Research on lane detection algorithms based on GTNs

Zhenfang Wei

School of Software Engineering, Southwest University, Chongqing, China

niuhuang@ldy.edu.rs

Abstract. Lane detection technology plays an important role in the lane departure warning and adaptive cruise control functions of autonomous vehicle systems. Traditional lane line detection includes experimental steps such as preyreatment, color space conversion, feature extraction, and lane line tracking, but there are still some problems such as recognition accuracy in complex environments and parameter adjustment dependency. To overcome these limitations, this study uses a novel method that can directly predict the parameters of the lane shape model, thus avoiding complex post-processing steps. This research method uses the network structure based on Graph Transformer Networks (GTNS) to capture richer structural information and contextual relationships, thereby improving the accuracy of detection. In terms of feature extraction, Graph Transformer Networks is used to accurately capture the structural information of the lane through the attention mechanism. At the same time, the results in the later stage show that our method has better advantages in accuracy and speed. In short, this optimization scheme not only improves the detection speed but also ensures a certain degree of accuracy. The method will perform better in the future after further optimization such as multi-sensor fusion and real-time optimization.

Keywords: Lane Detection Technology, Learning Rate Scheduling, Attention Mechanism, Lane Line Tracking.

1. Introduction

Lane line detection is an important basic module in automatic driving. Based on the influence of different road environment, vehicle occlusion, camera position and other objective factors, as well as the flat shape of the lane line itself, it brings challenges to the task of lane line detection. Based on the above characteristics, it is necessary to have strong recognition ability, running speed and context recognition ability to adapt to complex environmental needs. These methods often lack understanding of the overall structure and context of the lane, limiting their performance in complex scenarios.

In traditional lane detection techniques, the first step is to identify key areas through image processing techniques such as filtering and noise reduction. Custom-designed feature extraction algorithms are then used to capture the details of lane edges. By fitting these features with curves, the outline of the lane is reconstructed [1]. As proposed by Borkar A et al., a lane detection technique has been developed that accurately tracks lane markings and optimizes parameters through inverse perspective transformation, threshold segmentation, Hough transform, RANSAC fitting, and Kalman filtering [2]. However, such traditional methods perform poorly under conditions of changing light, adverse weather, or unclear road markings. Furthermore, they are typically computationally intensive, making it difficult to meet real-time requirements. To address the issue of poor performance in complex environments, Ke [3] utilizes the Sobel operator and sliding window to detect lane lines, transforming yellow and white lane lines in complex environment. Recently, Lucas Tabelini [4] and his team have introduced a new method that redefines the lane detection task through polynomial regression, enhancing processing efficiency. However, this method, due to its failure to fully consider the global contextual information, still has room for improvement in accuracy compared to other techniques.

To enhance the accuracy and efficiency of lane detection, this study proposes a novel hybrid model architecture that combines the strengths of Graph Neural Networks (GNNs) and Transformers. In the new architecture, the model takes structured road data as input. The framework defines the output of lane detection as parameters of the lane shape model, such as the camera pitch angle and the overall lane structure. It utilizes GNNs to process the input graph-structured data, capturing the connections between nodes and the geometric properties of the road. Introducing the Transformer architecture to process sequence data related to lane detection allows the model to learn more general feature representations by sharing parameters between GNNs and Transformers, thereby enhancing its generalization capabilities. End-to-end training is conducted with Hungarian loss. Experiments are conducted using samples from different weather environments to ensure the adaptability of the model for various scenarios.

2. Research Method

2.1. Overall Framework

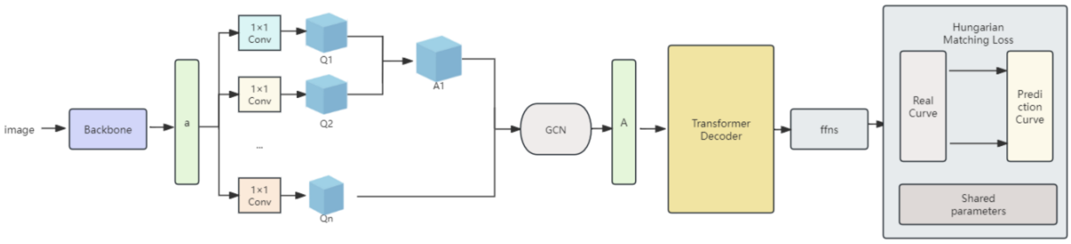

Figure 1. The basic framework diagram of the algorithm.

