2016 IEEE Intelligent Vehicles Symposium (IV)
Gothenburg, Sweden, June 19-22, 2016

Accurate and Robust Lane Detection based on Dual-View Convolutional Neutral Network

Bei He[1], Rui Ai[1], Yang Yan[1] and Xianpeng Lang[1]

Abstract— **In this paper, we propose a Dual-View Convolutional Neutral Network (DVCNN) framework for lane detection. First, to improve the low precision ratios of literature works, a novel DVCNN strategy is designed where the front-view image and the top-view one are optimized simultaneously. In the front-view image, we exclude false detections including moving vehicles, barriers and curbs, while in the top-view image non-club-shaped structures are removed such as ground arrows and words. Second, we present a weighted hat-like filter which not only recalls potential lane line candidates, but also alleviates the disturbance of the gradual textures and reduces most false detections. Third, different from other methods, a global optimization function is designed where the lane line probabilities, lengths, widths, orientations and the amount are all taken into account. After the optimization, the optimal combination composed of true lane lines can be explored. Experiments demonstrate that our algorithm is more accurate and robust than the state-of-the-art.**

I. Introduction

The lane detection technique, which is applied in either generating the Highly Automated Driving Map (HAD Map) data or locating the vehicle during driving, has attracted increasing interest recently. It is widely explored in multiple vehicle safety systems, such as Lane Departure Warning System (LDWS), Advanced Driver Assistance System (ADAS) and so on [1]. To detect the lane lines accurately and robustly, researchers have tried various of sensors, including LIDAR, cameras and radar. Since visual data is the closest modality to human and cameras are the cheapest media for imaging, the camera-based sensor takes a leading role in lane detection techniques. In literature, lane detection methods are divided into 3 parts [2]: pre-processing for normalizing scenes and excluding other disturbed objects, feature extraction to reinforce lane lines and model fitting for the estimations of lane line parameters.

Early methods [3], [4], [5] filtered the front-view or top-view image with classical edge operators, followed by hough, spline or b-snake fitting on the binarized regions. Considering there existed many confusing objects and false detections, [6], [7], [8] introduced tracking strategies to utilize the temporal connection. Different from them, Aly [9] designed a classic coarse-to-fine framework. First in the top-view image, lines and splines were fitted with the RANdom SAmple Censensus (RANSAC) method. Then Aly [9] re-mapped lane lines to the front-view image and refined them by searching maximal responses in limited neighborhoods. The above methods were based on manual rules, which usually failed in complex and noisy scenes, like urban villages. On one hand, [10], [11] introduced the machine learning methods, e.g., boosting and Support Vector Machine (SVM), to recognize the patterns of lane lines. Unfortunately, features of lane lines were deficient and not easy to represent. On the other hand, [12], [13] aimed at image cleaning and enhancement [2] such as the exposure correction and shadow removal, which is lack of generalization and robustness due to those case-by-case strategies. Recently the Convolutional Neutral Network (CNN) framework [14] showed shocking influence on computer vision applications, known for the discriminability and stability. [15], [16] applied the CNN framework, where regions traversed in the front-view image were trained and would be predicted directly. In the following, easy line or spline fitting was presented for accurate locations. As the sizes of the data layer and the labelling one were comparable, the trained model was large and easily over-fitted. Consequently, Kim [15] enlarged the size of the pooling layer, while Huval [16] reduced the size of the labelling layer. However, there existed 2 disadvantages. On the one hand, the above pixel-based methods neglected thin lane lines, especially along the curbs, which were fatal for the map-data generation. On the other hand, objects similar to lane lines in the limited neighborhood were confusing so that many false detections were reserved.

Focusing on those problems, this paper proposes an accurate and robust lane detection algorithm. Our contributions are threefold. First, we design a weighted hat-like filter to initialize lane line candidates. It not only recalls all potential lane lines, but also excludes most false detections via measuring the neighboring textures. Second, the front-view and top-view images are taken as input to our Dual-View Convolutional Neutral Network (DVCNN) framework. From the front-view image, disturbance of moving vehicles, barriers and curbs are excluded; while from the top-view image, we reject non-club-shaped structures, e.g., ground arrows and words. Third, this paper presents a novel global optimization strategy by taking into account lane line probabilities, length, widths, orientations and the amount. Via traversing all valid combinations quickly, accurate and robust lane lines are detected. Visual and quantitative experiments demonstrate the high-accuracy and robustness of our algorithm.

