Light Invariant Lane Detection Method Using Advanced Clustering Techniques

Aleksandr Karavaev
ITMO University
Saint-Petersburg, Russia
alexkaravaev@protonmail.com

Rami Al-Naim
ITMO University
Saint-Petersburg, Russia
rami.naim2010@yandex.ru

***Abstract*—In this paper we propose a novel approach to detecting road lanes from video stream in bad-light road scenarios. The main focus of this article is given to the introduction new image binarization method in non-common color space followed by improved density hierarchical clustering algorithm called HDBSCAN. These techniques allow to detect lane boundaries even in low-light scenarios with robust and parameter-free setup.**

***Keywords*—Lane detection, Computer vision, HDBSCAN, DBSCAN, Image Thresholding**

I. Introduction

The aim of this paper is to design low-cost in terms of processing time pipeline for robust road marking detection in low-level of light road scenarios.

Self-driving cars are actively introduced into people lives. Their number and complexity of software of on-board computers are increasing [1]. Cars of only one Waymo company managed to drive twenty million miles in self-driving mode[1]. Nowadays most of companies are relying on solutions with deep neural networks for perception module of the car [2]. Moreover most of them are using radars and lidars, which allow to perceive the environment even in dim and dark road scenarios unlike to usual camera [3]. Despite the fact that these approaches are dominant in the field, they have major limitations that restrict them from full implementation in the industry and interfere with scalability to more users [4].

There is a need for training deep neural networks on a powerful machine equipped with a lot of video cards for achievement of good precision and recall of an output detections. Furthermore, even after successful training and deploying such big model developers need to install high performance computers on the car because real-time execution is essential in the case of self-driving car. This solution is more difficult to scale, not to mention the trend to small-size components and reducing their cost. Interest in single-board computers is gradually rising because their best qualities — compactness and price [5].

A lot of disputes in the community of self-driving cars are ongoing right now. Some researchers consider advantages and disadvantages of using cameras or lidars [6]. The experience of Tesla company shows, that only-camera solution is possible[2]. Main disadvantage of lidar is its price, that is comparable in some cases to the price of the car itself. And most of the solutions require a lot more than one lidar on the car (up to 6 small and big lidars, that are mounted on various sides of the car). Some researches and forecasters insist that the price of lidars will eventually fall down[3]. However, other ones compare camera with human visual cortex, which can reliably identify and detect distance to various objects[4].

And this is our motivation why we focus on developing system, that detects road lane markings with the cameras. We believe that such a solution should meet the following requirements:

- System should be cheap for maintaining and producing for the purpose of high scalability.
- System should be robust to rapidly changing light environment.
- System should be fast for not powerful computers and shouldn't require a lot of computational power.

Main proposals of our paper:

- Image processing in color space CIE L*a*b, that is decreasing light impact on the scene and image.
- New formula for image binarization.
- Using HDBSCAN — more recent method of density clustering instead of DBSCAN.
- Using color information in clustering.

Combined together, these methods provide a robust pipeline with few parameters that need to be configured. Thus, the time for configuring is reduced, which seems very useful. By the term pipeline here and after we mean a set of data processing elements connected in series, where the output of one element is the input of the next one[5].

[1] K. Wiggers, "Waymo's autonomous cars have driven 20 million miles on public roads", 6 Jan 2020, https://venturebeat.com/2020/01/06/waymos-autonomous-cars-have-driven-20-million-miles-on-public-roads/

[2] Tesla official website, autopilot description, https://www.tesla.com/autopilot

[3] A. Davies, "This Lidar Is So Cheap It Could Make Self-Driving a Reality", 7 Nov 2019, https://www.wired.com/story/lidar-cheap-make-self-driving-reality/

[4] B. Templeton, "Elon Musk's War On LIDAR: Who Is Right And Why Do They Think That?", https://www.forbes.com/sites/bradtempleton/2019/05/06/elon-musks-war-on-lidar-who-is-right-and-why-do-they-think-that/#5b6971232a3b

[5] Pipeline (Computing), https://en.wikipedia.org/wiki/Pipeline_(computing)

The paper is organized as follows. Section 2 is giving brief review of other papers in the field. Section 3 describes proposed approach in details. Section 4 compares proposed approach with other ones. Finally, Section 5 gives the conclusion.

