Multi-Lane Detection Using CNNs and A Novel Region-grow Algorithm

Yi Sun¹, Jian Li¹ and Zhen Ping Sun¹
¹National University of Defense Technology, Changsha, China
1171272375@qq.com, lijian316@163.com, 13974913933@139.com

Abstract. In this paper, we propose a novel approach to detect lanes robustly in different scenarios. Theoretically, HT (Hough Transform) can extract straight lines efficiently from images, but it cannot distinguish whether the straight lines are belong to lane-markings. To solve this problem, we integrate a simple convolutional neural networks modified from Lenet and geometry constraints to distinguish the types of lines. And in the following part, a region-grow algorithm is adopted in this paper to fit lanes, which is realized by growing local ROI (Region-of-Interest) gradually. We use RANSAC to extract the main direction in local ROI and guide the growth of the ROI. At the same time, in order to ensure right growth of ROI when the lane is dashed, we use a DVF (Direction Vector Field, modified from GVF [19]), which has large capture range and is generated by the edge-direction map. Tests of the proposed approach under kinds of conditions are discussed.

1. Introduction
Lane detection algorithms play an important role in ADAS (Advanced Driver Assistance System), not only it is significant for lane-keeping tasks, but it can tell the traffic rules represented by lane markings on structured roads. Research in this domain mainly focus on how to extract the correct lane features (such as edge feature, line feature) and line fitting, which are the key problem to ensure the robustness of the algorithm.

We use HT to extract line segments in this paper, which is used in many works. However, efficient post-process is necessary for eliminating the noisy line segments in its result, because a few line segments extracted by HT are not belong to traffic lane. Geometry constraints for post-process were used in [12, 17, 18], such as vanishing-point-based constraints, road width constraints and so on. Unfortunately, this kind of constraint usually failed to deal with some line segments on rails, appearance features shall be used to improve the performance in post-process. We have designed a simple CNNs (Convolutional Neural Networks) modified from Lenet [20] to extract feature of the result of HT and purify it. Patches used by CNNs are proposed by line segments. Sometimes, although the global scene changes, the scenarios in proposal-region actually are changeless, so that the generalization risk is limited.

To make our algorithm fitting the lane more accurate at distance, a region-grow algorithm is adopted in our work which is used to choose reasonable edge points for fitting, and is realized by growing local ROI gradually from bottom to top in image. We extract the main direction, which is provide by RANSAC and DVF, in local ROI and guide the growth of the next ROI. We can obtain a series of ROIs growing along the road direction through this method, then we can extract edge points for fitting from these ROIs.

Related Works are introduced in the second part of this paper, part three describes the proposed CNNs,
and a novel Region-grow algorithm which is constrained by RANSAC and DVF is introduced in the fourth part.

2. Related Works
The extraction of lane features, and lane fitting are important problems in a lane detection system. Several approaches were proposed in the past few years to solve these issues.

2.1 Extracting Lane Feature
Edge operators such as Canny are usually used to extract the edge features [4, 12, 18]. In order to improve the sign-to-noise ratio in edge-map, Yoo H et al. [4] strengthen the lane-markings feature by training a LDA model, Gaikwad V et al. [12] propose a brightness stretching function PLSF which makes lane-marking more clearly. Pollard E et al. [14] combine two different extraction algorithm and make use of different local threshold, which makes the extraction algorithm perform well. Geometry constraints have been used to filter false lines in [12, 17, 18]. Machine learning methods are also used to improve the detection performance, Kim [21] uses classifier to judge whether a patch includes the lane-marking, these patches are provided by small sliding windows. Zhao K et al. [7] use the magnitude and direction of gradients to validate lines.

Convolutional neural networks often achieves very good results when extracting features. In [2], a semantic segmentation networks is trained by using the ground truth generated by high-precision map after an on-line calibration. In [9], CNN and RNN are combined for extracting image structures and lane-markings, Lee et al. [15] introduce a multi-task learning networks based on predicting vanishing point location, which can extract lane-markings under complex conditions.

