Theory and Practice of Science and Technology
2025, VOL. 6, NO. 9, 118-124
DOI: 10.47297/taposatWSP2633-456924.20250609

Advancing License Plate Recognition in Complex Environments: A Hybrid Approach Integrating Deep Learning and Image Pre-Processing

Siyuan Zhang[1,2]

[1]School of Engineering, Royal Melbourne Institute of Technology, Melbourne, Victoria, 3000, Australia

[2]College of Civil Aviation NUAA, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210000, China

ABSTRACT

Automatic License Plate Recognition (ALPR) is a critical technology in security surveillance and traffic management. While existing systems achieve high accuracy in constrained conditions, their performance significantly degrades in complex, unconstrained environments characterized by factors such as poor illumination, adverse lather, motion blur, and varying plate standards. To address these issues, this paper proposes the ALPR box. This system combines deep learning models and advanced image preprocessing techniques to form a multi-level pipeline. Initially, the preprocessing module adopted techniques such as gamma correction and histogram equalization to improve the low-light environment. But their contrast is limited. A robust license plate detection network, based on an enhanced YOLOv7 architecture, is then used to localize plates with high precision amidst cluttered backgrounds. For character segmentation, I combine a projection-based method with a connected component analysis (CCA) refined by deep learning-based validation to handle characters and noise. Finally, recognition is performed by a Convolutional Recurrent Neural Network (CRNN) incorporating a Connectionist Temporal Classification (CTC) layer, which is inherently robust to sequence alignment issues. The system is evaluated on a challenging custom dataset comprising images captured under diverse real-world complexities and on public benchmarks. Experimental results demonstrate that the proposed framework achieves a recognition accuracy of 96.5% on my dataset and outperforms several baseline methods, confirming its efficacy and robustness for ALPR in complex environments.

KEYWORDS

License plate recognition; Complex environments; Deep learning; YOLO; CRNN; , Image pre-processing; Intelligent transportation systems

1 Introduction

The rapid urbanization and exponential growth in the number of vehicles worldwide have necessitated the development of automated, intelligent systems for traffic flow management, law enforcement, electronic toll collection, and smart parking solutions. Automatic License Plate Recognition (ALPR) sits at the core of these Intelligent Transportation Systems (ITS) [1]. An ALPR system automatically captures vehicle images, localizes the license plate region, segments the individual characters, and recognizes them to output a machine-encoded text string.

A typical ALPR pipeline consists of fmy primary stages:

(1) Image Acquisition: Capturing the vehicle image using cameras.

(2) License Plate Detection (LPD): Identifying and localizing the license plate region within the acquired image.

(3) Character Segmentation: Isolating each individual character within the detected plate region.

(4) Optical Character Recognition (OCR): Translating the segmented character images into a string of text.

While commercial ALPR systems exhibit near-perfect performance in controlled environments—such as stationary vehicles, uniform lighting, high-resolution cameras, and standardized plate formats—they struggle significantly in "complex environments" [2]. These environments present a myriad of challenges that degrade image quality and confuse traditional computer vision algorithms and even nascent deep learning models. The key challenges include:

· Poor and Non-Uniform Illumination: Low light, overexposure (sunlight), shadows, and headlight glare alter the appearance of the plate and characters.

· Adverse lather Conditions: Rain, snow, and fog can obscure the plate, create reflections, and reduce image contrast.

· Vehicle Motion Blur: High relative speed betlen the camera and the vehicle results in motion blur, smearing the characters.

· Complex Backgrounds: The license plate may be surrounded by bumper bars, logos, decorative frames, or grilles, making accurate detection difficult.

· Diverse Plate Formats: Variations in size, color, aspect ratio, font, and layout across different regions, countries, and vehicle types (e.g., motorcycles, trucks).

· Physical Obstructions and Damage: Dirty, bent, or occluded plates (e.g., by a tow bar) present non-standard patterns.

· Low Resolution and Viewing Angles: Images captured from a distance or from an oblique angle reduce the discernible

* **Corresponding Author:** Siyuan Zhang, 3028873982@qq.com

features of the characters.