II. The proposed algorithm

A. The Framework

An overview of our algorithm is illustrated in Fig.1, which is divided into 3 steps. 1) The top-view image is inverse-perspective-mapped from the front-view one, where

[1]The authors are with the group of Baidu Map, China. {hebei01, airui01,yanyang01,langxianpeng}@baidu.com

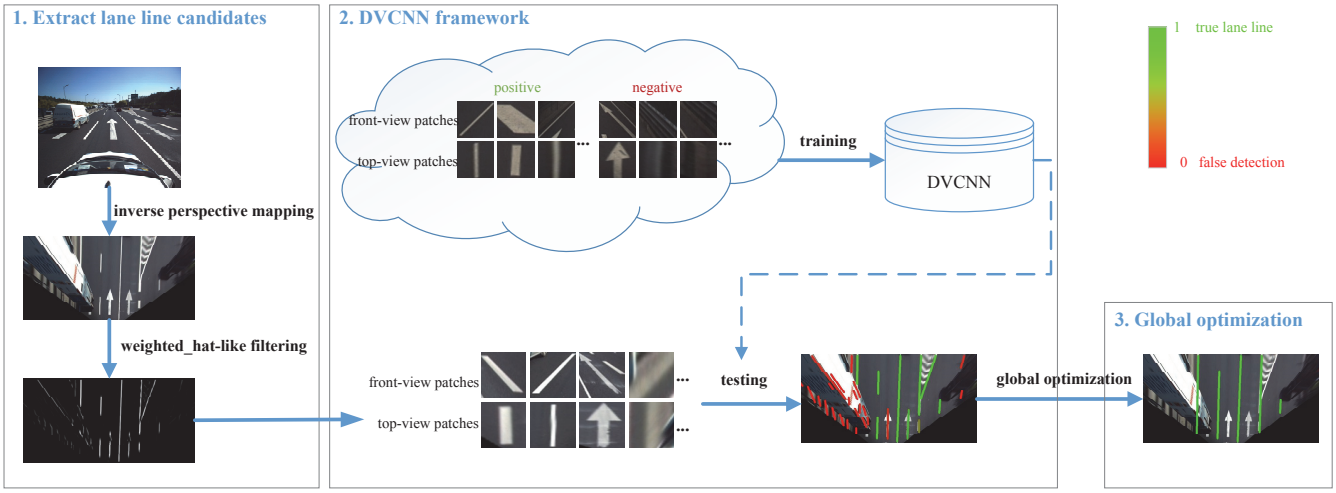


Fig. 1. Overview of the proposed algorithm. 3 steps are listed, including the lane line candidates extraction, the DVCNN framework and the global optimization. Detected lane lines along with the corresponding probabilities are drawn. Probabilities arise while the color changes from red to green. The following figures in this paper are similar.

lane line candidates are extracted with our weighted hat-like filter. 2) The front-view and top-view patches are utilized as input to the DVCNN framework simultaneously. The outputs of our framework correspond to the probabilities of the true lane lines. 3) All achieved results, including lane line probabilities, lengths, widths, orientations and the amount are combined for our global optimization, where the final outputs are refined. In Fig.1, 5 lane lines marked by the green color (green means the high probability) are detected accurately and robustly.

The differences between [15], [16] and ours can be listed as follows. First, we extract the patterns of patches instead of the ones of pixels. As features of lane lines are deficient, pixel-based model easily induces false detections. Second, the front-view and top-view patches are utilized in the meantime. Consequently, we can exclude lots of confusing objects from different views, such as barriers, ground arrows and so on. Third, we regard the lane lines on the ground as a whole, where not only probabilities, but also lengths, widths, orientations and the amount are all considered. Thus, our global optimization reinforces the true lane lines and extracts them precisely.

B. Extract Lane Line Candidates

As we all know, testing all patches extracted from the image is impractical and unbearable. Therefore, firstly we propose to extract lane line candidates via filtering the top-view image. In the top-view image, lane lines are related to actual physical properties of 3D world, as the widths are consistent from near to far. Further, orientations of lane lines are all vertical approximately, though they may be affected a bit by the angle between the road and the driving directions. Based on the above conditions, lots of false detections are excluded. It can be referred to [9] that how to map the front-view image to the top-view one.