As shown in Figure 1, the basic framework of this algorithm is illustrated. Backbone: In constructing our network architecture, the paper adopted an optimized version of ResNet18 as our main feature extraction backbone [5]. To adapt to the lane detection task and reduce the risk of over-fitting, the paper adjusted this architecture by reducing the number of channels and the sampling rate. The aim is to preserve the key structural features of lanes while minimizing unnecessary details, thereby optimizing computational efficiency. Then, the subsequent constructed sequence is used as input to the encoder.

Encoder: Using the powerful ability of Graph Transformer Networks (GTNs) in processing structured data, integrate GTNs into the encoding layer to capture multi-dimensional relationships in data more effectively. This study uses the GCN [6]and HAN [7] as GNN based methods. GTNs learn meta-paths for given data and tasks and operate graph convolution on the learned meta-path graphs.

The first layer of this GTNs is the adjacency matrix in the set of gt layer softmax image matrix. A new meta-path graph A1 is obtained by matrix multiplication of the selected adjacency matrices Q1 and Q2 [8]. The adjacency matrix is derived from this formula

$$Q = F\left(A; W_{\varphi}\right) = \varphi\left(A; \text{softmax}\left(W_{\varphi}\right)\right) \tag{1}$$

That is to get the weighted sum of non-negative weights of the matrix through 1*1 convolution, and then stack multiple layers to further abstract and integrate node features. Using the softmax function to normalize it can meet the requirements of the probability distribution of the output, and also provide an effective numerical stability to avoid gradient disappearance or gradient explosion problems when dealing with large or small values. Multiple node representations from the same GCNs on multiple meta-path graphs are integrated by concatenation and improve the performance of node classification and it is beneficial to learn different node representations via multiple different graph structures [5].

FFNs (Feed-Forward Networks) and Hungarian Fitting Loss:

Employing Hungarian loss for bipartite matching between predicted parameters and actual lanes, and further optimizing the data using the matching results.

2.2. Adjusting the Learning Rate Dynamically

This study employs a cosine annealing strategy, which starts with a high learning rate and then gradually reduces this rate through multiple cycles of cosine function adjustments. This method generates a learning rate curve that appears exceptionally smooth, with the curve starting at a higher learning rate level and gradually approaching zero. Such a learning rate scheduling mechanism not only promotes rapid convergence of the model but also helps to enhance the model's generalization ability.

$$\eta_{max} = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_i}\right)\pi\right) \tag{2}$$

When $T_{cur} = T_i$; set$\eta_t = \eta_{min}$. When $T_{cur} = 0$ set $\eta_t = \eta_{max}$ initialize the current temperature T_{cur} to a predetermined value of T, at which point the algorithm starts and randomly adjusts the temperature to reach its lowest value. As the algorithm progresses, the temperature T_{cur} begins to decrease gradually following the pattern of a cosine function, ensuring that the learning rate can fluctuate within a reasonable range.

Restart Strategy: When T_{cur} drops to zero, it indicates that the algorithm will perform a restart. In this restart process, the paper set η to its maximum value η_{max} to ensure that the model can have different learning rates at various training stages, thereby avoiding getting stuck in local minimum value in this new cycle. Although the adjustment of the learning rate still follows the pattern of a cosine function, due to the resetting of η to η_{max} at the restart and this effectively provides the model with a "fresh" starting point, allowing the model to undertake further learning tasks. As the cycles continue to repeat, the model's understanding of the data gradually deepens, and its generalization ability steadily improves. If a model performs exceptionally well within a certain cycle, then the length of that cycle can be appropriately extended to allow the model to utilize the current learning rate more fully for training; conversely, if the model becomes stuck or performs poorly, then the cycle can be restarted prematurely, and adjustments can be made to the rate or magnitude of the learning rate decrease, in hopes of achieving better training results in the new cycle. This strategy not only helps the model converge to the optimal solution more quickly but also enhances the model's generalization ability, allowing it to demonstrate better stability and robustness when facing unknown data.

