Theory and Practice of Science and Technology
2025, VOL. 6, NO. 2, 40-44
DOI: 10.47297/taposatWSP2633-456909.20250602

CNN-based lane detection

Yanliang Liu

Wuhan University of Technology, Wuhan, Hubei, 430000, China

ABSTRACT

With the continuous advancement of intelligent transportation systems, lane detection technology plays a crucial role in enhancing road safety and promoting the development of autonomous driving. This study focuses on implementing an efficient and accurate CNN-based lane detection algorithm to address the challenges posed by complex and variable road environments. By leveraging the powerful pattern recognition capabilities of deep learning, we have developed a customized CNN model capable of accurately detecting lane markings under different lighting conditions, weather situations, and road surface qualities. In this project, I adopted the TuSimple lane detection dataset and preprocessed it to meet the requirements for model training. I chose YOLO as the base model architecture and made optimization adjustments to ensure the effectiveness and generalization ability of the model. Extensive experimental validation was also performed to confirm these aspects.Experimental results show that the proposed CNN model performs as expected in lane detection tasks, achieving high-precision detection results while maintaining good real-time processing performance. The model exhibits robustness when facing harsh weather and complex road conditions. Additionally, by comparing the performance differences under various parameter configurations, we further explored the key factors influencing model performance.In summary, this study presents a novel and effective solution for lane detection. Future work will be dedicated to exploring more improvement strategies, such as integrating multi-sensor information fusion, to achieve higher detection accuracy and broader application scope.

KEYWORDS

Convolutional Neural Networks; Lane Detection; Autonomous Driving; Image Processing

1 Introduction

These guidelines, The prospects for intelligent vehicles and autonomous driving in China are very broad, with this field undergoing rapid technological progress and market expansion. With the continuous maturation of sensor technology, computing platforms, and artificial intelligence algorithms, more powerful autonomous driving technologies are gradually achieving commercialization. Meanwhile, the application of 5G communication technology makes real-time information exchange between vehicle-to-vehicle, vehicle-to-infrastructure, and vehicle-to-cloud possible, greatly enhancing the intelligence level of transportation systems.

At the same time, the Chinese government places great importance on the development of intelligent vehicles, promoting technological innovation and industrialization in this field through a series of policies and plans. For example, policy documents such as "Made in China 2025", "Outline for Building a Strong Transportation Nation", and "New Generation Artificial Intelligence Development Plan" have clearly outlined the development direction and goals for intelligent vehicles. Multiple cities and regions have launched road tests and demonstration projects for intelligent connected vehicles. These pilot projects not only accelerate the maturity and application of the technology but also provide a basis for formulating relevant laws, regulations, and standards.

For these reasons, according to forecasts by multiple research institutions, the global market size for intelligent connected vehicles is expected to reach $1.5 trillion by 2030. At the same time, with the continuous popularization of autonomous driving technology, consumer acceptance is also increasing. Many consumers are willing to pay extra fees for high-level autonomous driving features, which will further drive market development. As the world's largest automotive market, China's rapid development in the field of intelligent connected vehicles will have a significant impact on the global market.

As a burgeoning industry, the development of intelligent vehicles and autonomous driving still faces multiple challenges. The safety of autonomous driving systems is the primary concern. Any technical failure can lead to serious accidents; therefore, it is necessary to ensure the normal operation of autonomous driving through program design, which is one of the significances of this study.

This research aims to implement a simple and effective lane detection system that utilizes computer vision technology to achieve real-time road environment perception. By analyzing the characteristics of lane lines in video streams, a set of detection algorithms has been realized. This approach not only has wide applications in the field of vehicle autonomy but also holds significant importance for traffic safety.

Currently, Convolutional Neural Networks (CNNs) have been widely applied in the field of lane line recognition, achieving very high accuracy. Through the use of deep network structures such as ResNet, VGGNet, and specially designed lightweight networks like ENet and ERFNet, researchers have been able to enhance the robustness of lane line detection under various environmental conditions. Moreover, for autonomous driving and Advanced Driver Assistance

Systems (ADAS), real-time performance is critical. Lightweight CNN architectures such as MobileNet and ShuffleNet can provide sufficient real-time performance while ensuring accuracy, meeting the demands of real-time driving. Semantic segmentation techniques have also been used to precisely distinguish lane lines from other road elements in images. U-Net, DeepLab, and other semantic segmentation networks have been successfully applied to the fine-grained detection of lane lines.