Different from related methods [9], [15], this paper proposes a weighted hat-like filter for extraction. Denote I_f, I_g as the filtered and grayed images, we define the filter as,

$$I_f(x,y) = \delta(x,y) \cdot \{2 \cdot Block_{middle} - Block_{left} - Block_{right}\}, \quad (1)$$

where $Block_{middle} = Block(x - w/2, y - h/2, w, h)$, $Block_{left} = Block(x-3w/2, y-3h/2, w, h)$, $Block_{right} = Block(x + w/2, y + h/2, w, h)$ and

$$Block(x,y,w,h) = \sum_{(r,c)\in[x,x+w]\times[y,y+h]} I_g(r,c). \quad (2)$$

Here w and h denote the width and the height of that block. Since the lane lines are club-shaped, the hat-like property can reinforce the response of lane line regions. However, gradual textures are also strengthened [9], [15]. That would corrupt the following intensity normalization and bring in many false detection regions. To overcome the problem, we add an adaptive weight $\delta(x,y)$ for the pixel (x,y) in Eq.(1),

$$\delta(x,y) = \begin{cases} 0, & if\ Block_{middle} < Block_{left}\ or\ Block_{right} \\ 1, & otherwise \end{cases}. \quad (3)$$

For each pixel, if $Block_{middle}$ is not larger than $Block_{left}$ or $Block_{right}$, it will be restrained. Consequently, the influence of gradient textures is alleviated significantly by the assigned weight. As shown in Fig.2, the pixel, marked by the green circle, belongs to the gradient textures. With traditional hat-like filter [9], [15], the noisy peak value is reserved, while our weighted one can suppress that significantly. From the figure, it also can be drawn out that our filter reduces lots of false detections.

From the filtered image, we can achieve the binarized one via intensity normalizing and thresholding. Considering the recall should be guaranteed first and the authenticities of lane lines can be verified by the following work, we reserve all regions with "white" pixels as lane line candidates. Considering the vertical property of lane lines in the top-view image, we remove lines when the angles between them

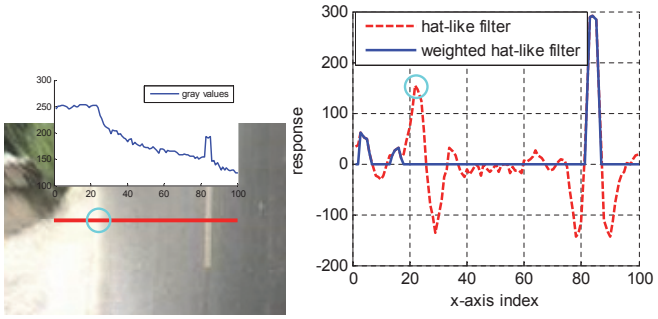


Fig. 2. The gray values along the red line are shown in (a). The blue circle corresponds to a false detection. In (b), response values of the traditional hat-like filter [9], [15] and ours are plotted.

and the vertical axis are larger than $45°$. This operation can avoid the effects of lane changing and curves. In practical, generally $30 \sim 50$ candidates are extracted from an image with 2448×2048 pixels, where the computational cost is acceptable. Finally, we apply the RANSAC method on each candidate and initialize the parameters of lane lines, such as the slopes, intercepts and crossed pixels.

C. DVCNN

Since lane line candidates are extracted from the top-view image as mentioned above, the corresponding top-view patches can be cropped easily. To get the front-view ones, we re-map the lane lines from top-view to front-view and then collect the bounding rectangles. The two patches are utilized as the input simultaneously based on 3 considerations. First, club-shaped structures, including the lane lines, can be easily extracted from the top-view image. Different lengths are normalized so that we can recognize the lane lines in the "global" space. Second, from the front-view image, it is convenient to utilize other information such as the context, area and so on. That means the lane lines are recognized in the "local" space. Third, since those two patches are extracted from different sources, the features to be learnt should not be confused. Consequently, the two patches much be sent independently, but not merged into a multi-channel one.