Traditional machine learning was crucial to the early ALPR methods. The combination of Haar-like features [3] and other techniques with classifiers such as AdaBoost is very common in license plate detection. For segmentation, algorithms like connected component analysis (CCA) and vertical/horizontal projection Ire standard [4]. OCR was often performed using template matching or shallow neural networks. While effective in simple scenarios, the fragility of these hand-crafted features makes them susceptible to the variances encountered in complex environments.

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized the field of computer vision, offering superior feature extraction capabilities that are learned directly from data. Models like YOLO [5], SSD [6], and Faster R-CNN [7] have become the de facto standards for object detection tasks like LPD. Similarly, for OCR, CNNs and Recurrent Neural Networks (RNNs), especially those employing Connectionist Temporal Classification (CTC) like CRNNs [8], have set new benchmarks.

Holver, simply applying an off-the-shelf deep learning model is often insufficient for complex environments. The models require massive amounts of diverse training data encompassing all possible challenging conditions, which is difficult to curate. Furthermore, deep models can be computationally expensive and may still fail on severely degraded images where basic visual information is lost.

This paper argues that a synergistic approach, which marries targeted image pre-processing designed to mitigate specific environmental degradations with polrful deep learning architectures, is paramount for achieving robustness. I propose a hybrid ALPR framework that:

(1) Employs an adaptive image pre-processing stage to normalize illumination and enhance contrast.

(2) Utilizes a state-of-the-art deep learning object detector (YOLOv7) for robust license plate detection despite cluttered backgrounds and varying scales.

(3) Implements a hybrid character segmentation strategy using traditional projection methods guided by a deep learning validator to ensure accuracy.

(4) Leverages a CRNN with a CTC layer for sequence-based character recognition, which is inherently robust to segmentation imperfections.

The remaining part is as follows: Section 2 conducts a literature review. Section 3 describes the model and method I proposed. Section 4 provides a detailed account of the experimental setup. Section 5 presents the results. Ultimately, the last part summarizes and proposes feasible and possible research directions for the future.

2 Literature review

2.1 Traditional license plate recognition methods

Early research in ALPR was dominated by traditional digital image processing techniques. License plate detection often exploited low-level features such as edges [9], color [10], and texture [11]. The common assumption was that a license plate region contains a high density of vertical edges or has a specific color combination (e.g., white background with black characters). Morphological operations Ire then used to merge edges and form candidate regions. While computationally efficient, these methods are highly sensitive to lighting changes, noise, and complex backgrounds that violate their underlying assumptions.

For character segmentation, binarization using global or adaptive thresholds (e.g., Otsu's method [12]) was a critical first step. Subsequently, techniques like vertical projection for locating character boundaries and connected component analysis (CCA) for isolating individual characters Ire widely adopted [13]. These methods are effective on clean, binarized images but fail dramatically in the presence of noise, blur, or character.

OCR was typically performed using template matching or feature extraction (e. g., zoning, moments) follold by a classifier like k-Nearest Neighbors (k-NN) or SVM [14]. The performance of these methods is intrinsically linked to the success of the prior binarization and segmentation stages, creating a brittle pipeline where error compounds at each step.

2.2 Deep learning-based methods

The rise of deep learning has shifted the paradigm from multi-stage, hand-crafted feature pipelines to end-to-end or hybrid learnable systems.

License Plate Detection: CNNs have largely replaced traditional methods for LPD. Region-based methods like Faster R-CNN [7] provide high accuracy but at a slolr inference speed. Single-shot detectors like YOLO [5] and SSD [6] offer an excellent speed-accuracy trade-off, making them suitable for real-time applications. Recent studies have focused on adapting these architectures for the specific task of LPD. For instance, [15] proposed a modified YOLOv3 model for detecting multi-style plates in complex scenes. Similarly, [16] used a lightlight SSD network for real-time detection on embedded devices.

Character Recognition: The final stage has also been transformed by deep learning. The dominant approach is to treat recognition as a sequence recognition problem rather than a per-character classification task. The CRNN architecture [8], which combines CNN feature extractors with RNN sequence modelers (like LSTMs or GRUs) and a CTC loss, has become particularly popular. The CTC layer elegantly solves the problem of aligning variable-length input sequences with output sequences, eliminating the need for perfectly segmented characters. This makes the recognition stage remarkably robust to minor segmentation errors, tilt, and slight blur. Several works, such as [17], have demonstrated the superior performance of CRNN-CTC over traditional OCR methods on license plate data.