2.2 Curve-Fitting Methods
Zhou S et al. [1] and Ozgunalp U et al. [6] use the road tendency information provided by the vanishing point to estimate an optimal parameters of the curve model they used. Wang Y et al. [14] divide the image into several parts, and extract control points from each part, finally fitting these points by B-snake. Kim [21] uses RANSAC algorithm to fitting curve. In [16], this problem is transformed into an optimal connection problem between super-pixels, and they use CRF (Conditional Random Field) to select the optimal connection. Neven D et al. [11] turn lane detection task into an instance segmentation task, which solve both feature-extraction and curve-fitting at the same time.

3. Line extraction with HT and classification with CNNs
In this paper, we use the edge-direction of edge points to narrow the voting range of edge points when using HT according to equation (1). Actually, it is proved to be good for reducing noise in most cases. In equation (1), \((x, y)\) is the location of edge point \(P(x, y)\) in image, and \(\theta\) is the edge direction of \(P(x, y)\), \(\nu\) represents the half width of voting range, we set \((\rho, \theta)\) as the Hough space.

\[
\rho = x \cdot \cos \theta + y \cdot \sin \theta \quad \theta \in [\phi - \nu, \phi + \nu]
\]

(1)

It is worth mentioning that we calibrate the edge direction of \(P(x, y)\) according to equation (2), and get a new edge-direction map \(D(x, y)\). In equation (2), it shows that the edge-direction of \(P(x, y)\) is same as the slope \(k\) of the line segments it belong to. \(D(x, y)\) will be used to calculate DVF in part 4.

\[
y = k \cdot x + b
\]

\(D(x, y) = k\)

(2)

Convolutional neural networks is a good feature extractor, which can free us from designing handicraft features and can often extract better features under supervision. To eliminate false lines extracted by HT, we propose a CNNs which is modified from Lenet to classify patches provided by line segments. Table 1 shows the structure of our CNNs.
Table 1. Structure of CNNs.

| Index | 1  | 2  | 3  | 4  | 5  | 6  |
|-------|----|----|----|----|----|----|
| Layer | data | conv+relu | pooling | conv+relu | interp | conv |
| Index | 7  | 8  | 9  | 10 | 11 | 12 |
| Layer | pooling | conv | pooling | Inner-Product | Inner-Product | Prob |

We choose two endpoints of straight line as diagonal ends to get the patches described in figure 2. These patches will be used as the input of our networks. We have compared handicraft features composed of HOG and RGB feature with feature extracted by CNNs. We extract two different features from 5000 patches, and use PCA to reduce feature dimensions for visualization, which is displayed in figure 1. We also test both two different ways on our test dataset in different scenarios, the result shows that CNNs do better than handicraft features combined with SVM, and it also have a good generalization performance.

![Figure 1. Two different Features (positive samples are drawn in red, negative samples are drawn in cyan): (a) handicraft feature composed of HOG and RGB feature; (b) Feature extracted by CNNs](image)

![Figure 2. Samples: (a) positive samples, (b) negative samples.](image)

Table 2. Accuracy of two features on classification mission in different scenarios. HOG and RGB features are used by SVM.

|          | Easy condition | Poor visibility | Overexposure | Night | Background | Occlusion |
|----------|----------------|-----------------|--------------|-------|------------|-----------|
| SVM      | 0.822          | 0.634           | 0.687        | 0.737 | 0.651      | 0.764     |
| CNNs     | 0.948          | 0.952           | 0.953        | 0.986 | 0.965      | 0.981     |

4. Curve fitting based on RANSAC and Direction Vector Field

We propose a region-grow algorithm based on RANSAC and Direction Vector Field, which are used to guide the growth of ROI. This kind of ROI will grow from the top of each line segment extracted by Hough Transform. RANSAC algorithm is used to extract local line segment \( \left(l : y = k \cdot x + b\right) \) in ROI, this line segment \( l \) provide a guidance for the growth of next ROI. We define \( P1(x_1, y_1), P2(x_2, y_2), P3(x_3, y_3), \) and \( P4(x_4, y_4) \) are four endpoints of the ROI, we can get new ROI by updating these endpoints according to equation (3), where \( w \) is the width of ROI, \( \Delta y \) is the height, and \( (\hat{x}_i, \hat{y}_i) \) is the last location of \( P1 \). Figure 3 shows this process.

