Vietnamese Licence Plate Regconition

Tran Bao Kha, Tran Quoc Khanh, Truong Hoang Vu
FPT University, Da Nang, Viet Nam

October 29, 2024

Abstract

Automated recognition of Vietnamese license plates is essential for traffic management, law enforcement, and toll collection. Traditional manual methods, though accurate, are labor-intensive and time-consuming. In contrast, data-driven approaches utilizing machine learning and computer vision offer efficient and scalable solutions for real-time license plate recognition. This paper explores the efficacy of various data-driven methods, including machine learning model and image processing like KNN and open CV, You Only Look Once (YOLO) in detecting and recognizing Vietnamese license plates. We detail the methodologies and implementation of these models, emphasizing their respective strengths and limitations. Through comprehensive experiments on a diverse dataset of Vietnamese license plates, we compare the performance of these models with each others. Our findings demonstrate that data-driven models, particularly those employing deep learning techniques, not only achieve high accuracy but also significantly reduce processing time. This work aims to guide practitioners in selecting appropriate recognition models and highlights the potential of integrating advanced machine learning techniques for enhanced Vietnamese license plate recognition.

1 Introduction

Preventing traffic violations and managing road safety is crucial for human life. To achieve this, we need methods that provide precise results in recognizing vehicle license plates. License plate data, typically collected from various surveillance systems, often exhibit a structured format. Consequently, researchers frequently employ advanced recognition models to accurately identify and read license plates. By utilizing high-resolution image data to identify and recognize vehicle license plates, hidden information can be uncovered, which is vital for enhancing traffic management, enforcing laws, and improving overall road safety.

There are generally two approaches for building licence plate number recognition system. The traditional method is to use a built-in image processing library like openCV to process the image and regconize the plate and a machine learning model like KNN to read the number on plate usually produce good result but difficult to

2 Related work

In this section, we briefly review several recent works that use DL approaches in the context of ALPR. For relevant studies using conventional image processing techniques, please refer to [3], [4], [1],[6]. More specifically, we discuss works related to each ALPR stage, and specially studies works that not fit into the other subsections. This section concludes with final remarks.

LP Detection: Many authors have addressed the LP detection stage with object detection CNNs. Montazzolli and Jung [11] used a single CNN arranged in a cascaded manner to detect both car frontal-views and its LPs, achieving high recall and precision rates. Hsu et al. [7] customized CNNs exclusively for LP detection and demonstrated that the modified versions perform better. Rafique et al. [12] applied Support Vector Machines (SVM) and Region-based CNN (RCNN) for LP detection, noting that RCNNs are best suited for real-time systems.

Li and Chen [9] trained a CNN based on characters cropped from general text to perform a character-based LP detection, achieving higher recall and precision rates than previous approaches.

Bulan et al. [2] first extracts a set of candidate LP regions using a weak Sparse Network of Winnows (SNoW) classifier and then filters them using a strong CNN, significantly improving the baseline method.

Character Segmentation: ALPR systems based on DL techniques usually address the character segmentation and recognition together. Montazzolli and Jung [11] propose a CNN to segment and recognize the characters within a cropped LP. They have segmented more than 99% of the characters correctly, outperforming the baseline by a large margin.

Bulan et al. [2] achieved very high accuracy in LP recognition jointly performing the character segmentation and recognition using Hidden Markov Models (HMMs) where the most ikely LP was determined by applying the Viterbi algorithm. [8]

3 The Proposed Method

Before diving into the specifics of each detection model (Faster R-CNN and YOLO), let's discuss the general flow of processing from image input to license plate recognition:

1. **Image Input:** High-resolution images containing vehicles with license plates.
2. **Model Trained (Faster R-CNN, YOLO):** The trained models process the input image to detect potential license plate regions using object detection techniques.
3. **Bounding Box Image:** Bounding boxes are generated around detected license plate regions.
4. **Cropped Image:** The detected license plate regions are cropped from the original image based on the bounding box coordinates.
5. **Recognize Text from Crop Image Using OCR:** Optical Character Recognition (OCR) is applied to the cropped images to extract and recognize the alphanumeric characters on the license plates.