II. Related Work

A. Neural Network approach

In recent years neural networks have become a common tool in image processing tasks. In the field of driving an autonomous car, neural networks are often used to detect road markings, obstacles and road signs. The article [7] describes a method for detecting road marking lines based on neural networks. The authors provide experimental data indicating a high accuracy of detection of road marking lines and a high image processing speed. However, the experiments were conducted on equipment, the cost of which is approximately equal to $2,000, and such a price may not be acceptable. The article [8] presents experimental data for several algorithms for detecting lanes using neural networks. All algorithms presented in this article use either expensive or specialized equipment for image processing. This can lead to a significant increase in cost, or narrow the scope of the possible application of the system.

B. Binarization and Processing in a Different Color Spaces

Materials used for carriageway marking lines often have bright colours which differ a lot from the rest the road's surface. Consequently, some algorithms for road marking line detection utilize this quality and use prior knowledge of colours for thresholding [9], [10]. For binarization based on beforehand estimated colours' thresholds the HSV color space is widely used. It shows a good results, however, with different illumination of a scene the colors detected by the camera are different [11]. Thus, with changes in the illumination the chromatic values of the pixels on the image may also differ, which leads to a significant amount of noise on the binarized image (Fig. 1). As a result, the further applied algorithms for lane detection, such as Hough transformation [12], may not give an expected result and fail in finding the lanes. Such behavior can be handled by setting the algorithm's parameters, but this often decreases robustness and reduces number of scenarios in which the algorithm can be used.

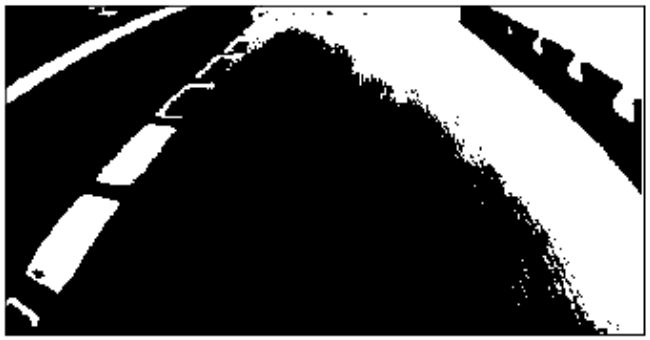


Fig. 1. Example of a poor binarized mask: color of road lines in HSV color space changed due to changes in illumination

C. Using Clustering For Denoising

Some road lane detection methods use density clustering for finding lane on the road image. For example, authors in paper [13] use similar approach for clustering points belonging to lane marking. However, they use pretty old and outdated clustering algorithm DBSCAN. Besides that, authors use different threshold operation for the initial steps of processing the image and Otsu binarization [14], which can fail or give bad result in some cases of road scenarios.

Aside from this paper, clustering in the lane marking scenario is used in the work [15]. In this paper authors use simple hierarchical clustering and new method of post-processing clustering results. For every calculated cluster they calculate the slope line of it, after that they count the intersections of these slopes with other clusters. The clusters with the most number of intersections are used later. This improvement deals with filtration of clusters that are elongated more on the y axis instead of x axis and utilized the geometrical feature of almost every road marking, which are located mostly one on the top of another on one line. This improvement work well on straight road scenarios, nonetheless it can fail on turns or curb road environment.