The architecture of our DVCNN framework is plotted in Fig.3. The siamese network joins two sub-network at the head, to estimate the probability of the true lane line. The data layer accepts the front-view and top-view patches as inputs, where sizes are normalized into 128×128 and 64×64 pixels respectively. The 2 sub-networks are similar, both composed of several layers of convolution, ReLU [14], pooling and ending with a fully-connected one. Considering the pattern of the front-view patch is more complex than the one of the top-view patch, we fix 5 layers for the former and 4 ones for the latter. The sub-networks apply the 5×5 convolutional kernels, since larger sizes will neglect the thin pattern of the lane lines while smaller ones are not suitable for the convergence. On the top, the sub-networks output vectors with both the sizes 256×1, which are concatenated as one vector. Finally the labelling layer is fully connected

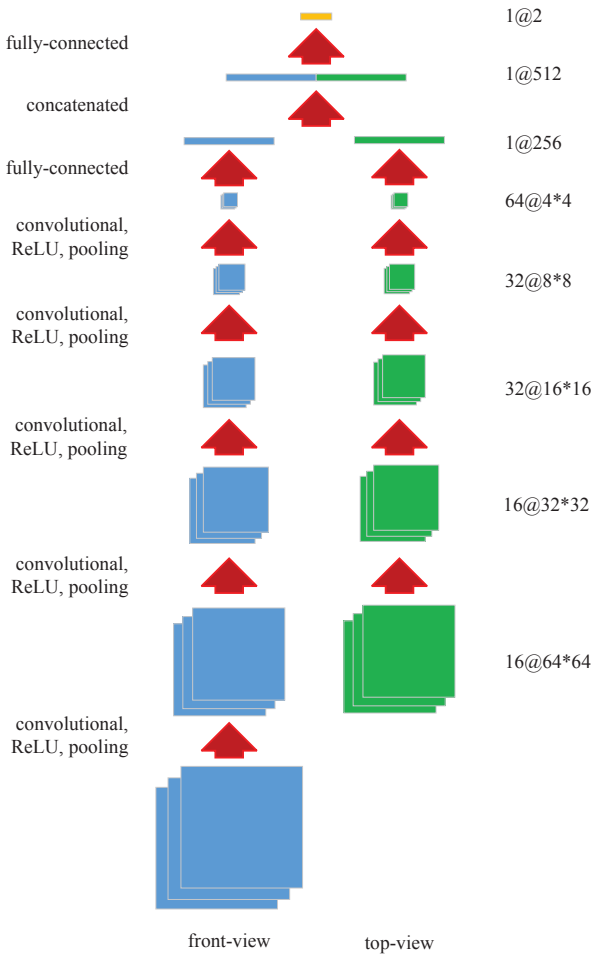


Fig. 3. The architecture of our DVCNN framework.

with the above vector, which can predict the authenticities of the test lane line. In our network, there exists the dominate number of model parameters in the first fully-connected layer, about $64 \times 4 \times 4 \times 256 \approx 262,000$.

D. Global Optimization

With the DVCNN framework, the probability p_l of each lane line candidate l belonging to the truth is provided. Traditional methods [16], [15] directly set an empirical threshold and only preserved high-probability ones as the truths. However, we find it inappropriate to neglect other messages. The road is composed of multiple lanes, where the width, lengths, orientations and the amount are constrained simultaneously. Therefore, the lane detection task is equivalent to searching an optimal combinations of the lane lines and achieving the best consistency in the real world. Messages needed to be considered are listed as follows.

1) As the number of lanes is unknown, the optimum amount of the lane lines is hard to be determined. If we sum up all probabilities of available lane lines for global optimization, there exists a dilemma that the more lane lines the better. Consequently, we utilize the amount of the lane lines to suppress the optimization energy.

2) Generally lane lines extend from near to far in the front-view image, and from bottom to top in the top-view one. Compared with false detections, true ones are always longer which can be utilized as another measurement. Furthermore, we should prevent that the length dominates the global optimization since the type of lane lines are various.
3) There exist several differences between the true lane lines and false detections. First, if a big gap in orientations exists between two neighboring lane lines, more than one of them should be recognized as the false detections. Second, the lane lines with conflicting positions and orientations should be separated. Third, if the width of the lane composed of two lane lines is narrower than the specified threshold, we believe that there exist contradictions.

