

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/336276131>

Scalable ROS-Based Architecture to Merge Multi-source Lane Detection Algorithms

Conference Paper · November 2019

CITATIONS
0

READS
138

3 authors:



Tiago Almeida
University of Aveiro
7 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)



Vitor M F Santos
University of Aveiro
157 PUBLICATIONS 588 CITATIONS

[SEE PROFILE](#)



Bernardo Lourenço
University of Aveiro
5 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PHUA - Project Humanoid at the University of Aveiro [View project](#)



ATLASCAR [View project](#)

Scalable ROS-Based Architecture to Merge Multi-source Lane Detection Algorithms

Tiago Almeida¹, Vitor Santos², and Bernardo Lourenço¹

¹ DEM, University of Aveiro, Portugal

² DEM, IEETA, University of Aveiro, Portugal

Abstract. Road detection is a crucial concern in Autonomous Navigation and Driving Assistance. Despite the multiple existing algorithms to detect the road, the literature does not offer a single effective algorithm for all situations. A global more robust set-up would count on multiple distinct algorithms running in parallel, or even from multiple cameras. Then, all these algorithms' outputs should be merged or combined to produce a more robust and informed detection of the road and lane, so that it works in more situations than each algorithm by itself. This paper proposes a ROS-based architecture to manage and combine multiple sources of lane detection algorithms ranging from the classic lane detectors up to deep-learning-based detectors. The architecture is fully scalable and has proved to be a valuable tool to test and parametrize individual algorithms. The combination of the algorithms' results used in this paper uses a confidence based merging of individual detections, but other alternative fusion or merging techniques can be used.

Keywords: visual perception, data combination, ROS, deep learning, computer vision, road detection, driving assistance

1 Introduction

The fields of Autonomous Driving (AD) and Advanced Driver Assistance Systems (ADAS) have carried out a wide range of studies that can bring new possibilities to the drivers. One of the most relevant is the detection of the road boundaries or lane detection. For that, the car is equipped with relevant sensors to extract information from the real scene.

Different approaches have been implemented over the years to detect those lane and road boundaries, and they can be divided into two main types: methods that use classic computer vision techniques, and more recent approaches that use learning-based techniques, namely using AI (Artificial Intelligence)¹.

Despite this abundance of techniques, there is not a single algorithm that performs accurately in all situations, but it is expectable that a combined result of multiple algorithms can increase the overall performance and robustness when compared to individual algorithms.

¹ <https://emerj.com/ai-sector-overviews/machine-vision-for-self-driving-cars-current-applications>

This work describes a road boundary detection method that includes an ensemble of different types of road detection algorithms, thus providing a more robust detection than any individual algorithm, and being suitable for both unstructured and structured roads. The former are roads with no lane lines on it, commonly presented on rural locations and the latter are normal roads that have clear lane marks and road boundaries. This is a problem since it is necessary to perceive two different types of features. In addition to this, three main problems have to be considered: lighting changes, shadows and vehicle occlusions [15]. Also, this approach can optionally merge the detection from multiple cameras.

2 Related Work

There are several algorithms to detect the road lanes through classical computer vision techniques. The most used pipeline is composed of different phases: image pre-processing, feature extraction and model fitting [1, 2, 9]. Some of these algorithms ally the road detection with the road lane lines tracking [3, 7, 11]. Recently, methods based on Deep Learning (DL) have been developed, providing more robust results, but still at a higher computational cost [5, 8, 10]. The problem presented by many of the algorithms based on classical computer vision techniques is the inability to deal with different types of road scenarios. Algorithms based on modern DL techniques are now appearing but expected to have intensive applications in the future.

The combination of different types of algorithms for road and line detection is not very common in the literature. However, it has been applied successfully in other fields, such as machine learning, where, for example, there is a technique that combines the predictions from multiple trained models to reduce variance and improve prediction performance [13, 16]. The objective is similar to the one presented in this work: to combine different results in order to obtain a more accurate one.

3 Proposed Approach

The proposed approach in this paper consists of a special architecture, which is responsible to combine the detection of multiple algorithms from multiple cameras, in order to produce a more confident detection. The technique can be divided in three parts: the definition and parametrization of detection algorithms, the combination of multiple algorithms from a camera, and finally, the combination of several cameras, each with one or more algorithms.