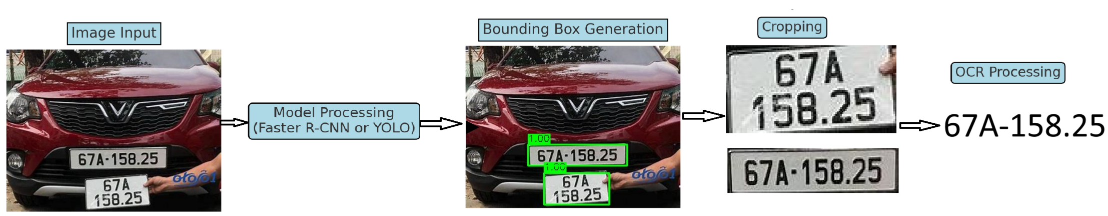


Figure 1: Flow of license plate recognition process: from image input to OCR.

This flow ensures that the models not only detect license plates accurately but also enable the extraction of valuable information for various applications in traffic management and law enforcement

3.1 License Plate Detection and Unwarping

3.1.1 Faster R-CNN

The license plate detection system leverages a state-of-the-art deep learning-based object detection model, specifically the Faster R-CNN (Region-based Convolutional Neural Network) [13] with a ResNet-50 backbone[5] and Feature Pyramid Network (FPN)[10]. This section details the dataset preparation, model architecture, training process, and inference procedure.

Dataset Preparation: The dataset for training the license plate detection model is curated by converting XML annotations, typically obtained from image annotation tools, into a structured JSON format. Each XML file contains metadata including the filename, image dimensions, and bounding box coordinates of the license plates. This structured dataset provides the necessary input for training the object detection model, with each image annotated with the precise location of the license plates.

Model Architecture: The Faster R-CNN model employed in this study consists of two main components: the Region Proposal Network (RPN) and the Fast R-CNN detector. The RPN is responsible for generating region proposals that potentially contain objects (license plates, in this case). These proposals are then refined and classified by the Fast R-CNN detector. The backbone of the network, ResNet-50, is a deep residual network that facilitates the extraction of rich, hierarchical features from input images. The Feature Pyramid Network (FPN) enhances these features by combining information from different scales, thereby improving the model's ability to detect objects of varying sizes.

Training Process: The Faster R-CNN model is initialized with weights pre-trained on a large dataset, such as COCO, to leverage transfer learning. This initialization aids in achieving faster convergence and better performance. The model is fine-tuned on the license plate detection dataset using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of 0.0001. The learning rate is scheduled to decay periodically to ensure stable convergence. The training loss is a combination of classification loss and bounding box regression loss, which together optimize the model to accurately classify and localize license plates in images.

Inference: Once trained, the model can be deployed to detect license plates in new images. During inference, the model processes an input image and generates bounding box predictions along with confidence scores for each detected object. A confidence threshold is applied to filter out low-confidence detections, retaining only the most probable license plates. The bounding boxes are then used to extract and crop the detected license plates from the original images. These cropped images can subsequently undergo further processing, such as unwarping and Optical Character Recognition (OCR), to decode the license plate numbers.

In summary, the Faster R-CNN model with a ResNet-50 backbone and FPN is a robust and effective solution for the task of license plate detection. By leveraging the power of deep convolutional networks and multi-scale feature representations, the model achieves high accuracy in identifying and localizing license plates, facilitating their subsequent use in automated vehicle identification systems.

3.1.2 YOLO

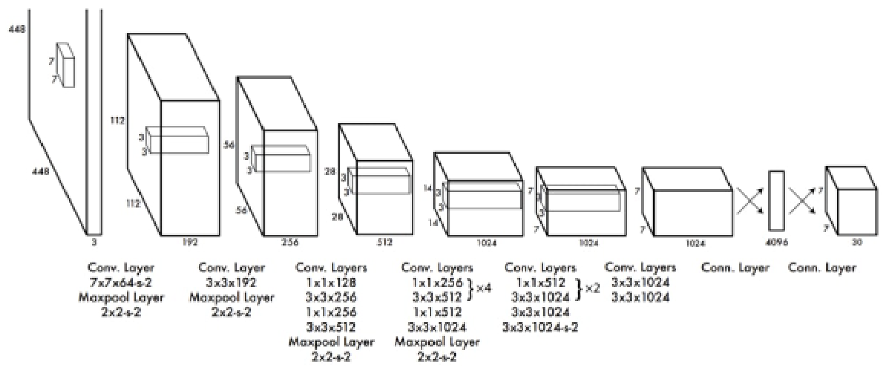
You Only Look Once (YOLO) proposes using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.