III. Road Marking Detection

In this section we examine our algorithm in details. The full block diagram of the approach can be found on Fig. 2

Data: Input video stream
Result: Clustered image points
initialization;
while *not end of video* **do**
  Read current color frame;
  Convert image into CIE Lab color space;
  Normalize L-channel using MIN-MAX method;
  Calculate mean and standard deviation of the L-image;
  Calculate threshold value;
  Binarize L-channel;
  Image $<-$ Select pixels from initial color frames, where thresholded image != 0;
  Initalize HDBSCAN with selected parametrs;
  Downscale image;
  Cluster Labels, Labels Probabilites $<-$ Clusterize image with HDBSCAN algorithm;
  Discard labels with low probability(< 0.75);
  Mask labels to image;
  Resize image with labels to initial size;
end

Algorithm 1: Full proposed algorithm

A. Binarization in CIELab Color Space

It is necessary to define two assumptions in order to choose a threshold value of binary mask with road marking from an image.

Assumption 1: Road marking lines takes about 5% of an image.

Assumption 2: Road marking lines brighter than majority of objects on a road image (road surface, roadside, cars, etc.).

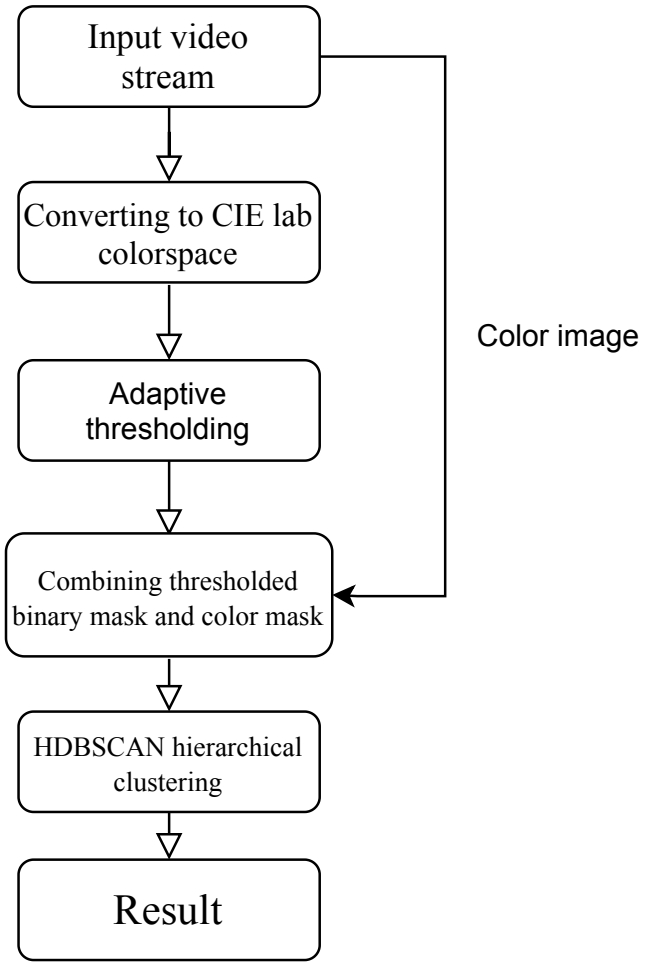


Fig. 2. Total diagram of our approach

The image with road segment is captured by camera in RGB color space. In order to reduce noise on the image we preprocess the image by applying Gaussian blur [16]. In experiments the kernel with size 15 was chosen to filter high-frequency noises on the image.

The next step is converting the image from RGB to desired color space. We propose to use one of the perceptually uniform color space — CIELab [17]. In this color space each pixel is encoded with three values: L, a and b. L describes the brightness of a pixel, in other words, characteristic of luminance. a and b values describe chromatic characteristics, from green to red and from blue to yellow, respectively.

Further, we propose to calculate histogram of the L channel of the image in CIELab color space and normalize it. Considering aforementioned assumptions, pixels of road marking lines are located in the upper right part of the distribution. For the histogram equalization the max-RGB like method is used [18]. This approach utilize the fact that humans perceive color relative to the contrast of the full image, i.e. difference between brightest and darkest point of the image. As a result, the histogram is normalized in such a way that its low boundary is a minimum value of a non-zero value of a brightness of the image in L channel, and upper boundary is maximum value of a brightness of the image in L channel.