3.1 Detection Algorithms

One of the challenges of this infrastructure is to work with multiple types of road lane detectors. Therefore, it is crucial to define a common interface for the inputs and outputs of these algorithms. For the inputs, it was observed that

every algorithm expects an image from the camera. Furthermore, two types of outputs were found: either a polygon of the lane, i.e. the lines delimiting the lane, or a binary image of the region of the road lane. Either way, it is important that all algorithms output the same data type; hence, the binary image was chosen because it is the most flexible representation. This image of the road was named the *confidence map*.

To transform the polygons and lane lines into *confidence maps*, an extra step was added to these algorithms to handle the conversion, and unify the road representation. This can be seen in Fig. 1.



Fig. 1: Example of the conversion between polylines to the confidence map. In this example, the lane lines detected by an algorithm (left) are converted to the corresponding *confidence map* (right).

3.2 Combination of multiple algorithms

This part of the procedure receives the multiple confidence maps for the multiple algorithms, and combines them into a single confidence map, that is expected to represent a more confident detection. The steps of this procedure are described next:

1. The computational mechanism tries to synchronise the multiple incoming detections from the various algorithms. This is, however, not always possible, because the latency is different for each algorithm. A threshold for the latency was defined in order to ignore messages that have a rate smaller than the camera frame rate; in the case of this implementation, a threshold for the latency of 0.06 s was used, which corresponds to a frame rate of 15 Hz (due to a hardware limitation). In other words, if a message is older than the threshold, it is discarded. This is possible because the detection algorithms keep the acquisition time stamp of the image.
2. The maps are combined through the logical **AND** operation, to obtain the *intersection zone*.
3. Next, a larger region of confidence is generated after a convolution with a square kernel of size `kernel_size`. This operation smooths the intersection zone of the confidence map, creating a smooth transitions on the boundaries of the detection. An example of this operation being applied is shown in Fig. 2.

4. The maps are also combined through the logical **XOR** operation to obtain the *non-intersection* zone.
5. The region of less confidence is generated by multiplying the pixel values by a constant value L_C given by

$$L_C = \frac{\text{ceil}(kernel_size/2)}{kernel_size} - \alpha, \quad (1)$$

where α is a parameter defined by the user.

6. Finally, the two maps are combined with a **SUM** operation to obtain the final *confidence map*.

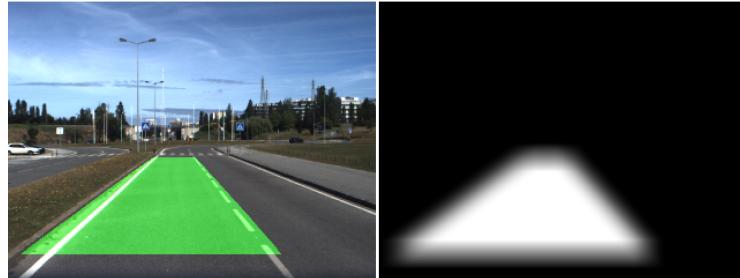


Fig. 2: Example of the effects of the smoothing operation in the confidence map.

The overall procedure is illustrated in Fig. 3.

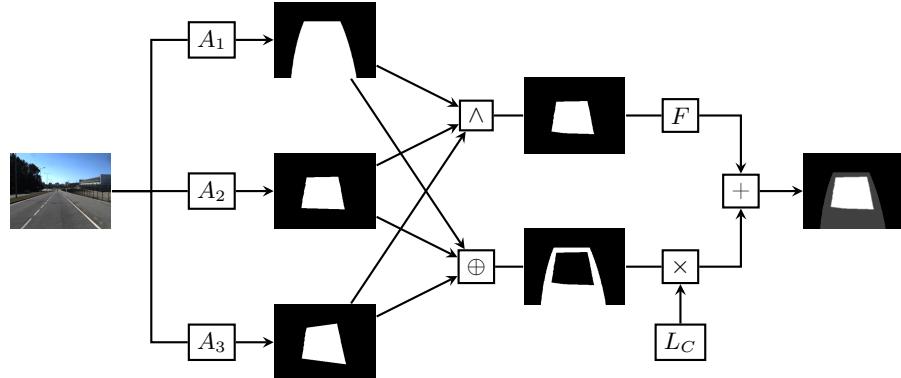


Fig. 3: Example of the technique to combine multiple algorithms. First, the image is processed by the A_1 , A_2 , A_3 algorithms, where each produces a map. The maps are then combined using the AND (\wedge) and the XOR (\oplus), and the corresponding smoothing filter F and the multiplication by L_C are applied, producing the maps with the most and less confidence. Finally, the maps are summed ($+$), producing the final confidence map.