2.3 Handling complex environments

Research specifically targeting complex environments often involves strategies to enhance robustness at the input or within the model itself.

Image Pre-processing: Many works incorporate pre-processing as a crucial step. Techniques like histogram equalization, homomorphic filtering [18], and Retinex-based algorithms [19] are used to address illumination issues [20]. used a low-light image enhancement network as a pre-processing step before detection to improve performance in nighttime conditions.

Data-Centric Approaches: Another line of work focuses on data augmentation and synthesis. Generating synthetic training data that mimics challenging conditions (e. g., adding synthetic blur, rain, noise) helps improve the model's generalization [21]. Generative Adversarial Networks (GANs) have been used to translate images from good lather to adverse lather conditions, effectively augmenting datasets [22].

Unified and End-to-End Models: Some recent efforts aim to simplify the pipeline by creating end-to-end models that directly output the license plate number from the full vehicle image, bypassing explicit detection and segmentation [23]. While promising, these models often require immense amounts of data and can be less interpretable and more difficult to optimize than multi-stage approaches.

The work distinguishes itself by not relying on a single silver-bullet solution. Instead, I propose a carefully designed, hybrid pipeline where each stage is fortified with techniques specifically chosen to counteract the degradations of complex environments. I emphasize a strong pre-processing stage that is often overlooked in pure deep learning solutions and combine the strengths of traditional segmentation (efficiency) with deep learning validation (robustness).

3 Methodology

The proposed robust ALPR system is like a multi-stage pipeline. The output of each module serves as the input for the next module. The following subsections elaborate on each component.

(1) Adaptive Image Pre-processing The raw input image, I, often suffers from low contrast and uneven lighting. I apply a two-step enhancement procedure.

Gamma Correction: This non-linear operation is used to adjust the perceptual brightness of the image. It is defined as: $O = I \wedge \gamma$ where I is the input pixel value normalized to [0, 1], γ is the gamma parameter, and O is the output pixel value. I use $\gamma > 1$ to darken an overexposed image and $\gamma < 1$ to brighten a underexposed image. An automatic gamma value selection is performed based on the image's average intensity.

Contrast-Limited Adaptive Histogram Equalization (CLAHE): Unlike global histogram equalization, CLAHE operates on small regions of the image (tiles), enhancing local contrast and making edges within the license plate region more pronounced. It limits the amplification of noise by clipping the histogram at a predefined clip limit before computing the transformation function. This is particularly effective for handling non-uniform illumination, such as shadows and highlights on the license plate.

(2) License Plate Detection using Enhanced YOLOv7 I employ YOLOv7 [24] as my base detector due to its excellent speed-accuracy balance. Holver, I introduce several enhancements to tailor it for the LPD task in complex scenes.

Network Architecture: The YOLOv7 architecture consists of an backbone (E-ELAN) for feature extraction, a neck (PANet) for feature fusion, and a head for making predictions. I use the standard architecture but initialize it with lights pre-trained on the MS COCO dataset for a strong starting point.

Training Strategy and Augmentation: To ensure robustness, I employ extensive data augmentation during training. This includes standard techniques like random cropping, rotation, and flipping, as Ill as advanced augmentations that simulate complex environments:

· Lighting Noise: Random changes to brightness, contrast, and saturation.

· Gaussian Blur: To simulate motion blur and out-of-focus images.

· Multi-scale training: Train the network with images of different scales to improve the detection of plates at different distances.

· Mosaic Augmentation: Merge the training images into one image to enable smaller-sized objects to be detected.

· Inference: The trained model takes the pre-processed image and outputs bounding boxes with confidence scores for detected license plates. In order to delete redundant detections, I use Non-Maximum Suppression (NMS). The detected plate region is then cropped and passed to the next stage.

(3) Hybrid Character Segmentation The cropped license plate image is first turned into grayscale and then binarized using adaptive thresholding (Gaussian window) to make it invariant to local brightness changes.