Existing research is not limited to pure lane line detection but also involves multi-task learning, such as simultaneously performing lane line detection, vehicle positioning, obstacle detection, etc. Such multi-task models can share underlying network features, thereby improving the overall efficiency and consistency of the system. To address the differences in lane line appearance at different distances and perspectives, researchers are exploring how to effectively fuse multi-scale and multi-view information. For instance, using pyramid structures or Feature Pyramid Networks (FPNs) to capture lane line features at different scales.

In summary, the application of CNNs in lane line recognition is in a stage of rapid development. With the continuous advancement of hardware acceleration technology and algorithmic innovations, future lane line recognition technology will become more precise, reliable, and capable of adapting to more complex driving environments.

The goal of this study is to establish a CNN-based lane detection system that, through training, can accurately identify lane lines on the road and provide precise location information. Based on existing datasets and models, combined with acquired knowledge, programming implementation is carried out.

2 Review

2.1 Technical Background of Lane Line Detection

Traditional image processing for lane line detection typically involves three stages: the first stage is to determine the region of interest, usually by directly cropping the top background; the second stage employs perspective transformation and other methods, such as converting color spaces and defining boundaries. The third stage uses algorithms like RANSAC (Random Sample Consensus) based on least squares or curve matching algorithms to determine and calibrate lane lines, with local correction operations added post-processing.

2.2 Existing Research and Challenges

Despite many successful cases, lane line detection still faces a series of challenges:

Environmental Diversity: Different weather conditions, lighting situations, and road surface conditions can all impact detection performance.

Computational Resource Constraints: To ensure real-time performance, it's necessary to consider the computational complexity and hardware requirements of the algorithm.

High Cost of Data Annotation: Obtaining high-quality annotated data is difficult and expensive, which poses higher demands on model training.

2.3 Advantages of the TuSimple Dataset

The TuSimple lane line dataset is created by Tusimple and is collected from highway scenes, including 3,626 training images and 2,782 test images. This dataset features varying degrees of occlusion, different types of lane markings, and diverse road conditions, covering traffic scenarios throughout the day. However, it does not include extreme weather or extreme scenarios, making the recognition difficulty moderate. It should be noted that in the TuSimple dataset, dashed lane lines are labeled as solid lines, and the annotations do not distinguish between different categories of markings.

As an important benchmark in the field of lane line detection, the TuSimple dataset provides rich annotation information, covering multiple actual driving scenarios. This enables researchers to evaluate different algorithms under unified standards, promoting rapid development in this field. Moreover, the diversity and challenge of the TuSimple dataset help train more robust models.

2.4 Reasons for Choosing the YOLO Model

The YOLO series of models are renowned for their fast inference speed and high detection accuracy, performing excellently in object detection tasks. Although YOLO was initially designed for general-purpose object detection rather than specific lane line detection, appropriate adjustments and optimizations make it suitable for this particular application scenario. Specifically, YOLO has several advantages:

Real-time Performance: YOLO can operate at extremely high frame rates, meeting the low-latency requirements of autonomous driving systems.

Ease of Use: Compared to other complex network architectures, YOLO is easier to implement and deploy.

Multi-scale Detection: YOLOv3 and subsequent versions support multi-scale predictions, which is particularly useful for capturing lane lines of different widths.

2.5 Significance of the Research

This study aims to explore how to build an efficient and accurate lane line detection system using the YOLO model combined with the TuSimple dataset. By analyzing existing issues and improvements, it is hoped that valuable references will be provided for future research.

3 Methodoloy

3.1 Data Preparation

Data Source:

The TuSimple public dataset is adopted for this project, which is specifically designed for lane detection tasks and contains a large number of images from real-world driving scenarios.

Data Preprocessing:

Road images captured by in-vehicle cameras contain not only lane information but also a substantial amount of irrelevant information, such as green belts, buildings, other vehicles, and the sky. Moreover, due to camera shake under dynamic conditions and external lighting, there can be considerable noise in the images. This irrelevant information can lead to excessive invalid computations, thus reducing the real-time performance of lane line recognition. To improve the efficiency and accuracy of the algorithm, image preprocessing before lane line recognition is necessary and commonly used.