3.3 Combination of multiple cameras

The final step in the proposed architecture is the combination of the detection of the multiple cameras. Note that this step can sometimes be not useful because the camera's fields of view may not intersect. Thus, it can be disabled, then the algorithm produces confidence maps equal to the number of cameras in the system. To merge the confidence maps from multiple cameras, a perspective transformation is necessary to place the cameras into a common reference frame. This can be achieved through the warp transformation using the intrinsic and extrinsic calibrations of each camera.

In the field of autonomous driving, the most common choice for the reference frame is the top view, or *birds-eye view*. This technique is quite commonly named *Inverse-Perspective-Mapping*, or *IPM*. An example of this technique can be seen in [12].

After warping all the confidence maps into this new perspective, the combination is done by averaging the multiple maps. An example of this can be seen in Fig. 4.

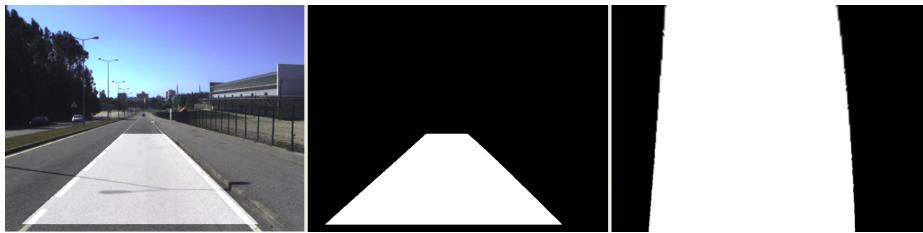


Fig. 4: Example of the warping transformation. The left and center image show the detection map in the camera point of view. The image on the right shows the warped image resulting from the *IPM* technique.

4 Experimental Infrastructure

This work was developed under the ATLASCAR2 project ², which is an Mitsubishi i-MiEV equipped with cameras, LiDAR and other sensors. The sensors used for this work were two PointGrey Flea3 cameras, installed on the roof-top of the car, as can be seen on Fig. 5. The software architecture that handles the connection of the sensors and the algorithms developed is ROS.

Next, the two methods used for this work are described: the classical method, based on standard computer vision algorithms, and a Deep Learning-based approach.

² <http://atlas.web.ua.pt/>



Fig. 5: The AtlasCar2 vehicle. The two cameras used for this work are placed on car roof.

4.1 Detector based on classical methods

The detector that represents the classical method used in this work can be found in a *Udacity* course³. This algorithm is composed of the following steps:

1. Image rectification and warping: this step undistorts the camera image according to the intrinsic parameters. Then, an inverse perspective transform is applied, changing the perspective of the image to a top-down view.
2. Road lanes segmentation: this step is composed of two methods: a segmentation of the red channel of the image and a sobel edge detector. These two partial segmentations are then merged into a final binary segmentation.
3. Curve fitting: this method is applied to the binary segmented image to extract the road lines according to a set of parameters.

An example of the application of this algorithm can be seen in Fig. 6.

4.2 Detector based on deep learning method

Deep Learning (DL) is a thriving field that is revolutionising and constantly achieving state-of-art results in computer vision challenges. It is undeniable that the implementation of these new algorithms could greatly expand the capabilities of the technologies we have today. However, it is important to understand the capabilities and limitations of any algorithm before its implementation in any task, and even more importantly, in the field of autonomous driving. This can

³ <https://medium.com/deepvision/udacity-advance-lane-detection-of-the-road-in-autonomous-driving-5faa44ded487>



Fig. 6: Example of the lane detection using classical computer vision techniques.

be achieved in this work, through a side-by-side comparison of the classical algorithms, described earlier, and the DL approach.