Based on the above considerations, our global optimization function is defined as,

$$\arg\max_{L_i \subseteq L} \; w_{amo} \cdot \{\sum_{l \in L_i} \{p_l + w^l_{len} + \sum_{m \in L_i, m \neq l} w^{ml}_{link}\}\}. \quad (4)$$

l refers to the lane line belonging to L_i, which is the i^{th} combination selected from L. w_{amo}, w_{len} and w_{link} denote the weights of the amount, length and linking respectively,

$$w_{amo} = \exp\{-n^2/\sigma^2\}, \quad (5)$$

$$w^l_{len} = 1/\{1 + \exp\{-s^l + H/2\}\}, \quad (6)$$

$$w^{ml}_{link} = \begin{cases} -\infty, \; if \; r_{min} < \|r_l - r_m\| < r_{max} \\ -\infty, \; if \; \|\rho_l - \rho_m\| > \rho \\ 0, \; otherwise \end{cases}. \quad (7)$$

Here n and σ correspond to the amount of the lane lines and the threshold. s^l and H denotes the length of the lane line l and the height of the top-view image. r^l and ρ^l are the radium and the angle represented in the parameter space, where r_{min}, r_{max}, ρ constrain the values. Particularly, we prefer the sigmoid function for the length weight in Eq.(6), which not only reinforces longer lane lines, but also avoids that the length weight dominates the global optimization.

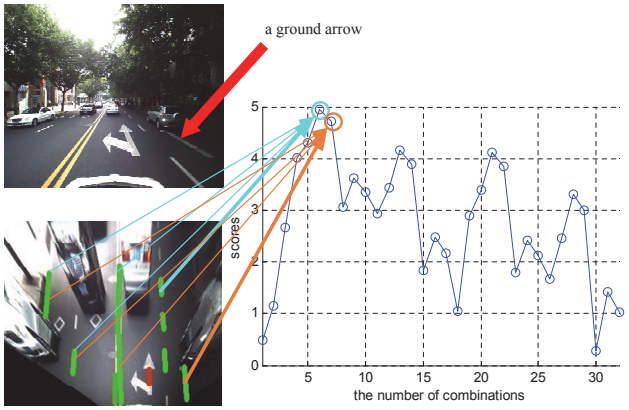
Due to our weighted hat-like filter for extracting lane line candidates, the total number is limited. Hence, searching in all combinations is affordable. First, we utilize the linking weight in Eq.(7) to prune lots of impossible combinations. Second, all residual combinations are traversed for calculating the outputs according to Eq.(4). Third, the combination with the maximal output of the energy function is reserved as the one composed of true lane lines.

III. Experiments and Results

To verify the performance of our algorithm, we implemented it in C++ and ran it on the Intel Xeon CPU with 1.8GHz Dual. The main parameters were fixed as follows: $w = 5$, $h = 11$, $\sigma = 10$, $r_{min} = 20$, $r_{max} = 40$ and $\rho = 15°$. The sizes of the front-view image and the top-view one are 2448×2048 and 1000×300 respectively. Our dataset is composed of 47 batch images, where $20,000$ images are included in each batch. From that, we choose 8 batches

(a) (b) (c)

Fig. 4. Results of (a) the top-view only CNN model, (b) the front-view only CNN one and (c) our DVCNN one are shown. Those blue ellipses in (a) and (b) illustrate several false detections.



(a) (b)

Fig. 5. Extracted lane line candidates are plotted in (a). In (b), the first and second highest scores among all combinations are marked by the blue and orange circles. The only difference between them is choosing the dash lane line or the ground arrow.

with total $10,000$ images randomly for the experiments. The test scene covers the expressway, streets, avenues and country roads in order to guarantee the completeness of our experiments. Those images are manually annotated, where true lane lines are marked accurately.

A. Module Evaluations

In Fig.4, results of the top-view only CNN model, the front-view only CNN one and our DVCNN are listed. As marked by the blue ellipses in Fig.4(a) and 4(b), there exist multiple false detections. The top-view only CNN model does not utilize messages such as the context and area, where the club-shaped structures are all reserved. Moreover, the front-view only CNN model suffers from local disturbances like ground arrows. Compared with them, our DVCNN model achieves accurate and robust results, as demonstrated in Fig.4(c).

Fig.5 provides an example of our global optimization. The blue and orange circles in Fig.5(b) represent to the first and second highest scores among all combinations. The only difference is the choice of the dash lane line or the ground arrow. Due to the occlusion of the moving vehicle, the ground arrow is similar to the lane line, as marked by the red arrow. In traditional methods [16], [15], both would be reserved where the false detection is induced, where our algorithm utilizes the linking weight to detect the confliction. Furthermore, the dash lane line defeats the ground arrow due to our length weight.