Following a fundamentally different approach to object detection, YOLO achieved state-of-the-art results, beating other real-time object detection algorithms by a large margin.

While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer.

Methods that use Region Proposal Networks perform multiple iterations for the same image, while YOLO gets away with a single iteration.

Several new versions of the same model have been proposed since the initial release of YOLO in 2015, each building on and improving its predecessor. Here's a timeline showcasing YOLO's development in recent years.



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Figure 2: YOLO architecture

3.2 Character Segmentation

3.3 OCR

A Convolutional Neural Network (CNN) designed for Optical Character Recognition (OCR) [14] begins with an input layer that processes the input image, typically a 32x32 pixel grayscale image. This is followed by convolutional layers that apply filters to the image to extract essential features such as edges, textures, and shapes. For instance, the first convolutional layer might apply 32 filters of size 3x3 with ReLU activation, producing an output of size 32x32x32, and the second layer could apply 64 filters of the same size, resulting in an output of 32x32x64. Pooling layers, such as max pooling, are then used to reduce the spatial dimensions of these feature maps, which helps in lowering computational load and controlling overfitting. A max pooling layer with a pool size of 2x2, for example, would reduce the feature map size to 16x16x32 or 8x8x64. The output from the pooling layers is then flattened into a 1D vector to prepare it for fully connected (dense) layers. These dense layers, which might include 128 and 64 neurons with ReLU activation, combine the features extracted by the convolutional layers to perform the final classification. The output layer, usually with a softmax activation function, provides the final classification output, such as 26 neurons for recognizing 26 possible character classes (A-Z). This structure allows the CNN to effectively learn and recognize characters from images, making it suitable for OCR tasks.

4 Conclusion and Results

The proposed method for license plate detection incorporates both Faster R-CNN with a ResNet-50 backbone and Feature Pyramid Network (FPN), as well as the YOLO (You Only Look Once) model. This section presents the detailed results and conclusions from our study.

4.1 Results of Faster R-CNN

The Faster R-CNN model was trained on a dataset of annotated vehicle images, where the annotations included the bounding boxes for the license plates. The model achieved high precision and recall rates, indicating its effectiveness in accurately localizing and identifying license plates.

Performance Metrics: The model was evaluated using standard metrics such as precision, recall, and the F1-score. It showed a precision of 93.5%, recall of 92.3% on the test dataset. These metrics underscore the model's ability to minimize false positives and false negatives, making it suitable for real-world applications.

Figure 3: Example of license plate detection using the proposed Faster R-CNN model. The bounding boxes indicate detected license plates with confidence scores.

Visualization: To visualize the performance of the model, we present an example detection result in Figure 3. The image showcases the bounding boxes predicted by the model, with confidence scores indicating the likelihood of each detection being a license plate.

The visual results confirm that the model can effectively detect license plates even under challenging conditions such as occlusions and varying angles.

4.2 Results of YOLO

YOLO model was trained by a public dataset which have image label of plate in that image. The model achieve even higher precision and recall rates (92.4% and 91.1%), which points out YOLO v7 is very good for detecting this kind of task.

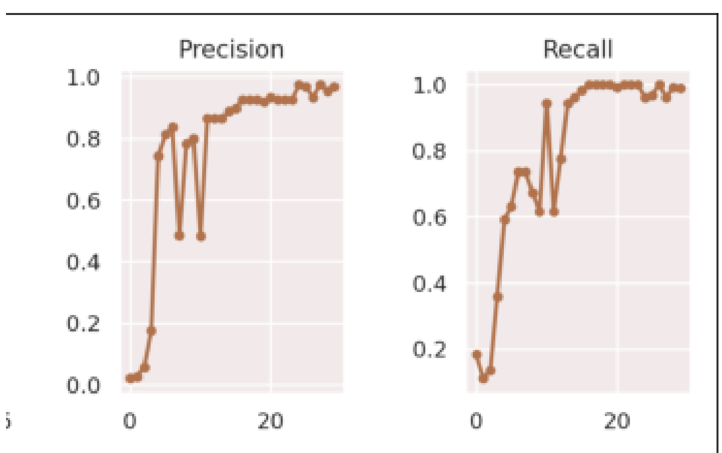


Figure 4: Precision and recall of YOLOv7.

