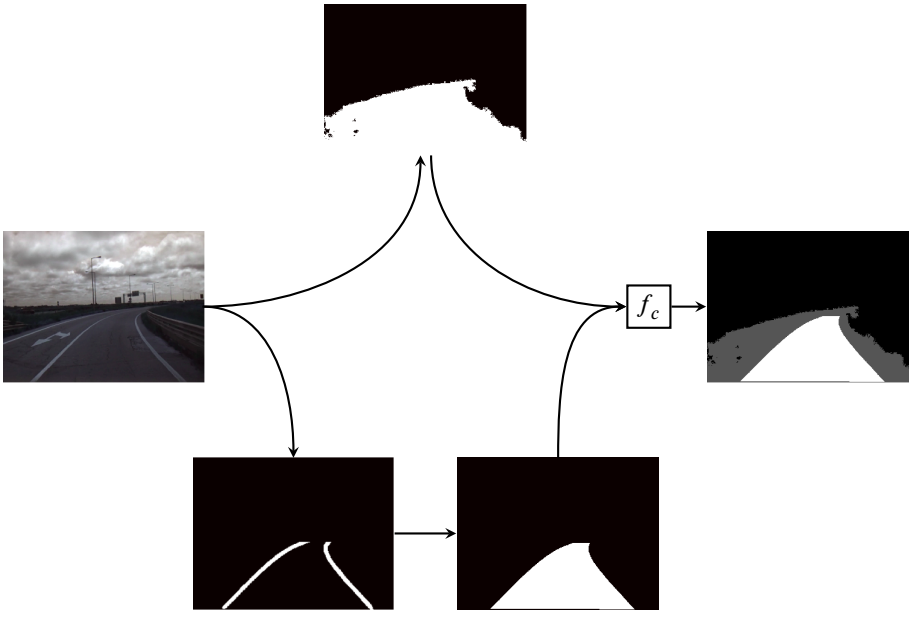
Graphical Abstract

Road Detection based on simultaneous Deep Learning Approaches

Tiago Almeida,Bernardo Lourenço,Vitor Santos



Highlights

Road Detection based on simultaneous Deep Learning Approaches

Tiago Almeida,Bernardo Lourenço,Vitor Santos

- Adaptation and tuning of ENet and LaneNet architectures to detect road features.
- Definition of a technique to combine in a weighted form simultaneous different road detection approaches.
- Demonstration of the validity of a scalable and redundant software architecture to merge multiple data sources and processing algorithms onboard the ATLASCAR2.

Road Detection based on simultaneous Deep Learning Approaches

Tiago Almeida[a,b], Bernardo Lourenço[a] and Vitor Santos[a,b]

[a]Department of Mechanical Engineering, University of Aveiro
[b]Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro

ARTICLE INFO

Keywords:
visual perception
data combination
deep learning
computer vision
road map detection
road lane lines detection
road segmentation
driving assistance

Abstract

One of the most important challenges for Autonomous Driving and Driving Assistance systems is the detection of the road to perform or monitor navigation. Many works can be found in the literature to perform road and lane detection, using both algorithmic processing and learning based techniques. However, no single solution is mentioned to be applicable in any circumstance of mixed scenarios of structured, unstructured, lane based, line based or curb based limits, and other sorts of boundaries. So, one way to embrace this challenge is to have multiple techniques, each specialized on a different approach, and combine them to obtain the best solution from individual contributions. That is the central concern of this paper. By improving a previously developed architecture to combine multiple data sources, a solution is proposed to merge the outputs of two Deep Learning based techniques for road detection. A new representation for the road is proposed along with a workflow of procedures for the combination of two simultaneous Deep Learning models, based on two adaptations of the ENet model. The results show that the overall solution copes with the alternate failures or under-performances of each model, producing a road detection result that is more reliable than the one given by each approach individually.

1. Introduction

The fields of Autonomous Driving (AD) and Advanced Driver Assistance Systems (ADAS) have carried out a wide range of studies that can shape the future of transportation by cars and other vehicles. One of the propelling topics fueling this changes is the field of Deep Learning (DL). Deep Learning algorithms are known to be superior in generalizing for several hard conditions that are the source of error in the classical algorithms, mainly changing lighting conditions, road occlusions, shadows, and different camera setups. Because the inclusion of sensors in cars is a current possibility, these algorithms could be used to understand the road scenario.