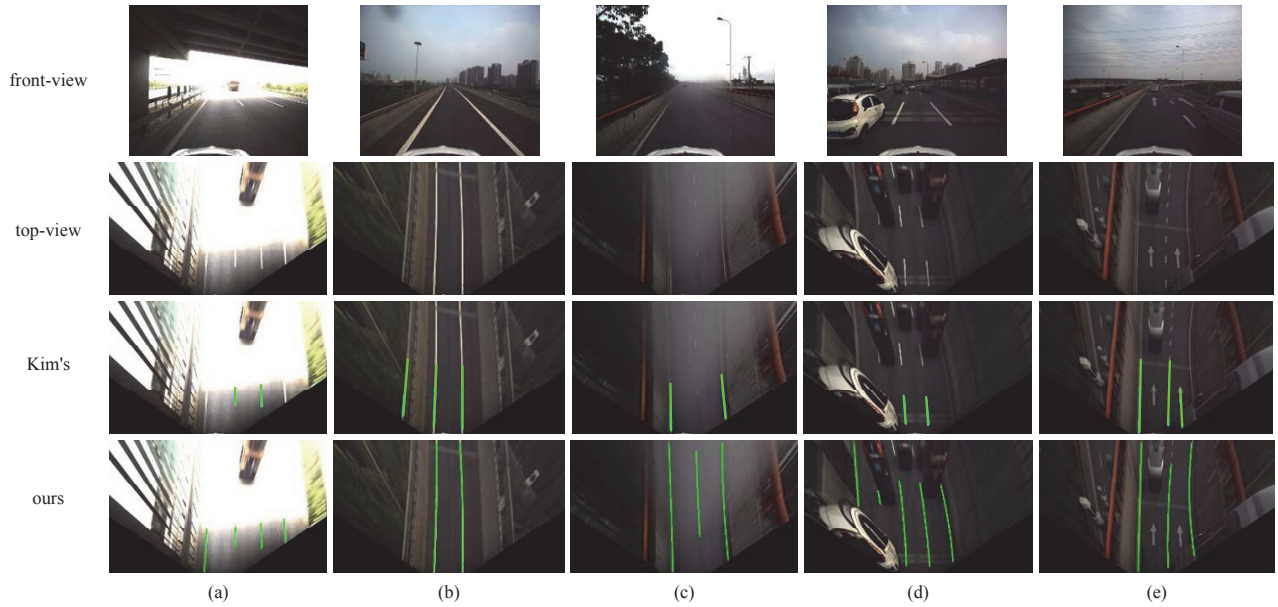


Fig. 6. Visual comparisons of Kim's [15] and ours. (a)-(e) correspond to scenes of illumination variation, the disturbance of barriers, the fog, the occlusion and the confusing ground arrows. For each scene, the front-view image, the top-view one, Kim's [15] result and ours are listed from top to bottom respectively.

B. Overall Evaluations

The overall evaluations are compared with Kim [15], which also applied the CNN model for detection. We arrange the experiments both from the visual and quantitative comparisons. In the visual experiments, we list 5 representative images and corresponding results. While in the quantitative comparison, the classical recall (detected ground truths / ground truths) and precision (detected ground truths / detected all) ratios for different batches are tabulated.

Visual results of Kim [15] and ours are drawn in Fig.6(a)-(e). Under illumination variance in Fig.6(a), our algorithm outperforms Kim's [15] since 2 more true lane lines along the curbs are explored. That is contributed to the proposed weighted hat-like filter. In Fig.6(b), the barrier is mistaken as the lane line by Kim [15], whereas our DVCNN framework eliminates the false detection in the front-view image. The scene is fogged in Fig.6(c); however, our algorithm recalls all true lane lines robustly, especially the one in the middle. Fig.6(d) illustrated that the disadvantage of Kim's [15] relying only on the front-view image. As the near parts of the true lane lines are occluded, those ones are neglected by Kim [15]. Different from Kim [15], we extract the lane line candidates from the top-view image where true ones are all detected. In Fig.6(e), due to our global optimization, ground arrows are removed from the optimal combination.

The quantitative results, including the recall and precision ratios, are tabulated in Table.I. On the one hand, the recall ratios of our algorithm are much higher than Kim's [15]. This is attributed to our weighted hat-like filter, which extracts all potential lane line candidates. Furthermore, the proposed DVCNN framework and global optimization verify the authenticities of the lane lines accurately and robustly. On the other hand, our precision ratios also outperform Kim's [15]. Though more lane line candidates are recalled, our DVCNN strategy and the global optimization module can cooperate to remove false detections. In batch 2 and 6, there exist multiple thin barriers which is exactly similar to lane lines, the precision ratios of ours are affected slightly. On average, our recall and precision ratios are 92.80% and 95.49% respectively.