Pixels of road marking lines on the image are located at the rightmost side of the normalized histogram (5% of the whole distribution). In order to choose appropriate threshold value the property of a Gaussian normal distribution and 3σ rule is used [19]. We calculate the mean μ and standard deviation σ for the normalized histogram of the image and use Gaussian distribution to estimate the threshold value. This value bounds 5% of the distribution of the brightest pixels which represent road marking lines. Example of approximation of a normalized histogram can be seen at Fig. 3, where red line represents normal distribution. The following equation is used for threshold value estimation:

$$t = \mu + \sigma(k + \frac{\sigma}{2\sigma_u}), \tag{1}$$

where t — threshold value; μ — mean value from normalized histogram; σ — standard deviation from normalized histogram; σ_u — standard deviation of an uniform distribution; k — scaling coefficient.

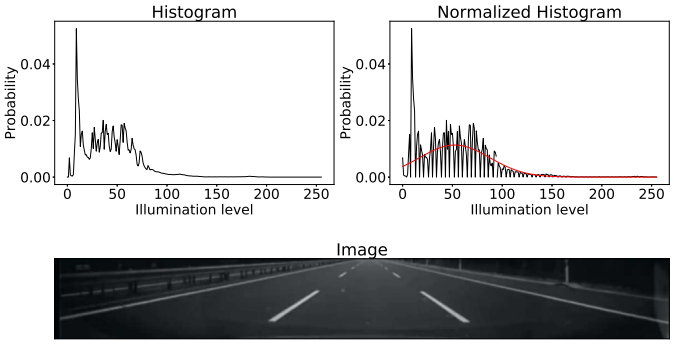


Fig. 3. Example of histogram normalization and its approximation with Gaussian distribution

From (1) it is clear that the threshold value lies in the half-interval $(2\sigma; 3\sigma]$ and the exact value depends on standard deviation. The standard deviation of the uniform distribution is used to normalize the standard deviation of the histogram and to accurately determine where the threshold value lies between the mentioned interval. The scaling parameter k is chosen depending on how much area of an image is covered by road marking lines. For real road application we propose to use $k = 2$. To illustrate the interval in which the threshold value lies the image from Duckietown Project is used (Fig. 4) [20]. Since road marking lines on this image cover a larger area compared to real road images, scaling coefficient is set to 1. Group of pixels on the right side of normalized histogram represent road marking lines. They are inside of the threshold marked by a rectangle on the plot.

If a scene of image is well-illuminated, its contrast is high. Thus, values on the histogram that correspond to the pixels of the road will be located near each other at the left side of the distribution, while values with intensities of a road marking lines will be located at the most right part of the histogram. In that case standard distribution will be rather small and the ratio of σ/σ_u consequently also will be small. Because of this threshold value will be close to 2σ, which guarantees that it will be less than values of the pixels of bright road marking lines. Contrariwise, when the scene of image is dark, histogram might not have distinct groups of pixels of the road or road

marking lines. The standard deviation of such distribution will be greater than in the case described before, so the ratio of $\frac{\sigma}{\sigma_u}$ will be relatively large. As a result, calculated threshold value will be close to 3σ, so it might be grater than values of the road marking lines, but it reduces amount of fake pixels marked as road marking lines (pixels of background or road itself).

The results of image binarization using described algorithm can be seen in Fig. 5. As a method for comparison Otsu binarization was chosen because it is one of the most popular method of threshold value calculation. Our approach for threshold value calculation shows better results: binary mask contains less noise and registers even yellow lines of road marking. This is possible because of those two assumptions at the beginning of this subsection.

B. Hierarchical Clustering

After the initial processing stage (binarization) it is necessary to obtain information from the black-and-white binary image about which pixels belong to road marking and which are just road or other miscellaneous noise or even error from binarization algorithm used before. The proposed algorithm is as follows.