A preliminary manual review of the dataset is conducted to mark out obviously unsuitable samples (e. g., severely blurred or heavily occluded images). YOLO's input layer requires a fixed size (such as 416x416 pixels), which helps accelerate the training process and increase inference speed. During resizing, padding techniques are used to maintain the original aspect ratio of the image, preventing distortion that could affect feature extraction. Pixel values are scaled to [0, 1] or [-1, 1] to reduce the impact of numerical range differences on gradient descent. Code is written to parse the annotation files provided by TuSimple, extracting each lane line's coordinate points. These coordinates are converted into the bounding box format required by YOLO, i.e., center point (x, y) and width and height (w, h), expressed as proportions relative to the image dimensions. Since TuSimple may contain multiple categories of lane lines, category ID mappings to YOLO's output layer need to be defined accordingly, updating the label information.

Data Augmentation:

From a geometric transformation perspective, random cropping of different parts of the image simulates different viewing angles; small-angle rotation and translation operations are applied while ensuring lane lines do not exceed the image boundaries; horizontal flipping increases data diversity, especially suitable for symmetric road scenes. From a color transformation perspective, changing the brightness and contrast of images simulates visual effects under different weather conditions. Adjusting hue and saturation simulates various lighting conditions. Introducing Gaussian noise or other types of interference moderately enhances the model's tolerance to actual environmental noise.

Using tools provided by the PyTorch framework, a class inheriting from torch.utils.data.Dataset is created to implement custom data reading logic. Parameters such as batch size, whether to shuffle the order, and the number of threads (num_workers) are set to optimize data flow efficiency. After completing all these steps, some samples are randomly selected to compare their original images with preprocessed versions, ensuring no errors or anomalies have been introduced. Preliminary tests ensure that the preprocessed dataset can smoothly go through the entire training process, including but not limited to model initialization, forward propagation, backpropagation, etc. Through detailed data preprocessing steps, not only the quality and diversity of data input into the YOLO model can be ensured, but also the stability of model training and the final prediction accuracy can be effectively improved. Data preprocessing is one of the keys to the success of deep learning projects and must be given sufficient attention and meticulous operation.

3.2 Model Selection and Configuration

YOLO Architecture:

YOLO is renowned for its fast inference speed, capable of achieving near-real-time detection speeds while maintaining high accuracy, which is critical for time-sensitive applications like lane line detection. It can effectively capture lane lines of different widths and shapes, enhancing detection flexibility. YOLO allows direct learning of features from raw images for object detection, reducing the need for manually designed feature extractors, simplifying the development process.

Additionally, YOLO has an active open-source community providing abundant resources and tools, facilitating quick start-ups and problem-solving.

Considering the pros and cons of different versions within the YOLO series, YOLOv5 was chosen as the base model. These versions introduce more advanced improvements, such as CSPNet, which enhances computational efficiency and expressive power, and PANet, which improves feature fusion effects, aiding in the precision of detecting small targets like slender lane lines. Anchor box sizes are automatically optimized according to dataset characteristics, making the model fit better to the actual distribution of lane lines.

Model Customization:

ImageNet pre-trained weights are used to initialize model parameters, accelerating convergence and improving final performance. The input size is set to 416x416 pixels, recommended by YOLO, which ensures adequate detail while avoiding excessive computational load. Given that the TuSimple dataset usually includes multiple categories of lane lines (such as left lane lines, right lane lines, etc.), the number of categories in the final layer output needs to be adjusted based on actual conditions. An initial learning rate is set at a lower value, gradually decaying over the course of training to prevent gradient explosion or oscillation without converging. Batch size is set to 8 to balance training speed and stability. Early stopping is employed based on validation set performance to avoid overfitting, dynamically adjusting as needed.

Special Adjustments for Lane Line Detection:

YOLO was originally designed for general object detection and requires specific adjustments when applied to lane line detection:

Utilizing the continuity and shape characteristics of lane lines, additional feature representation methods, such as orientation encoding and distance encoding, are explored to assist the model in understanding the context of lane lines; loss weights for different categories are dynamically adjusted based on their importance or frequency, making the model pay more attention to key areas; post-processing logic is formulated according to the physical properties of lane lines (such as continuity and curvature constraints) to correct abnormal detection results.