2.3. Hungarian Matching

In this collection, each curve is labeled with a binary tag indicating whether it represents a lane or not. The lane prediction is represented as $\mathcal{H} = \left\{h_i | h_i = \left(c_i, g_i\right)\right\}_{i=1}^{N}$, where N is set to a relatively large value that exceeds the number of lanes in the existing samples. Each element h_i represents a possible lane or non-lane. Meanwhile, the paper constructed a real-world sequence of lane markings that encompasses all observed lanes in a specific scenario, represented as$\mathcal{L} = \left\{\hat{l}_i | \hat{l}_i = \left(\hat{c}_i, \hat{s}_i\right)\right\}_{i=1}^{N}$. Since the size of the predicted set may be larger than the actual number of lanes, the paper expands the real lane markings to the same length, filling the unoccupied positions with non-lane labels. Next, the paper defines an optimization problem aimed at finding an optimal mapping function z, which establishes a one-to-one correspondence between the set of predicted curves and the augmented set of real lane markings. By minimizing the cost of this mapping, the paper can ensure that each real lane achieves the best possible match with its predicted counterpart.z(i) represents the most suitable measurement curve for the actual vehicle i.

$$\hat{z} = \arg_z \min \sum_{i=1}^{N} d\left(\hat{l}_i, h_{z(i)}\right) \tag{3}$$

This process transforms the lane detection problem into a bipartite graph matching problem. By solving this minimization problem, the paper is able to precisely match the predicted curves with the actual lanes, thereby significantly enhancing the accuracy of lane detection [9]. Hungarian loss [10] for all pairs matched in the previous step. Referring to the method of Nicolas Carion [10] and others,this study define the loss similarly to the losses of common object detectors:

$$L = \sum_{i=1}^{N} -\omega_1 \log_{P_{\hat{z}(i)}}\left(\hat{c}_i\right) + \left(\hat{c}_i = 1\right)\omega_2 L_1\left(\hat{s}_i, s_{\hat{z}(i)}\right) + \left(\hat{c}_i = 1\right)\omega_3 L_1\left(\widehat{\alpha}_i, \alpha_{\hat{z}(i)}, \hat{\beta}_i, \beta_{\hat{z}(i)}\right) \tag{4}$$

2.4. Lane Shape Model

This formula is used to represent a single-lane line on approximately flat ground:

$$X = kZ_3 + mZ_2 + nZ + b \tag{5}$$

Where k, m, n, and b are real number parameters, and (X, Z) represents a point on the horizontal plane.

For an oblique camera with an angle φ to the ground, the curve of the inclined image plane is

$$u' = \frac{k' \times \cos^2 \varphi}{\left(v' - f \sin \varphi\right)^2} + \frac{m' \cos \varphi}{\left(v' - f \sin \varphi\right)} + n' + \frac{b' \times v'}{\cos \varphi} - b' \times f \tan \varphi \tag{6}$$

Here, f is the focal length measured in pixels, and $\left(u', v'\right)$are the corresponding high-frequency transformed positions [9]. By combining the parameters with the pitch angle φ, the form of the curve in the tilted camera plane is given by:

$$u' = \frac{k''}{\left(v' - f''\right)^2} + \frac{m''}{\left(v' - f''\right)} + n' + b'' \times v' - b''' \tag{7}$$

In almost all scenarios, the lanes are of a thin and elongated shape. Therefore, the k, n, m, and f parameters are shared across all scenarios [11]. Thus, the parameters output by the t-th lane are:

$$g_t = \left(k'', f'', m'', n', b_t'', b_t''', \alpha_t, \beta_t\right) \tag{8}$$

3. Experiments Results

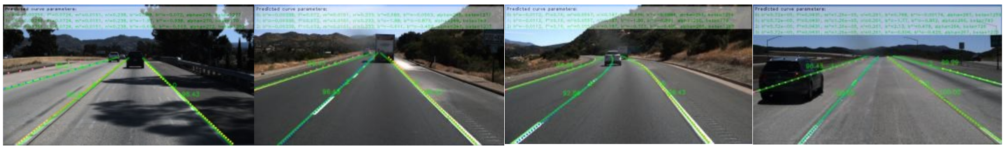

Figure 2. Visualization of predictions on the Tusimple dataset.