Given that we have the black and white binary mask and the original color image, we combine them into one resulting image. Pixels that were previously white receive color from the original image, and pixels that were plain black remain black.

This procedure is aimed to improve the next clustering part. This increases the useful information for the algorithm, because we will have not only information about the intensity of the pixels, but also the color, and therefore, it is easier to divide the points into meaningful clusters. For example, we can divide into separate groups the road lane and asphalt, which surrounds lane on all sides but has a completely different RGB color.

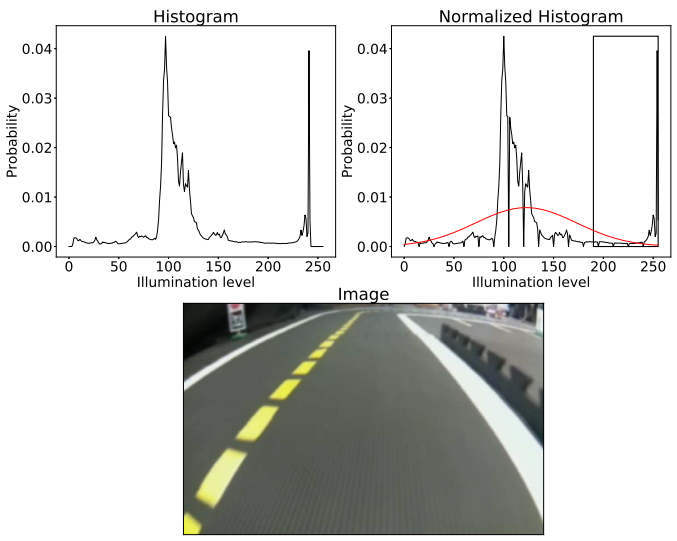


Fig. 4. Example of histogram normalization and its approximation with Gaussian distribution with rectangle representing calculated threshold

In this paper we selected HDBSCAN [21] as the main clustering algorithm which is declared as an improved hierarchical version of more older algorithm DBSCAN [22]. Main improvements over DBSCAN are as follows:

- The algorithm has much fewer parameters that need to be configured, because during data processing HDBSCAN selects the best parameters according to its own indicators, for example, the epsilon parameter.
- The algorithm can find clusters with densities varying within the cluster area.
- The clustering procedure not only assigns a cluster number to each input point, but also calculates vector of probabilities where each probability reflects how probable a point belongs to each cluster or noise.

Now it is worth to emphasize why the main advantages of the clustering algorithm are critically important in lane finding scenario. Even on one route a car might encounter different light conditions, from completely dim (e.g tunnel) to sunny and bright, therefore, there is a need to develop a robust pipeline with several hyper-parameters to configure.

The quality of road markings in reality might be poor and the markings might be partially erased, therefore, it is necessary to be able to detect road markings with holes inside and at the same time to detect them as single cluster.

In the task of autonomous driving the safety comes first, so we select only the points with the lowest probability of noise. The result of this step of the pipeline is showed at Fig. 6.

IV. Results

A. Comparison with Other Approaches

The Fig. 7 represents visual part of comparison of the clustering algorithms. The parameters for clustering algorithms were chosen as follows:

- For K-Means:
  - $n\_clusters = 6$,
  - $random\_state = 0$,
- For DBSCAN:
  - $eps = 10$,
  - $min\_samples = 200$,
- For HDBSCAN:
  - $min\_cluster\_size = 500$,
  - $min\_samples = 200$.

Interpretation of the visual result is as follows. The K-Means algorithm completely misses the actual clusters of the road and segments them only by y-value (Fig. 8b).

The result of DBSCAN is actually not so poor, since it clustered almost all the lanes into separate groups, with the exception of two lanes, which are located extremely right and left (Fig. 8c).