The challenge of road segmentation belongs, in the domain of DL, to the field of pixel-wise semantic segmentation. Or in other words, each pixel in the input image is labelled with the corresponding class (for example, road, car, sidewalk). For our purpose, only the pixels that correspond to the road are important, but there is enough flexibility to interpret more classes, which could be valuable in other ways (for example, traffic sign identification).

The work we developed in this area is mostly the same as the one described in [14], with slight modifications: we used the pretrained resnet50 [6] as the backbone for the UNet network; and we used a modified version of CamVid dataset [4] with just the most important 11 classes, in order to reduce the computational cost inherent to the training of the neural network. The Unet, as most fully convolutional networks, is characterised by an encoder-decoder network. The encoder part produces features that are produced by successive convolutions, where the raw pixel input is transformed into high level features. The decoder network interprets these high level features and associates the corresponding class to the image. Therefore, the encoder successively reduces the spatial dimension of the image, while increasing the depth, and the decoder decreases the spatial size of the images, while decreasing the depth. The encoder is usually transferred from a pre-trained classification network, and the decoder weights are learned through training.

An example of a detection using this algorithm is shown in Fig. 7.

5 Experiments and Results

To demonstrate the usefulness, scalability and reliability of the proposed architecture, several experiments were carried out. They range from the simple “one camera–one algorithm” up to “multiple cameras–multiple algorithms”. The experiments do not assess the architecture directly, but show how the architecture can be used to test the performance of the algorithms and their combination. Therefore, performance indices, related to the confidence maps, were created to



Fig. 7: Example of the DL algorithm successfully detecting the road map. The input image (left) is segmented into a pixel-wise semantic segmentation (center), where, for example, the cars correspond to the orange color and the road in purple. Because only the road is relevant, a map of the road is the result of this algorithm (right).

allow the evaluation and tuning of the algorithms depending on some variable parameter.

The proposed indices are based on the areas of the confidence maps described in section 3.3, and the variable parameter is the size of the smoothing filter used. The first index (I_1) is defined by (2):

$$I_1 = \frac{W_{CA}}{A_T} \quad (2)$$

where W_{CA} is the "weighted confidence area" of the confidence map I , and A_T is the total area of the confidence map, or, formally:

$$W_{CA} = \sum_r \sum_c I(r, c) \quad (3)$$

being I the normalised matrix of the confidence map image, $I(r, c) \in [0; 1]$, and

$$A_T = \sum_r \sum_c \lceil I(r, c) \rceil = \sum_r \sum_c \text{ceil}(I(r, c)). \quad (4)$$

The second performance index (I_2) is given by (5):

$$I_2 = \frac{A_C}{A_T} \quad (5)$$

where A_C is the common area, that is, the area of the confidence map where the confidence is 100%. Formally:

$$A_C = \sum_r \sum_c \lfloor I(r, c) \rfloor = \sum_r \sum_c \text{floor}(I(r, c)) \quad (6)$$

As mentioned, the variable parameter used was the size of the convolution filter (`kernel_size` variable), and sizes from 3×3 up to 51×51 were tested. The results presented next were obtained on the same real road section (Fig. 8) which cover two situations from the several tested: one camera and two algorithms, and two cameras and two algorithms.



Fig. 8: The road section where the experiments were made.

5.1 Results from One Camera and Two Algorithms

The first experiment consists of combining the outputs of two distinct processor ROS nodes using images captured by one camera. The first result presented is the variation of each indicator with the filter size. Then, for each value of the filter size, an average of the two indices was calculated for the full set of 360 frames. Those are the values plotted for $I_{1,avg}$ and $I_{2,avg}$ shown in Fig. 9a.

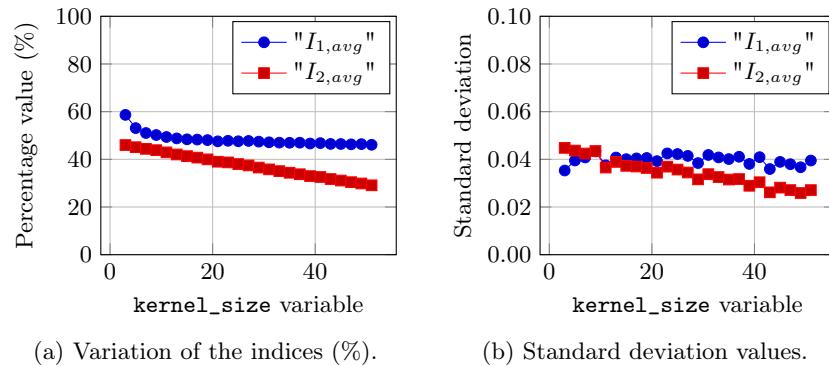


Fig. 9: Results of indices $I_{1,avg}$ and $I_{2,avg}$ for the experiment with one camera and two algorithms.

