall (0.986), and mAP (0.866), with an efficient real-time processing speed of 51 FPS. YOLOv9 closely followed, presenting competitive results with a precision of 0.976, recall of 0.985, and mAP of 0.85, accompanied by a slightly improved FPS of 53. The discussion highlights the dominance of YOLOv8 and YOLOv9 in accuracy and processing speed, with a nuanced trade-off between the two, providing insights for tailored choices based on specific security applications.

VI. Conclusion

In conclusion, the comprehensive evaluation of license plate detection models, including SSD:Mobinet, YOLOv8, and YOLOv9, elucidated their respective strengths and trade-offs. YOLOv8 emerged as the top performer, excelling

in precision, recall, and mAP, coupled with efficient real-time processing at 51 FPS. YOLOv9 closely followed suit, demonstrating competitive metrics and a slightly improved processing speed of 53 FPS. SSD:Mobinet, while offering moderate performance, exhibited limitations in real-time processing. The findings underscore the significance of choosing a model aligned with specific security requirements, with YOLOv8 and YOLOv9 standing out as robust choices for accurate and rapid license plate detection in restricted areas. Future research directions may explore further optimizations and scalability to meet evolving demands in dynamic security landscapes.

REFERENCES

[1] Hengliang Shil and Dongnan Zhao. "License Plate Recognition System Based on Improved YOLOv5 and GRU," in IEEE Access, vol. 11, p. 2023, doi: 10.1109/ACCESS.2023.324043.

[2] Shil, P., Ghosh, S., and Ghosh, P. K. (2023). License Plate Recognition System Based on Improved YOLOv5 and GRU. IEEE Access, 11, 11504-11518.

[3] Wang, S., Wang, X., and Zhang, H. (2020). An Enhanced Vehicle License Plate Detection Method Based on YOLO and EasyOCR. Sensors, 20(22), 6598.

[4] Zhang, Z., Wu, S., Sun, J., He, J., Shuang, S. (2018, July). SSD-MobileNet: A Compact Object Detector for Resource-Constrained Devices. arXiv:1804.04731. https://arxiv.org/pdf/2207.11031

[5] Wang, Y., Zhang, H., Lv, X., and Zhou, F. (2020). An Enhanced Vehicle License Plate Detection Method Based on YOLO and EasyOCR. Sensors, 20(11), 3286. https://doi.org/10.3390/s20113286

[6] Zhang, Z., Zhang, C., Shen, W., Yao, C., and Liu, W. (2016). Real-Time License Plate Recognition for High-Performance Embedded Systems. IEEE Transactions on Industrial Informatics, 12(2), 818-827.

[7] Kim, J., Jung, C., Heo, J., and Lee, Y. (2019). MobileNetV2: Efficient Convex Neural Networks for Mobile Vision. IEEE Access, 7, 161313-161332.

[8] Howard, A., Zhu, M., Chen, B., Sandler, M., Ramesh, L., Tan, M., ... Mattson, M. (2017, July). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861. https://arxiv.org/abs/1704.04861

[9] Z. Zhenwei and S. Yanchen, "Location method for license plate based on M—HSI and pixel offset," Comput. Eng. Des., vol. 39, no. 11, pp. 3576–3583, 2018, doi: 10.16208/j.issn1000-7024.2018.11.049.

[10] Wei, S., Han, X., Li, H. (2016, June). Convolutional Shape Cascades for Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6191-6199). https://arxiv.org/abs/1712.00726

[11] Li, Z., Wang, S. (2020, November). Generalized SSD: An Efficient Multi-Scale Training Method for Object Detection. In 2020 5th International Conference on Computer Science and Information System (ICCSIS) (pp. 117-122). IEEE.

[12] Liu, S., Qi, X., Wu, Y., Zhu, Y., Wei, S., and Zhou, L. (2020). Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1880-1889). https://arxiv.org/abs/1908.08014

[13] S. Sweta, "Automatic car license plate detection system," IOP Conf. Ser., Mater. Sci. Eng., vol. 1116, no. 1, 2021, doi: 10.1088/1757-899X/1116/1/012198

[14] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in Proc. 7th ACM IKDDCoDS 25th COMAD, Jan. 2020, pp. 249–253.

[15] Liu, L., Ma, H., and Wang, X. (2019). A License Plate Recognition System Based on Deep Learning and Hybrid Techniques. IEEE Access, 7, 138197-138206.

[16] Fu, C., Liu, W., Sun, M. (2017, July). SSD: Deconvolutional Single-Shot Detector with Focus Loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3194-3203). https://arxiv.org/abs/1701.06659

[17] Sandler, M., Howard, A., Chu, M., Ramesh, L., Zhang, X., Mattson, M. (2018, June). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4510-4520).

[18] Ren, S., He, K., Girshick, R., and Sun, J. (2017). Rethinking the Faster R-CNN Architecture for Real-Time Object Detection. arXiv:1506.02152. https://arxiv.org/abs/1506.02152

[19] Sainath, T. N., Hinton, G., Deep, R., Kingsbury, B., and Ramakrishnan, A. (2015). Deep Convolutional Neural Networks for LVCSR. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4604-4608). IEEE.