![Figure 3.](image)
\[ \{P1, P3\} \in \{(x, y) \mid y = k \cdot x + b - w\} \]
\[ \{P2, P4\} \in \{(x, y) \mid y = k \cdot x + b + w\} \]
\[
x_3 = x_4 = \hat{x}_i \\
x_i = x_2 = x_i + \Delta s
\] (3)

Figure 3. Region-grow constrained by RANSAC.

However, sometimes there might not exist edge points such as some parts in a dashed lane, to solve this problem, we create a directional vector field (DVF) \( V \) to replace RANSAC algorithm, which have a larger influence range than edge-point. This DVF are modified from GVF [20], it will provide general trend guidance for the growth of next ROI.

\[
V = [u, v] \quad (4)
\]

The initial value of \( V \) is provided by \( D(x, y) \). We can get the final DVF by minimize \( E \) in equation (5). We can obtain the Euler equation of equation (5), and then obtain the following iterative formulas (6), where \( t \) represents the iterative times:

\[
\begin{align*}
E &= \iint \left( u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) dx dy \\
\left. u_{t+1} \right| &= \nabla^2 u + u_t \\
\left. v_{t+1} \right| &= \nabla^2 v + v_t
\end{align*}
\] (5)

(6)

\( V \) will guide the updating of ROI when there is no enough edge points for RANSAC, a new line \( l: y = k \cdot x + b \) is extracted from ROI by using \( V \) according to equation (7), where \( N \) is the number of pixels in ROI. Then \( P1(x_1, y_1), P2(x_2, y_2), P3(x_3, y_3) \) and \( P4(x_4, y_4) \) can be calculated by equation (3).

\[
k = \frac{1}{N} \sum_{i=1}^{N} \frac{u_i}{v_i} \\
b = -k \cdot \hat{x}_i + \hat{y}_i + w
\] (7)

Finally, we can extract sets of edge points \( \{(x_i, y_i) \mid i = 1, 2, ..., n\} \) from ROIs gotten by region-grow algorithm, equation (8) shows the curve model used in this paper and finally we use least square method to fitting this points described by equation (9).

\[
y = a \cdot x + b + \frac{c}{x - vx} \quad (8)
\]
5. Experiment

We test our algorithm in different scenarios, using a dataset provided by Xi’an Jiaotong University [22]. Classification result is showed by figure 4, where the noisy lines are marked with green, and the true lanes are marked with red color. We have chosen 36 different scenarios for test, among which 14 roads are easy and 22 roads are challenging such as occluded, broken road.

Table 3 shows the TPR/FPR of our algorithm in easy and challenging road, it also compare the different performance when we use classification networks only and when we incorporate the CNNs and geometry constraints mentioned by [12, 17, 18], obviously, accuracy of our classification algorithm is improved. The TPR and FPR are calculated by equation (10).

\[
TPR = \frac{TP}{TP + FN} \quad (10)
\]

\[
FPR = \frac{FP}{TN + FP}
\]

Figure 4. Performance of our classification networks under different conditions

|                | No Geometry Constraints(Easy Road) | Geometry Constraints(Easy Road) | No Geometry Constraints(Challenging Road) | Geometry Constraints(Challenging Road) |
|----------------|-----------------------------------|---------------------------------|-------------------------------------------|----------------------------------------|
| TPR            | 0.961                             | 0.984                           | 0.891                                      | 0.924                                   |
| FPR            | 0.018                             | 0.007                           | 0.049                                      | 0.006                                   |

Figure 5 shows the DVF generated by method described in Section 4, which use quiver to represent the directional vector flow V. This kind of field provide a larger guidance range for region-grow especially when there is no edge points. And figure 6 shows the results of region-grow only using RANSAC when the lane is dashed and has a large curve, apparently, our region-grow algorithm failed under this kind of condition. Figure 7 shows the results integrating RANSAC and DVF, we can see the performance of region-grow is improved obviously. We show the final result of the whole algorithm proposed in this paper in figure 8.
6. Conclusions
In this paper, we propose using CNNs and Geometry Constraints to choose right lines from the result of Hough Transforms. The CNNs proposed in this paper is modified from Lenet, which takes little computing resources. By combining the Geometry Constraints and classification, we could eliminate noisy lines effectively. In order to fitting curve line better at a distance, we use region-growing algorithm to get edge-points used for fitting, this region-growing algorithm is guided by local direction, which is provided by RANSAC and Vector-Field. Then, we can use least square method to fitting these points, finally we get the curve model of the lane. Method introduced in this paper performs well in kinds of scenarios, but it still has some drawbacks, for example, the Vector-field is easily disturbed by noisy edges, then the wrong force-field will misleading the direction of the region-grow, and the FNR of the line classification will be supposed to reduce.