Since black pixels will turn non-black with the increase of the filter size, the A_T parameter increases. The opposite happens for the common area (A_C) because white pixels will be transformed into darker pixels as the filter size increases. Consequently, the weighted confidence area (W_{CA}) remains almost invariable.

The other analysis (Fig. 9b) concerns the standard deviation of each indicator for each filter size. The low values of the standard deviations values reflect a large uniformity in the samples analysed. These results, and potentially others,

demonstrate the usefulness of the architecture to study and tune parameters, either from each algorithm (not the case here) or from the merging technique it self (as was the case here).

An example of the usefulness of the architecture is shown in Fig. 10, where a classic computer vision and deep learning-based techniques were combined.

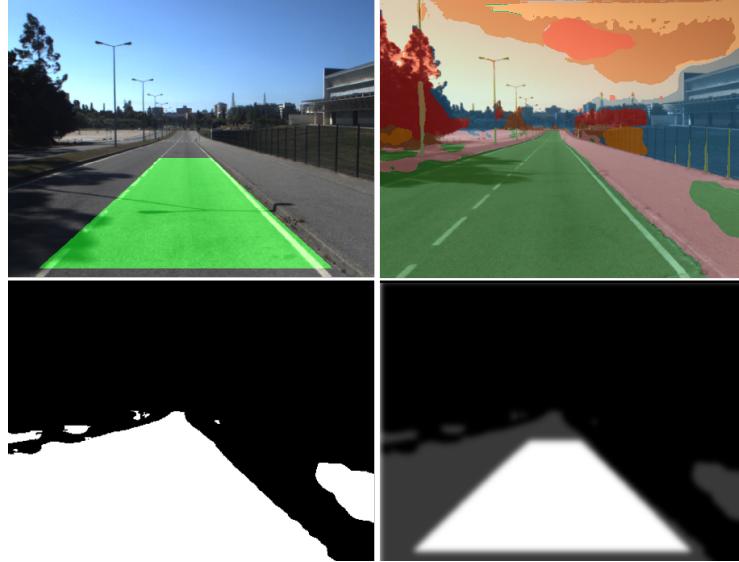


Fig. 10: Results using two detectors (classic and DL-based on top). The bottom left image represents the road zone given by DL algorithm. As can be seen, the imperfections in the deep learning detector are mitigated by the combination with the classical detector. The final result (bottom right) shows an enhanced representation of the road.

5.2 Results from Two Cameras and Two Algorithms

As explained in section 3.3, there are two options to launch the architecture in multi-camera mode: with or without a combination of the maps coming from each camera. If the two maps are not combined, the results would be the same as those presented in the previous sub-section. Therefore, the results presented in this sub-section are based on the combination of the maps coming from each camera. Since a warp transformation is applied to the polygons to be combined, the confidence map representation is expected to be more straight (road top-view). This implies that the blur caused by the application of the filter is smoother along the road lane border compared to the previous study.

In general, the results obtained are similar to those presented in the previous experiment. The only clear difference is the rate of change of both indices, which in this case is lower. This is due to the fact that, as mentioned earlier, the confidence map here is represented by a top-view perspective.

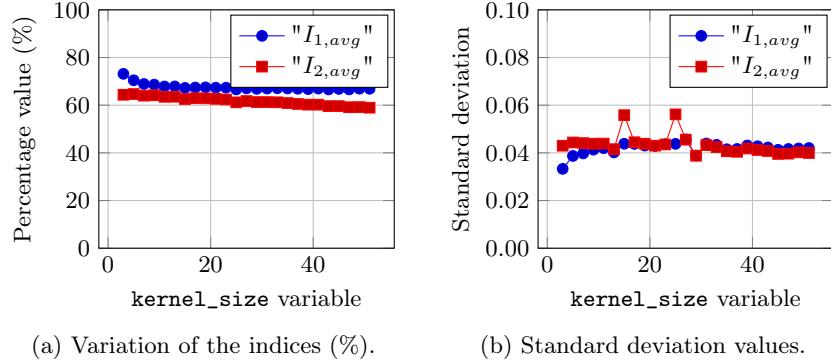


Fig. 11: Results obtained for the experiment related to the use of two cameras and the two algorithms.