HDBSCAN showed better results than other algorithms, finding all the lanes on the image, although he found only part of the lane on the right (Fig. 8d). It is clear that further

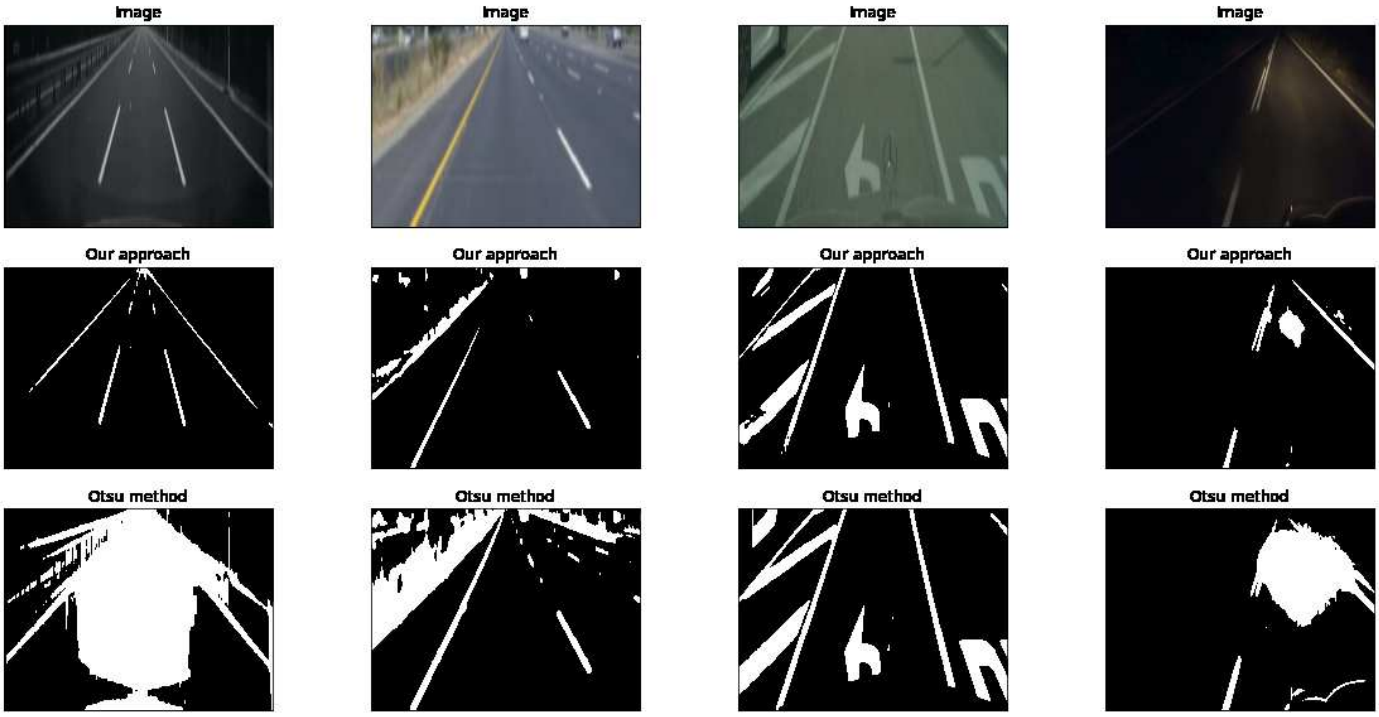


Fig. 5. Results of the proposed algorithm in comparison with Otsu binarization

(a) Initial cropped road image

(b) Road image after binarization

(c) Road image with combined color and binarization

(d) Final result of Clustering

Fig. 6. Pipeline of clustering module

we can fit this points with the spline or parabola and get the actual lane.

Implementations of algorithms in the Python language were chosen. For HDBSCAN we used native author's implementations, DBSCAN and K-Means were used from scikit-learn library [23].