3.3 Training Process

Set hyperparameters such as learning rate, batch size, and iteration count. The initial learning rate can be set relatively low, gradually adjusting as training progresses (as described earlier). Define evaluation criteria suitable for lane line detection, such as precision, recall, F1 score, etc., and regularly test the model's performance on the validation set.

3.4 Testing and Optimization

After training is completed, evaluate the model's performance on an independent test set and record the results of various metrics. Conduct an in-depth analysis of failed cases to identify potential issues, such as false positives or false negatives. Based on error analysis results, further adjust and optimize the model, such as fine-tuning network architecture, improving data augmentation techniques, or adjusting hyperparameters.

4 Conclusion

4.1 Main Achievements

Model Performance:

After a series of experiments and adjustments, the YOLO model achieved a relatively high level of accuracy on the TuSimple test set, demonstrating the effectiveness of the YOLO architecture in handling lane line detection problems.

Real-time Performance:

Benefiting from YOLO's fast inference capability, the proposed system can maintain high accuracy while meeting the real-time requirements needed for autonomous driving applications.

Enhanced Robustness:

By making targeted optimizations to the model, such as introducing specific data augmentation strategies and adjusting the loss function, significant improvements were made in the model's adaptability to different lighting conditions, weather situations, and complex road conditions.

4.2 Comparison with Existing Methods

Compared to traditional rule-based methods, the YOLO-based lane line detection system of this project exhibits several advantages:

Faster Detection Speed, suitable for scenarios requiring immediate response;

Stronger Generalization Ability, better at handling unseen driving environments;
Lower Computational Resource Requirements, facilitating deployment on embedded devices.

4.3 Challenges and Limitations

Despite achieving positive results, the model's performance still has room for improvement under certain extreme conditions (such as severe occlusion or extremely harsh weather). Additionally, since YOLO was primarily designed for general object detection rather than lane line recognition, it may have deficiencies in capturing fine-grained features.

In summary, this study provides a feasible and efficient solution based on Convolutional Neural Networks (CNNs) for lane line detection. With the development of technology and continuous advancements in algorithms, I believe that lane line detection systems will become smarter and more reliable, contributing significantly to the development of autonomous driving technology.

References

[1] Xiong Lei, Yang Rou. (2022) Research on Lane Detection Technology Based on Deep Learning. https://kns.cnki.net/kcms2/article/abstract?v=l4Uq_U3qQ8Mz9uJEb3ZbRUazQSStv_ph8kMiQqJJWkW9riOMTsZSVfW0kfJ1He2kEn806qFdW

[2] Wang Hai. (2023) Research on Testing and Evaluation of Autonomous Driving Lane Detection Algorithms. https://kns.cnki.net/kcms2/article/abstract? v=l4Uq_U3qQ8P7I7tc0zqC1OheH3vLNnaPHUp6ivvEjqCCnkX0cIw--ER7Bf5QtlHV5U_K62BoC6sM1dQ288koahICK7QPtplB94OCaBhy-PcLcl2du2jlaW-CoMm0mxFmy3VhSg8YDNd7XS1GS2IB3RMDaa8p1F9UkDlZImkluK57TiZ63Fmy5-ozKJYQ9U75mUEXMkDYc7Y= &uniplatform=NZKPT&language=CHS

[3] Lu Pengcheng. (2024) Research on Lane Detection and Collision Warning Technology Based on Machine Vision.https://kns.cnki.net/kcms2/article/abstract?v=l4Uq_U3qQ8N_qmZzQONRi6brbtGdscckvCiJAx3T9hvYU3eZagE2tAgDxZ7MfPAyQBzgj7SNbxvkGV0emGcJyJ7YFSielwTAYR6V9wpUzGDR3dOorr_sgLuKnGVT3Uu3VTdkt_T339sXcuDR_yab3RG070Anuip__dv9KMwgBdjN_rAPVO-L15MAftD9LDvi&uniplatform=NZKPT&language=CHS