However, most algorithms perform poorly under some conditions, because the models are not able to learn to perform well in all situations, and datasets still do not reflect all the road scenarios possible. For example, some datasets use the same camera setup for all images, and do not generalize very well for other camera setups. Similarly, other datasets have images with low diversity of information, for example, datasets with only highway road scenarios do not generalize very well for other scenarios. Also, models trained for different tasks often perform differently for different conditions. A better solution would be to use multiple deep learning algorithms trained for different tasks and datasets, and have an algorithm to combine the outputs of these models into an unique road representation.

This work describes a novel road space representation method that combines two deep learning models trained to perform two different tasks: road segmentation and road lines detection. This method was found suitable and performant for both unstructured and structured roads.

2. Related Work

There are several algorithms in the literature for road segmentation and detection of the road lanes. Most classical algorithms are composed of several phases, such as pre-processing, feature extraction and model fitting [4, 3, 14]. Some of these classical algorithms even perform both tasks with the same pipeline [6, 15, 11]. Recently, Deep Learning has proven to be effective in training neural networks in a end-to-end fashion. That is, all the steps mentioned are performed by the neural network and are optimized together. These models have now outperformed the classical methods and are expected to improve over time. The most well-known and used architectures for object segmentation problems were based on [17, 26, 5]. Then, new more accurate and reliable approaches appeared in [22, 32, 8], in which one of them would give rise to one of the first real-time segmentation methods — ICNet [31]. Recently, the investigation in

ORCID(s):

this Computer Vision task (segmentation) goes towards obtaining the best trade-off between real time and accuracy, so new methods were presented in [23, 24, 29]. Besides the existing general architectures for object segmentation, several authors develop more restrict works for road segmentation in [9, 16] only based on Convolutional Neural Networks (CNNs). Moreover, more complex methods were also developed with the usage of CNNs and Recurrent Neural Networks (RNNs) techniques in order to increase the robustness of each architecture by Lyu [18, 19]. In terms of road lane lines segmentation, there are also several works, one example is LaneNet [20]. Also, in [12], the authors propose applying self attention distillation (SAD) to increase the robustness of existing architectures in the detection of road lane lines. This would be achieved by training models that output attention maps, which encode contextual information about the road scenario and that are then incorporated in an architecture as a supervision method. Finally, in [30], the authors use layers to perform the perspective transformation that makes the road lane lines detection easier.

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 in machine learning, where, for example, there is a technique that combines the predictions from multiple trained models to reduce variance and improve prediction performance [28, 25]. The objective is similar to the one presented in this work: combine different results in order to obtain a more accurate one.

3. Proposed Approach

In this work, we use the overall architecture conceived in [2] but with major changes regarding the processor algorithms as well as the combination method. Thus, we propose the usage of two deep learning convolutional models trained to perform different tasks, and try to merge their outputs to create a confidence map. This final output, the confidence map, is a grayscale image that represents the road scenario, where each pixel value corresponds to the respective confidence: 1 (white pixels) is the maximum confidence possible and 0 (black pixels) is the minimum confidence.

The modifications performed in this work allow to claim that, now, the system consistently produces road representations due to the combination of two DL algorithms instead of classical algorithms. As far as the combination method is concerned, which is detailed later, in this improved approach, not only it is possible to give more confidence to pixels that belong to a zone returned by more than one algorithm as being a road area (method presented in [2]), but also if those zones have a more similar shape with a polygon (shape closer to a road).

The two deep learning models used to perform the road/lanes segmentation tasks are based on the same neural network architecture, the ENet [22], which can achieve low-latency performance on semantic segmentation, which is a common application in the field of autonomous driving. The ENet model was trained to perform semantic segmentation, and it was modified to also perform road lines segmentation, based on the work of [20]. In the combination stage, a confidence map is obtained that has information about all the main features of the road: the main lane (also called *ego lane*), where the car is located, and the alternative lanes, where the car could drive in. In Fig. 1, an overview of the stages and algorithms can be seen.

In the following sections, the DL-based architecture, as well as the modifications performed to each of its applications are described.