The bechmark was conducted on Raspberry Pi 4 model B with 64-bit ARM Cortex A72 CPU @ 1.5GHz and 4 GB of RAM. The procedure of the experiment was the following. The video with road was loaded and every clustering algorithm was consecutively run on each frame. Frame had the shape (120, 1200, 3) — height of 120 px, width of 1200 px and 3 RGB colors. The full results of the comparison can be found in Table II and I.

The results showed that the second worst result in terms of performance was shown by HDBSCAN with 12.5 FPS (frames per second). However, it is worth noting that the implementations of the algorithms are taken from different libraries, and because of this, the comparison is not very fair.

By using downscaled image there is a significant trade-off between time and precision, which is going to be shown in following section.

TABLE I. PERFORMANCE COMPARISON OF ALGORITHMS WITHOUT DOWNCALING

Algorithm	Mean time on 5th image, sec	Mean time on 15th image, sec	Mean time on 25th image, sec
HDBSCAN	0.7578	0.7517	0.7274
DBSCAN	0.1641	0.1400	0.1438
K-Means	0.2810	0.2925	0.2740

TABLE II. PERFORMANCE COMPARISON OF ALGORITHMS. SCALED BY FACTOR 0.3

Algorithm	Mean time on 5th image, sec	Mean time on 15th image, sec	Mean time on 25th image, sec
HDBSCAN	0.0842	0.0783	0.0720
DBSCAN	0.0096	0.0094	0.0088
K-Means	0.1202	0.1208	0.1210

B. *Experiments*

In order to test mesurable accuracy the proper dataset for testing should have be chosen. We have chosen Unsupervised Llamas (The unsupervised labeled lane markers dataset) [24].

TABLE III. EXPERIMENT RESULTS

Scale factor	AUC	Precision	Recall	Threshold
1.0	0.36934	0.49383	0.44874	0.42679
0.3	0.36649	0.48310	0.33036	0.43664

The dataset consists of over 100 000 annotated images, so we took only 536 images from various parts of the dataset in order to test the proposed algorithm.

Overall, we achieve presion of 49% and AUC of 36%. As can be seen from Fig. 8 proposed solution is sometimes segments white car as road(c) and some artifacts are still present(d). Nonetheless, most of the images are good results such as (a) or (b). Decrease in precision and AUC can be explained by two things:

1) Firstly, we must specify ROI(Region Of Interest) on our own and farther parts of the road are removed from ROI and therefore are not being detected, which is okay, because there is no need to detect road markings from 200 meters from a car. But this markings are marked in the dataset and so in testing phase they are marked as non-detected.
2) Secondly, algorithm is not finding lanes that are on the sides, because they are harder to detect and have low-color intensity.

Future work can be focused on eliminating bad results such as (c) or (d) in Fig.8 by designing post-processing filter steps such as curv-fitting and discarding curves, that are inclined more horizontally, than vertically. Especially this step will be effective in eliminating noise that is coming from white car detection, because curve points are grouped in a form of a car, hence grouped more horizontally.

In order to understand how downscaling images beforehand affects the precision two tests were conducted. Test results with original size image and downscaled by factor of 0.3 is presented in III. As we can see, the decline of precision is not linearly proportional to decline of image size and this can be used to process images.