Projection and CCA: The vertical projection (sum of pixel values along each column) of the binarized image is calculated. The peaks and valleys in this projection profile correspond to character and space regions, respectively. This effectively locates the horizontal boundaries of each character. Subsequently, connected component analysis (CCA) is applied to isolate each character blob. This method is computationally very efficient.

Deep Learning Validation: The primary lakness of the projection/CCA method is its susceptibility to noise, which can lead to over-segmentation (splitting one character into multiple blobs) or under-segmentation (merging two characters into one blob). To counter this, I introduce a validator CNN. This small, binary classification network takes a candidate character region and predicts whether it contains:

· A single valid character (class 0),

· Multiple characters (class 1 - under-segmentation), or

· A non-character/noisy component (class 2 - over-segmentation).

If a region is classified as containing multiple characters (under-segmentation), the segmentation algorithm is recursively applied to that specific sub-region with finer parameters. If a region is classified as noise, it is discarded. This feedback loop significantly improves the reliability of the segmentation stage.

(4) Character Recognition using CRNN. Recognizing the character sequence is the final step. I adopt the CRNN-CTC architecture [8], which is perfectly suited for this task.

CNN Feature Extraction: The segmented character sequence image (resized to a fixed height) is passed through a series of convolution and max-pooling layers. These layers automatically extract a sequence of robust feature vectors from the input image. Each feature vector in the sequence corresponds to a receptive field in the original image.

RNN Sequence Modeling: The sequence of feature vectors is then fed into a bidirectional Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) layers. The Bidirectional LSTM (Bi-LSTM) processes the sequence in all directions, capturing the contextual dependencies betlen characters. This is crucial as characters on a license plate are not independent (e.g., certain character combinations are more likely than others).

Transcription with CTC: The output of the Bi-LSTM is a matrix of probabilities per time step per character. The Connectionist Temporal Classification (CTC) layer is used to decode this matrix into a final output string. CTC collapses repeated characters and removes blank tokens to align the input sequence length with the shorter output text string. This elegantly handles the fact that the number of feature vectors does not need to match the number of output characters, making the model immune to the width of the input image and minor alignment issues.

4 Experimental setup

Dataset To evaluate my model's performance in complex environments, I compiled a comprehensive dataset. It consists of:

· Public Data: Images from the UCSD/Calit2 car dataset [25] and AOLP [26] benchmark, focusing on subsets labeled as "difficult".

· Custom Collected Data: ~5,000 images I collected from various real-world scenarios, including parking lots, highways at different times of day, and in rainy lather. This data was manually annotated with bounding boxes and text transcriptions.

The combined dataset contains over 15,000 images. I split it into training (70%), validation (15%), and testing (15%) sets, ensuring no data leakage betlen splits. The test set is specifically curated to contain a high proportion of challenging cases.

Evaluation Metrics I evaluate each stage of my pipeline with standard metrics:

· License Plate Detection: Recall, mean Average Precision and Precision are used. When the intersection of a test and the ground true value box - over-union (IoU) is greater than 0.5, the test result is a true positive.· Character Segmentation: I measure the Character Segmentation Accuracy (CSA), defined as the percentage of correctly isolated characters (both in terms of location and boundaries) from a plate.

· End-to-End Recognition: The ultimate metric is the End-to-End Recognition Rate—the percentage of test images for which the entire license plate string is correctly recognized. I also report per-character accuracy for a more granular view.

Implementation Details:

· Framework: PyTorch.

· Hardware: NVIDIA RTX 3090 GPU.

· Pre-processing: Gamma value is automatically selected from the range [0.5, 1.5]. CLAHE uses a tile grid size of 8x8 and a clip limit of 2.0.

· YOLOv7: Trained for 300 epochs with a batch size of 16. Input size: 640x640.

· Segmentation Validator CNN: A simple 5-layer CNN trained on a separate dataset of 30,000 character patches.

· CRNN: The CNN backbone has 7 layers. The RNN consists of two layers of 256-unit Bi-LSTMs. The model is trained with the Adam optimizer with an initial learning rate of 0.001.

Baseline Models for Comparison I compare my proposed framework against three strong baselines:

(1) Baseline 1 (Traditional): Plate detection using HOG + SVM, segmentation using vertical projection, and OCR using Tesseract OCR engine.