3.1. The ENet architecture

The ENet [22] is a convolutional neural network for semantic segmentation that was optimized for real-time inference and high accuracy. This model proved to be effective in this task with the big advantage of being easy to train from *scratch*, not requiring any pre-training. This is due to its low capacity (a small number of parameters), which is also the reason for its fast inference performance.

Its design was shaped by several choices and influences from other works developed for this task. The most important choices are describes as follows:

Encoder-Decoder architecture The basic shape of this model shows semblance to other semantic segmentation models such as the SegNet [5], DeconvNet [21] and RedNet [13]. These models are composed of two stages, the encoder, and the decoder. The former downsamples the input image and transforms the raw pixels into high-level features. The latter does the opposite, upsampling the output of the encoder and using the encoded features to classify each pixel. In these networks, the outputs of the pooling layers, such as the pooling indices are preserved and used in the decoder to preserve spatial information.

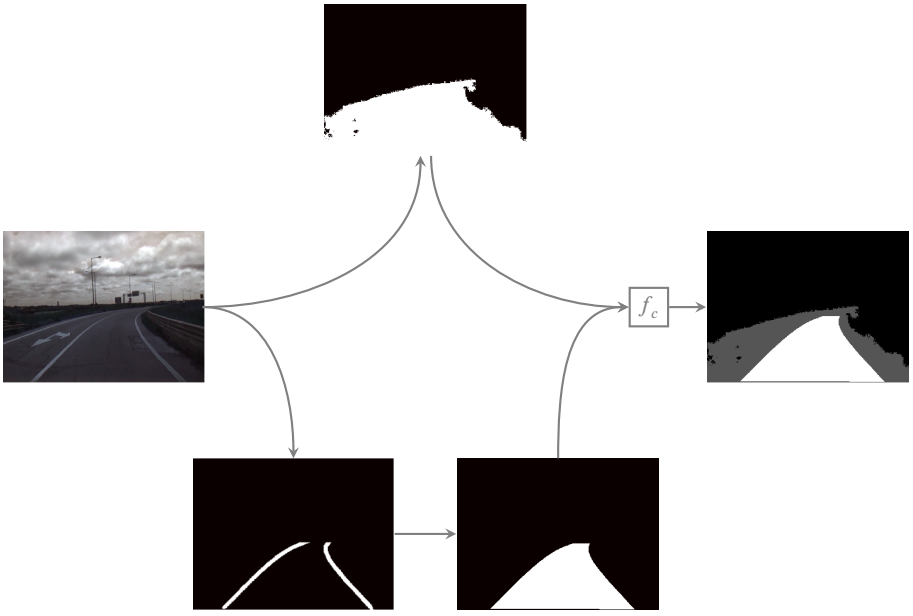


Figure 1: From left to right, the input image (left) is processed in parallel by the two models: the semantic segmentation (top image) and the line segmentation models (bottom left image). After that, the LaneNet output is transformed into a polygon (bottom right image) to, then, be combined with the region obtained by the ENet model to form the final confidence map (right), through the combination function (f_c). This is any function that classifies each segmented image (bottom right image and upper image) and gives as result a scalar that expresses the confidence of each of those images, which will be used for the final weighted combination. For this work, this function will be explained further on.

Bottleneck Blocks The bottleneck is the building block of this network. The idea behind a bottleneck block is to reduce the number of channels by a certain rate using a cheap depth-wise convolution so that the middle convolution has fewer parameters. In the end, the network is widened again with another depth-wise convolution. This strategy also applies some regularization to the model.

Dilated Convolutions Dilated Convolutions are commonly used in the task of semantic segmentation [8], as it allows the convolutions to have a wider receptive field. This enables the model to capture more context, without downsampling the feature maps. An equivalent transformation would be a pooling operation, which has the disadvantage of reducing the spatial information and implies an extra upsampling operation.

3.2. ENet for Semantic Segmentation

For the semantic segmentation task, a network was trained as described in [22]. The model was trained with the Cityscapes Dataset [10], which is composed of 5000 images, of which 3475 have pixel-wise annotations of 19 classes. For this task, the final convolution of the ENet model has 19 output channels. An example of the output of this model can be seen in Fig. 2.

For the training, the procedure in [22] was followed, with slight modifications:

- an additional data augmentation was performed, such as random rotation, random scaling, and color-jitter, in order for the model to generalize better in our road scenario;
- the model was trained using progressive size cropping, starting with crops from size 256×256 up to 784×784, to train with fewer epochs.