TABLE I
QUANTITATIVE COMPARISONS OF KIM'S [15] AND OURS.

Batch Index	Recall		Precision	
	Kim's [15]	Ours	Kim's [15]	Ours
1	56.92%	**90.77%**	88.40%	**96.27%**
2	74.00%	**95.27%**	**97.81%**	97.58%
3	59.91%	**91.04%**	92.03%	**92.12%**
4	64.57%	**94.22%**	93.12%	**95.66%**
5	66.09%	**95.16%**	93.68%	**97.71%**
6	70.98%	**93.93%**	**95.39%**	93.19%
7	38.43%	**91.44%**	95.91%	**97.48%**
8	48.20%	**90.54%**	93.04%	**93.93%**
Avg.	59.89%	**92.80%**	93.67%	**95.49%**

In experiments, our algorithm costs $11s$ for an image with 2448×2048 pixels. During the whole process, the DVCNN detection module occupies 90% computation time, which can be optimized by several techniques, such as multi-threading, GPU speedup and so on. To sum up, accurate and robust lane lines can be detected quickly, both from visual and quantitative view.

IV. CONCLUSION

This paper presents a robust and accurate lane detection algorithm based on the DVCNN framework. On the one hand, a weighted hat-like filter is designed to extract all potential lane line candidates. Moreover, gradual textures

are suppressed so that most false detections are removed. On the other hand, we propose a novel DVCNN strategy via combining the front-view image and the top-view one. The front-view image helps us exclude moving vehicles, barriers and curbs, while the top-view one only reserves club-shaped structures similar to the lane lines. In addition, we design a global optimization function which takes the lane line probabilities, lengths, widths, orientations and the amount into account. Consequently, the optimal combination composed of true lane lines are explored. Both visual and quantitative experiments illustrate that our algorithm gives rise to more accurate and robust lane detection results than the state-of-the-art.

Unfortunately, our algorithm will fail when the lane lines are occluded by the vehicles totally or the image is completely over-exposing. However, it can be resolved by using multiple different-orientation cameras or combing other sensors like LIDAR. In the future, we will involve more sensors to achieve better precision and stability.

REFERENCES

[1] Hao Li and Fawzi Nashashibi, "Lane detection (part i): Mono-vision based method," 2013.

[2] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[3] Massimo Bertozzi and Alberto Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62–81, 1998.

[4] Yue Wang, Dinggang Shen, and Eam Khwang Teoh, "Lane detection using spline model," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 677–689, 2000.

[5] Yue Wang, Eam Khwang Teoh, and Dinggang Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.

[6] Joel C McCall and Mohan M Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 533–537.

[7] Hsu-Yung Cheng, Bor-Shenn Jeng, Pei-Ting Tseng, and Kuo-Chin Fan, "Lane detection with moving vehicles in the traffic scenes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 4, pp. 571–582, 2006.

[8] Kun Zhao, Mirko Meuter, Christian Nunn, Dirk Muller, Stefan Muller-Schneiders, and Josef Pauli, "A novel multi-lane detection and tracking system," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 1084–1089.

[9] Mohamed Aly, "Real time detection of lane markers in urban streets," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 7–12.

[10] ZuWhan Kim, "Robust lane detection and tracking in challenging scenarios," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 16–26, 2008.

[11] Raghuraman Gopalan, Tsai Hong, Mike Shneier, and Rama Chellappa, "Video-based lane detection using boosting principles," *Snowbird Learning*, 2009.

[12] Jongin Son, Hunjae Yoo, Sanghoon Kim, and Kwanghoon Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1816–1824, 2015.

[13] Avishek Parajuli, Mehmet Celenk, H Bryan Riley, et al., "Robust lane detection in shadows and low illumination conditions using local gradient features," *Open Journal of Applied Sciences*, vol. 3, no. 01, pp. 68, 2013.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] Jihun Kim and Minho Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *Neural Information Processing*. Springer, 2014, pp. 454–461.

[16] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Royce Cheng-Yue, Fernando Mujica, Adam Coates, et al., "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.