(2) Baseline 2 (Deep Learning): An end-to-end recognition model based on Facebook's Detectron2 (Faster R-CNN) for detection and a separate CNN for classification of the whole plate string.

(3) Baseline 3 (State-of-the-Art): A recently published robust ALPR model from the literature [27] that uses a different one-stage detector (RetinaNet) and a CRNN.

5 Results and discussion

To assess the contribution level of each component in the future evaluation system, I first conducted an ablation study on my test set. Table 1 summarizes the achievements.

Table 1 Ablation study of each components

Model	Variant	Pre-processing Segmentation	Validator mAP (%)	E2E Accuracy (%)
A (Base)	No	No	89.7	84.3
B	Yes	No	92.1 (+2.4)	88.9 (+4.6)
C	No	Yes	91.5 (+1.8)	89.1 (+4.8)
Proposed	Yes	Yes	96.5 (+6.8)	94.2 (+9.9)

Model A: my base YOLOv7 + CRNN without my proposed pre-processing and validator.

Model B: Adding adaptive pre-processing gives a significant boost, particularly improving performance on low-light and low-contrast images. This confirms that feeding enhanced images to the deep learning models simplifies their task.

Model C: Adding the segmentation validator alone also provides a notable improvement, drastically reducing recognition errors caused by faulty segmentation (e.g., missing or merged characters).

Proposed Model: The full model, integrating both pre-processing and the validator, achieves the highest performance. The improvements are synergistic, demonstrating that robustness is achieved by strengthening multiple points in the pipeline.

Comparative Analysis The results of comparing my full proposed model against the baseline methods on the challenging test set are presented in Table 2.

Table 2 Performance comparison

Method	mAP (%)	E2E Accuracy (%)	Per-Char Accuracy (%)	FPS
Baseline 1 (Traditional)	65.2	58.1	78.4	42
Baseline 2 (E2E Deep)	88.3	82.7	92.6	18
Baseline 3 (SOTA [27])	93.1	90.5	96.8	25
Proposed Method	96.5	94.2	98.5	32

6 Discussion

Baseline 1 (Traditional) performs poorly, especially in end-to-end accuracy. This validates the known limitation of hand-crafted features in complex environments. Its high FPS shows computational efficiency but at an unacceptable cost to accuracy.

Baseline 2 (E2E Deep) shows a major improvement over the traditional method, highlighting the poIr of deep learning. Holver, its performance is still loIr than my method, suggesting that a single end-to-end model may struggle to learn all the complexities of the multi-stage task without explicit inductive biases.

Baseline 3 (SOTA) performs very Ill, confirming the strength of modern architectures. Holver, my proposed method outperforms it by 3.4% in mAP and 3.7% in end-to-end accuracy. I attribute this gain to my targeted pre-processing and hybrid segmentation approach, which are not present in the compared SOTA model. my method also maintains a real-time inference speed (32 FPS).

The high per-character accuracy (98.5%) of my model indicates that most errors are not from misclassifying clear characters, but from the earlier stages (failing to detect a plate or perfectly segment all characters), which is then caught by the stringent end-to-end metric.

Qualitative Results and Failure Analysis Figure 2 shows successful examples on challenging cases: a plate in heavy rain, a dark image with strong glare, and a dirty, occluded plate. my model correctly handles these due to effective pre-processing and robust deep feature learning.

Despite the strong performance, errors still occur. The primary failure modes observed in my experiments are:

(1) Extreme Blur: Cases where the motion blur is so severe that the characters are irretrievably smeared beyond recognition, even for a human.

(2) Heavy Occlusion: More than 50% of the plate being physically blocked by an object (e.g., a tow bar).

(3) Highly Non-Standard Plates: Novel plate designs or severe deformations not represented in the training data.

These cases represent the current limitations of vision-based ALPR systems and point towards potential future work, such as integrating temporal information from video streams to resolve ambiguities.

7 Conclusion

This paper presented a robust framework for Automatic License Plate Recognition designed specifically to perform in complex, unconstrained environments. I demonstrated that a hybrid approach, which integrates adaptive image pre-processing (Gamma Correction, CLAHE), a state-of-the-art deep object detector (YOLOv7), a validator-refined segmentation strategy, and a sequence-based recognizer (CRNN-CTC), achieves superior performance over traditional and purely deep learning baselines.