3.3. ENet for Road Lane Lines Segmentation

The LaneNet [20] is a modified ENet model for the task of road lane lines segmentation. This model has two branches (Fig. 3), sharing the first two stages of the ENet and the remaining stages are used as the backbone of each separate branch. The first branch is responsible for producing binary masks of the input image and the second generates

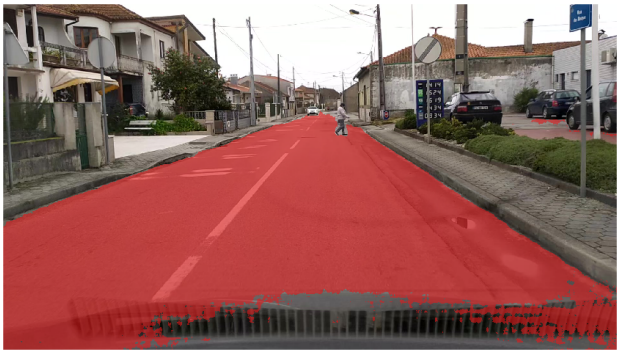


Figure 2: An example of the output of the semantic segmentation model. The segmented road is shown as the red overlay in the image.

dense embedding outputs. Embeddings are a mapping of the input pixels to a higher dimensional space, where the distances between points is a meaningful property for the instance segmentation.

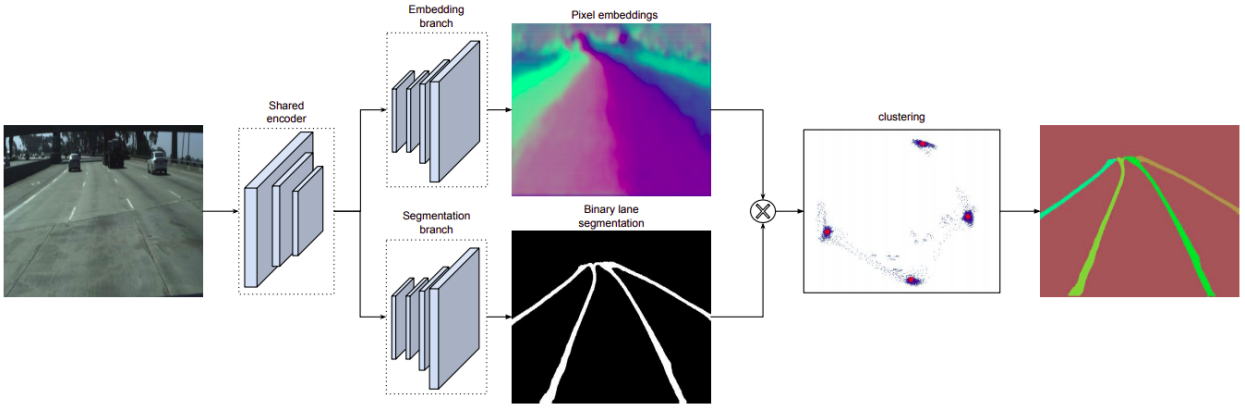


Figure 3: The LaneNet Model is composed of two branches. The segmentation branch (represented at the lower part of the image) is trained to produce binary masks of the input image (left). Besides being a binary mask producer, this architecture also provides embeddings, through its top branch: the embedding branch. It yields the pixel embeddings of the input image. These two branches, together, cluster the lane embeddings (blue dots), by repelling groups of pixels (clusters) from different lines and attracting embeddings of the same lines to the respective cluster center (red points). After the cluster post-processing stage, the output is an image with one channel, with the respective lines indexed [20].

The segmentation part produces binary masks of the input image, where the background of the image is black pixels and the road lines are white pixels. Since these two classes are highly unbalanced, as the majority of the pixels are background, alternatively to the original LaneNet approach (where the cross-entropy loss function is used), the dice loss [27] is used to solve this issue.

The embedding branch learns to map the lane pixels into a high-dimensional space, where the pixels of the same lane are closer together and pixels of different lines are farther apart. The dimension of the embedding space is a hyper-parameter that was determined by a grid-search, and in this case, it was set to 4. This model learns this representation mapping through the discriminative loss function [7].