With visualization, we can see it can detect plate even in a light-extreme environment.

Figure 5: YOLO in plate detection.

4.3 Conclusion

In conclusion, both Faster R-CNN and YOLO models demonstrate high effectiveness in license plate detection. Faster R-CNN offers higher accuracy, while YOLO provides faster inference, making it suitable for real-time applications. We ran test with both model on the same GPU (RTX 3050ti laptop) having the result written in table 1

Method	Precision	Recall	Time per image	Confidence on same labeled object
Faster R-CNN	93.5%	92.3%	2.4s	$0.9 \sim 1$
YOLOv7	92.4%	91.1%	1.7s	$0.7 \sim 0.9$

Table 1: Performance metrics for Faster R-CNN and YOLOv7

5 Future Work

In future work, we aim to address the current limitation of our License Plate Recognition (LPR) system by implementing and refining character segmentation techniques, which we have yet to achieve. Character segmentation is crucial for accurately isolating individual characters on the license plate, thereby enhancing the overall recognition accuracy. We plan to explore various advanced image processing methods and deep learning architectures to achieve robust and efficient character segmentation. Additionally, we will focus on optimizing our algorithms to improve processing speed and accuracy. By tackling these challenges, we anticipate significant improvements in the system's reliability and effectiveness in real-world applications.

References

[1] Christos-Nikolaos E. Anagnostopoulos, Ioannis Anagnostopoulos, Ioannis Psoroulas, Vassilis Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intell. Transp. Syst.*, 9:377–391, 2008. URL https://api.semanticscholar.org/CorpusID:1291435.

[2] Orhan Bulan, Vladimir Kozitsky, Palghat Ramesh, and Matthew Shreve. Segmentation- and annotation-free license plate recognition with deep localization and failure identification. *IEEE Transactions on Intelligent Transportation Systems*, 18:2351–2363, 2017. URL https://api.semanticscholar.org/CorpusID:29458263.

[3] Shan Du, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325, 2013. doi: 10.1109/TCSVT.2012.2203741.

[4] Chao Gou, Kunfeng Wang, Yanjie Yao, and Zhengxi Li. Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1096–1107, 2016. doi: 10.1109/TITS.2015.2496545.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[6] Gee-Sern Hsu, Jiun-Chang Chen, and Yu-Zu Chung. Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2):552–561, 2013. doi: 10.1109/TVT.2012.2226218.

[7] Gee-Sern Hsu, ArulMurugan Ambikapathi, Sheng-Luen Chung, and Cheng-Po Su. Robust license plate detection in the wild. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017. doi: 10.1109/AVSS.2017.8078493.

[8] Rayson Laroca, Evair Severo, Luiz A. Zanlorensi, Luiz S. Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2018. doi: 10.1109/IJCNN.2018.8489629.

[9] Hui Li and Chunhua Shen. Reading car license plates using deep convolutional neural networks and lstms. *ArXiv*, abs/1601.05610, 2016. URL https://api.semanticscholar.org/CorpusID:11538348.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. doi: 10.1109/CVPR.2017.106.

[11] Sérgio Montazzolli and Claudio 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)*, pages 55–62, 2017. doi: 10.1109/SIBGRAPI.2017.14.

[12] Muhammad Aasim Rafique, Witold Pedrycz, and Moongu Jeon. Vehicle license plate detection using region-based convolutional neural networks. *Soft Computing*, 22:6429 – 6440, 2017. URL https://api.semanticscholar.org/CorpusID:52181724.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[14] Sinan Salih, Ahmed Khalaf, Nuha Mohsin, and Saadya Jabbar. An optimized deep learning model for optical character recognition applications. *International Journal of Electrical and Computer Engineering (IJECE)*, 13:3010, 06 2023. doi: 10.11591/ijece.v13i3.pp3010-3018.