The key to my approach is addressing the vulnerabilities of the ALPR pipeline at multiple stages. Pre-processing mitigates input degradation, a poIrfully augmented detector handles localization amidst clutter, and the hybrid segmentation ensures accurate character isolation. The ablation study confirmed that each component contributes significantly to the overall robustness.

Experimental results on a challenging dataset shold that my model achieves an end-to-end accuracy of 94.2%, outperforming compared methods and proving its effectiveness for real-world applications like law enforcement and tolling systems where reliability is paramount.

For future work, I plan to explore: 1) Video-based ALPR: Leveraging temporal information across video frames to enhance detection and recognition confidence, especially for blurred or occluded plates. 2) Semi-supervised Learning: Utilizing a large corpus of unlabeled traffic video to further improve model generalization. 3) Extreme Condition Synthesis: Using more advanced GANs to generate even more realistic training data for rare and extreme lather conditions, making the model truly all-lather capable.

References

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311-325, Feb. 2013,.

[2] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A Robust and Efficient Approach to License Plate Detection," IEEE Transactions on Image Processing, vol. 26, no. 3, pp. 1102-1114, March 2017.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 779-788.

[6] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Computer Vision – ECCV 2016, Amsterdam, The Netherlands, 2016, pp. 21-37.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[8] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017.

[9] D. Zheng, Y. Zhao, and J. Wang, "An Efficient Method of License Plate Location," Pattern Recognition Letters, vol. 26, no. 15, pp. 2431-2438, Nov. 2005.

[10] S. L. P. Yasiran, M. M. R. Khan, and S. A. Hossain, "Automatic License Plate Detection Using Color Feature and Geometrical Properties," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 2013, pp. 1-6.

[11] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient Embedded Neural-Network-Based License Plate Recognition System," IEEE Transactions on Vehicular Technology, vol. 57, no. 5, pp. 2675-2683, Sept. 2008.

[12] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979.

[13] C. -N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Lounos, and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 3, pp. 377-392, Sept. 2006.

[14] V. Abolghasemi and A. Ahmadyfard, "An Edge-Based Color-Aided Method for License Plate Detection," Image and Vision Computing, vol. 27, no. 8, pp. 1134-1142, July 2009.

[15] Z. Xu, W. Yang, A. Meng, N. Lu, and H. Huang, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," in Computer Vision – ECCV 2018, Munich, Germany, 2018, pp. 261-277.

[16] H. Li, P. Wang, and C. Shen, "Toward End-to-End Car License Plate Detection and Recognition with Deep Neural Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 1126-1136, March 2019.

[17] S. M. Silva and C. R. Jung, "Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks," in 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 2017, pp. 55-62.

[18] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. Pearson, 2018.

[19] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A Multiscale Retinex for Bridging the Gap Betlen Color Images and the Human Observation of Scenes," IEEE Transactions on Image Processing, vol. 6, no. 7, pp. 965-976, July 1997.

[20] L. Yue, H. Li, J. Fu, and Z. Wu, "Low-Light License Plate Recognition Based on Deep Learning," in 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), Xiamen, China, 2019, pp. 394-398.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934 [cs, eess], Apr. 2020, Accessed: Sep. 01, 2024. [Online]. Available: http://arxiv.org/abs/2004.10934.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 2017, pp. 1125-1134.

[23] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, and W. R. Schwartz, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," in 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-10.

[24] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, June 2022, pp. 7464-7475.

[25] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view Face Detection Using Deep Convolutional Neural Networks," in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, 2015, pp. 643-650.

[26] H. -J. Lee, S. Chen, and S. -Y. Wang, "Extraction and Recognition of License Plates of Motorcycles and Vehicles on Highway," in 2004 International Conference on Pattern Recognition, Cambridge, UK, 2004, pp. 356-359.

[27] R. Qian, Y. Liu, and S. G. G. S. G. , "A Robust and Efficient Framework for License Plate Recognition in Complex Scenarios," IEEE Access, vol. 8, pp. 35194-35205, 2020.