6 Conclusion

This paper proposes a scalable architecture to merge multiple road detection algorithms, creating the conditions to obtain more robust detected road maps than by using the algorithms individually. The two types of output representations (polylines or regions of pixels) are converted into a unique representation to allow the merging procedures. This architecture has shown to be valuable also to combine traditional computer vision techniques and DL based classifiers to detect the road in more robust. Even though Deep Learning usually outperforms the traditional techniques it can fail by overestimation and traditional techniques can be used to limit that effect, although this issue still needs further studies and analysis. One planned next step is to migrate into a unified representation, probably based in occupancy grids, to merge data obtained also from LIDAR-based perception. In conclusion, although there is still space for improvements, the architecture proved to be a valid instrument to combine multiple source road detection algorithms with the possibility to tune them interactively, or, hopefully in the future, by an automatic procedure to optimise road detection algorithms and their parameters towards a more generic detector.

Acknowledgements

This work was partially supported by project UID/CEC/00127/2019.

References

1. Aly, M.: Real time detection of lane markers in urban streets. In: Intelligent Vehicles Symposium, 2008 IEEE. pp. 7–12. IEEE (Nov 2008)
2. Assidiq, A.A., Khalifa, O.O., Islam, M.R., Khan, S.: Real time lane detection for autonomous vehicles. In: 2008 Int. Conf. on Computer and Communication Engineering. pp. 82–88 (May 2008)

3. Boumini, F., Gingras, D., Lapointe, V., Pollart, H.: Autonomous vehicle and real time road lanes detection and tracking. pp. 1–6 (10 2015)
4. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 30, 88–97 (2009)
5. David Jenkins, M., Carr, T.A., Iglesias, M.I., Buggy, T., Morison, G.: A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. In: 2018 26th European Signal Processing Conf. (EUSIPCO). pp. 2120–2124 (Sep 2018)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *CoRR* abs/1512.03385 (2015)
7. Hou, C., Hou, J., Yu, C.: An efficient lane markings detection and tracking method based on vanishing point constraints. In: 2016 35th Chinese Control Conf. (CCC). pp. 6999–7004 (Jul 2016)
8. John, V., Kidono, K., Guo, C., Tehrani, H., Mita, S., Ishimaru, K.: Fast road scene segmentation using deep learning and scene-based models. In: 2016 23rd Int. Conf. on Pattern Recognition (ICPR). pp. 3763–3768 (Dec 2016)
9. Kluge, K., Lakshmanan, S.: A deformable-template approach to lane detection. *IEEE* pp. 54–59 (Sep 1995)
10. Li, L., Zheng, W., Kong, L., Ozguner, U., Hou, W., Lian, J.: Real-time traffic scene segmentation based on multi-feature map and deep learning. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 7–12 (Jun 2018)
11. Liu, S., Lu, L., Zhong, X., Zeng, J.: Effective road lane detection and tracking method using line segment detector. In: 2018 37th Chinese Control Conf. (CCC). pp. 5222–5227 (Jul 2018)
12. Oliveira, M., Santos, V., Sappa, A.D.: Multimodal inverse perspective mapping. *Information Fusion* 24, 108–121 (2015)
13. Pouyanfar, S., Chen, S.: Semantic event detection using ensemble deep learning. In: 2016 IEEE International Symposium on Multimedia (ISM). pp. 203–208 (Dec 2016)
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. *ArXiv* abs/1505.04597 (2015)
15. Su, C.Y., Fan, G.H.: An effective and fast lane detection algorithm. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.M., Monroe, L. (eds.) *Advances in Visual Computing*. pp. 942–948. Springer Berlin Heidelberg (2008)
16. TaSci, E., Ugur, A.: Image classification using ensemble algorithms with deep learning and hand-crafted features. In: 2018 26th Signal Processing and Communications Applications Conf. (SIU). pp. 1–4 (May 2018)