(a) Initial cropped road image

(b) Result of K-Means

(c) Result of DBSCAN

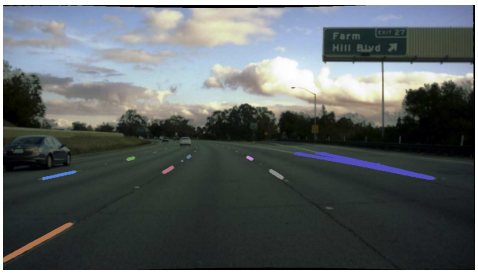
(d) Result of HDBSCAN

Fig. 7. Comparison between most popular clustering methods

V. CONCLUSION

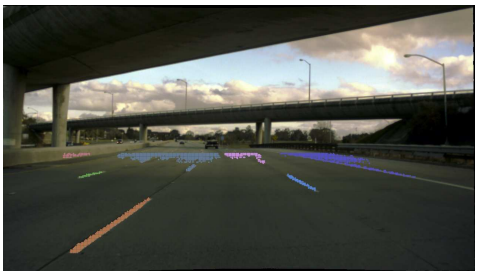
In this article we have proposed a novel approach for detecting lanes on roads that is used for dim and low-light scenarios. Proposed method was successfully tested on the real road images taken from highway. The proposed method of clustering binarized images gave better results than others. As a topic for the future researches we would like to study possibility of using methods of illumination correction and histogram processing for more accurate calculating threshold value.

(a) Good result. Scale 0.3



(b) Good result. Scale 1

(c) Bad result. Car detection. Scale 0.3



(d) Bad result. Road artifacts. Scale 0.3

Fig. 8. Examples of processed images from the dataset

REFERENCES

[1] P. Coppola and F. Silvestri, *1 - Autonomous vehicles and future mobility solutions*, P. Coppola and D. Esztergár-Kiss, Eds. Elsevier, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128176962000019

[2] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[3] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 445–467.

[4] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3266–3273.

[5] H. A. Shiddieqy, F. I. Hariadi, and T. Adiono, "Implementation of deep-learning based image classification on single board computer," in *2017 International Symposium on Electronics and Smart Devices (ISESD)*. IEEE, 2017, pp. 133–137.

[6] P. A. Lazar and V. Shyam, "Agile development of automated driving system: A study on process and technology," Master's thesis, Chalmers University of Technology, 2017.

[7] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.

[8] M. M. Yusuf, T. Karim, and A. F. M. S. Saif, "A robust method for lane detection under adverse weather and illumination conditions using convolutional neural network," in *Proceedings of the International Conference on Computing Advancements*, ser. ICCA 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3377049.3377105

[9] K.-B. Kim and D. H. Song, "Real time road lane detection with ransac and hsv color transformation," *J. Inform. and Commun. Convergence Engineering*, vol. 15, 2017.

[10] J. Kim, S. Kim, S. Lee, T. Lee, and J. Lim, "Lane recognition algorithm using lane shape and color features for vehicle black box," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, Jan 2018, pp. 1–2.

[11] M. S. Drew, J. Wei, and Z.-N. Li, "Illumination–invariant image retrieval and video segmentation," *Pattern Recognition*, vol. 32, no. 8, pp. 1369 – 1388, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003132039800168X

[12] A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *2008 International Conference on Computer and Communication Engineering*, May 2008, pp. 82–88.

[13] J. Wang, W. Hong, and L. Gong, "Lane detection algorithm based on density clustering and ransac," in *2018 Chinese Control And Decision Conference (CCDC)*, June 2018, pp. 919–924.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[15] R. N. Hota, S. Syed, S. Bandyopadhyay, and P. R. Krishna, "A simple and efficient lane detection using clustering and weighted regression," in *Proceedings of the 15th International Conference on Management of Data*, 2009.

[16] S. Eswar, "Noise reduction and image smoothing using gaussian blur." Ph.D. dissertation, California State University, Northridge, 2015.

[17] G. Sharma and R. Bala, Eds., *Digital Color Imaging Handbook*, ser. Electrical Engineering & Applied Signal Processing Series. CRC Press, 2017.

[18] E. H. Land, "The retinex theory of color vision," *Scientific american*, vol. 237, no. 6, pp. 108–129, 1977.

[19] A. Bovik, *The Essential Guide to Image Processing*. Elsevier Science, 2009.

[20] J. Tani, L. Paull, M. T. Zuber, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: An innovative way to teach autonomy," in *Educational Robotics in the Makers Era*, D. Alimisis, M. Moro, and E. Menegatti, Eds. Cham: Springer International Publishing, 2017, pp. 104–121.

[21] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, 03 2017.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] K. Behrendt and R. Soussan, "Unsupervised labeled lane marker dataset generation using maps," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.