As shown in Figure 2, the visualization results of this study on the Tusimple data set. At the coding level, the application of gtns demonstrates its evident advantages: these images are exactly the results our model has derived after a series of complex calculations. In more complex environmental conditions, the model is able to accurately calculate various parameters of the curve, such as curvature, rate of change, second-order curvature, third-order curvature, etc.

Table 1. Comparison with current advanced methods on the Tusimple data set.

Method	fps	*FP*	FN	*Acc*
SCNN	7	.0617	.0180	96.53
PINet	30	.0294	.0263	96.70
FastDraw	90	.0760	.0450	95.20
This study	420	.0293	.0402	96.21

As shown in Table 1, the results obtained by comparing this study with currently popular methods. At the coding level, the application of gtns demonstrates its evident advantages: these images are exactly the results our model has derived after a series of complex calculations. In more complex environmental conditions, the model is able to accurately calculate various parameters of the curve, such as curvature, rate of change, second-order curvature, third-order curvature, etc.

In the table, the paper uses the Tusimple metric to evaluate the accuracy of various methods. Data shows that this method performs the best in terms of speed, while still maintaining a high level of accuracy. Despite some minor compromises in error rate, this does not affect its effectiveness in practical applications.

Compared to the FastDraw method, our method has improved accuracy by about 1%. More importantly, compared to other two-stage methods, this approach achieves competitive accuracy with fewer parameters and faster speed, as well as the lowest false detection and missed detection rates.

4. Conclusion

The lane detection algorithm proposed in this study, based on GTNs, achieves efficient and accurate detection of lane lines by integrating the strengths of graph neural networks and the Transformer architecture. The experimental results demonstrate that the algorithm can stably output key parameters of the lane shape model, such as curvature and rate of change, under various weather conditions, providing robust technical support for the lane departure warning and adaptive cruise control functions of autonomous vehicle systems. Although algorithms perform well in lane detection tasks, their accuracy can be compromised under extreme weather conditions or when roads are severely worn, requiring further optimization to enhance robustness in complex conditions. Looking forward, it is possible to explore the integration of lane detection algorithms with other autonomous driving functionalities more closely.

References

[1] Zhang, Y. (2023). Research on Lane Detection Algorithm based on Convolutional Nural Network. Beijing Jiaotong University.

[2] Borkar, A., Hayes, M., Smith, M. T. (2011). A novel lane detection system with efficient ground truth generation. IEEE Transactions on Intelligent Transportation Systems, 13(1): 365-374.

[3] Ke, H. (2021). Research on Lane Line Detection and Tracking under Complex Road Conditions. Frontiers in Science and Engineering, 1(3): 162-165.

[4] Lucas, T., Rodrigo, B., Thiago, M., Paixao C. B., Alberto, F., De, S., and Thiago, O. S. (2010). Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. IEEE, Conference on Computer Vision and Pattern Recognition (CVPR).

[5] He, K. M., Zhang, X. Y., Ren, S. Q., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Kipf, T. N., and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR).

[7] Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P., and Ye, Y. (2019). Heterogeneous graph attention network. CoRR, abs/1903.07293.

[8] Seongjun, Y., Minbyul, J., Raehyun, K., Jaewoo, K., and Hyunwoo, J. K. (2019). Graph Transformer Networks. Neural Information Processing Systems (NeurIPS).

[9] Liu, R. J., Yuan, Z. J., Liu, T., and Xiong, Z. L. (2020). End-to-end Lane Shape Prediction with Transformers. Computer Vision and Pattern Recognition (cs.CV).

[10] Nicolas, C., Francisco, M., Gabriel, S., Nicolas, U., Alexander, K., and Sergey, Z. (2020). End-to-End Object Detection with Transformers. Computer Vision and Pattern Recognition (cs.CV).

[11] Ilya, L., Frank, H. (2016). SGDR: STOCHASTIC GRADIENT DESCENT WITH WARM RESTARTS. ICLR 2017 conference paper.