Acknowledgement
This work was supported by NSFC Grants 61473303

References
[1] Zhou S, Jiang Y, Xi J, et al 2010 A novel lane detection based on geometrical model and Gabor filter Intelligent Vehicles Symposium, pp 59-64.
[2] Behrendt K and Witt J 2017 Deep learning lane marker segmentation from automatically generated labels International Conference on Intelligent Robots and Systems. IEEE, pp 777-782.
[3] Kim J, Kim J, Jang G J, et al 2017 Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection *Neural Networks*, 87 109-121.

[4] Yoo H, Yang U and Sohn K 2013 Gradient-Enhancing Conversion for Illumination- Robust Lane Detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(3) 1083-1094.

[5] Sivaraman S, Trivedi M and M 2013 Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach[J]. *IEEE Transactions on Intelligent Transportation Systems*, 14(2) 906-917.

[6] Ozgunalp U, Fan R, Ai X, and Dahnoun N 2017. Multiple lane detection algorithm based on novel dense vanishing point estimation. *IEEE Transactions on Intelligent Transportation Systems*, 18(3) 621-632.

[7] Zhao K, Meuter M, Nunn C, et al 2012 A novel multi-lane detection and tracking system *Intelligent Vehicles Symposium. IEEE*, pp 1084-89.

[8] Ju H Y, Lee S W, Park S K, et al 2017 A Robust Lane Detection Method Based on Vanishing Point Estimation Using the Relevance of Line Segments[J]. *IEEE Transactions on Intelligent Transportation Systems* PP(99) 1-13.

[9] Li J, Mei X, Prokhorov D, et al 2017 Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene[J]. *IEEE Transactions on Neural Networks & Learning Systems*, 28(3) 690-703.

[10] Long J, Shelhamer E and Darrell T 2014 Fully Convolutional Networks for Semantic Segmentation[J]. *Transactions on Pattern Analysis & Machine Intelligence*, 39(4) 640-651.

[11] Neven D, De Brabandere B, Georgoulis S, et al 2018 Towards End-to-End Lane Detection: an Instance Segmentation Approach[J]. *preprint arXiv/1802.05591*.

[12] Gaikwad V and Lokhande S 2015 Lane Departure Identification for Advanced Driver Assistance[J]. *Transactions on Intelligent Transportation Systems* 16(2) 910-918.

[13] Wang Y, Teoh E K and Shen D 2004 Lane detection and tracking using B-Snake[J]. *Image & Vision Computing*, 22(4) 269-280.

[14] Pollard E, Gruyer D, Tarel J P, et al 2011 Lane marking extraction with combination strategy and comparative evaluation on synthetic and camera images[C] *International Conference on Intelligent Transportation Systems*. pp 1741-46.

[15] Lee S, Kim J, Yoon J S, et al 2017 Vpgnet: Vanishing point guided network for lane and road marking detection and recognition[C] *International Conference on Computer Vision*, pp 1965-73.

[16] Hur J, Kang S N and Seo S W 2013 Multi-lane detection in urban driving environments using conditional random fields[C] *Intelligent Vehicles Symposium* pp 1297-02.

[17] Lee C and Moon J H 2018 Robust Lane Detection and Tracking for Real-Time Applications[J]. *IEEE Transactions on Intelligent Transportation Systems*, PP(99) 1-6.

[18] Niu J, Lu J, Xu M, et al 2016 Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting[J]. *Pattern Recognition*, 59(C) 225-233.

[19] Xu C and Prince J L 1997 Gradient vector flow: A new external force for snakes *Computer Vision and Pattern Recognition*, pp 66-71.

[20] Lecun Y L, Bottou L, Bengio Y, et al 1998 Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11) 2278-2324.

[21] Kim Z W. 2008 Robust lane detection and tracking in challenging scenarios[J]. *IEEE Transactions on Intelligent Transportation Systems*, 9(1) 16-26.

[22] http://trafficdata.xjtu.edu.cn/index.do