This network was trained with the TuSimple Dataset [1], which consists of 3626 video frames as the training set and 2782 video frames as the testing set. The annotations used for the dataset labeling are polylines (for the lane markings). Finally, some data augmentation techniques were performed in the training set images such as: image resizing, dimming, and rotation.

During inference, the clustering is made by applying the *MeanShift* algorithm, which is a suitable function for finding dense cluster centers.

3.4. Output Post-Processing

After each of the networks returns its output, post-processing is done to combine the two outputs in a further step (Sec. 3.5). This procedure consists of applying simple techniques that enable the suppression of small groups of pixels (blobs) as well as the lane lines sorting of the LaneNet output. Therefore, the initial phase of this procedure consists of cleaning the top of the image, as there are cases where the road lines intersect each other. This would affect the polygon creation, which is a procedure mentioned in [2] and also applied in this work. Moreover, the top of the image is mostly composed of irrelevant information such as the sky. After that, small blobs (less than 4500 pixels) are removed from the image, since the outputs of both models have quite considerable areas, so it was possible to mitigate some small *false* detections. These post-processing steps (Fig. 4) are common to both models.

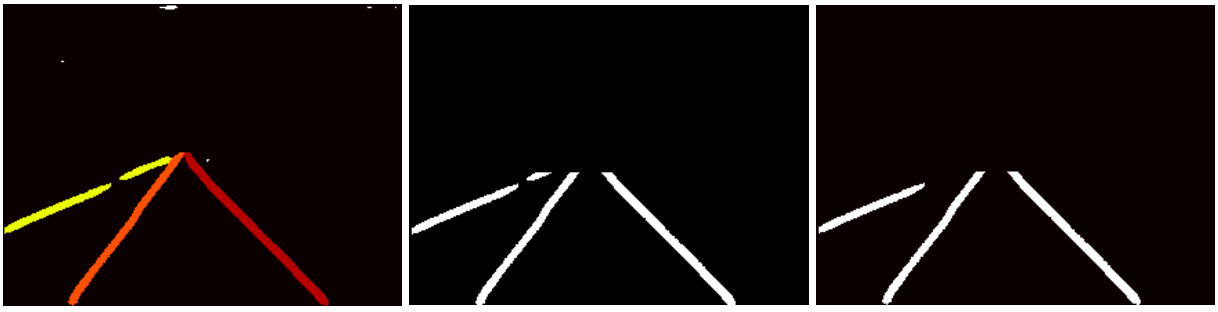


Figure 4: Example of an output post-processing. The input image (left) provided by the LaneNet model is binarized and its top is cleaned (center). Finaly, the small blobs in the image are removed (right).

In the case of the LaneNet output, it is necessary to sort the lines in order to proceed to the correct construction of the polygons. For this purpose, the coordinates of the centroids of each detected line are ordered horizontally, from left to right.

In short, the image returned by each model is cleaned. Furthermore, the lines in the LaneNet output are ordered to later built a polygon that will be combined with the road zone provided by the semantic segmentation model.

3.5. The combination approach

As described in [2], the outputs that return road lane lines pass through an intermediate phase of the combination architecture, to build a polygon based on those lines. This method consists of finding the extreme points of each line (after these lines are sorted) and then the drawing of lines connecting those points. Finally, the polygon is filled to yield the final representation of Fig. 5.

After the outputs of each model are in the same data type (polygons or closed zones), the problem is what is the best approach to combine these polygons. It is necessary to highlight some syntactic notation that will be used from now on: regions are closed zones of white pixels (e.g. in Fig. 5 there are 2 regions) and road representation means the two types of road representation that are combined at this stage (the two central images in Fig. 1).

The method now introduced is an alternative to the one presented in [2]. Here, the road representations are weighted summed based on an image descriptor: the solidity. Solidity is a measurement of the ratio between the regions and the area of its convex hull. Therefore, more confidence is given to zones whose shape is more similar to a road. Hence, this combination method allows having a road representation — confidence map — whose lighter pixels were returned from a higher number of algorithms and belong to a zone with a high solidity value. Concretely, the confidence map is given by:

$$C_M = \sum_{i=1}^{n} (w_i \times P_i), \tag{1}$$

where C_M is the final confidence map, based on n polygons P, and w_i is the weight of each polygon. This is the output

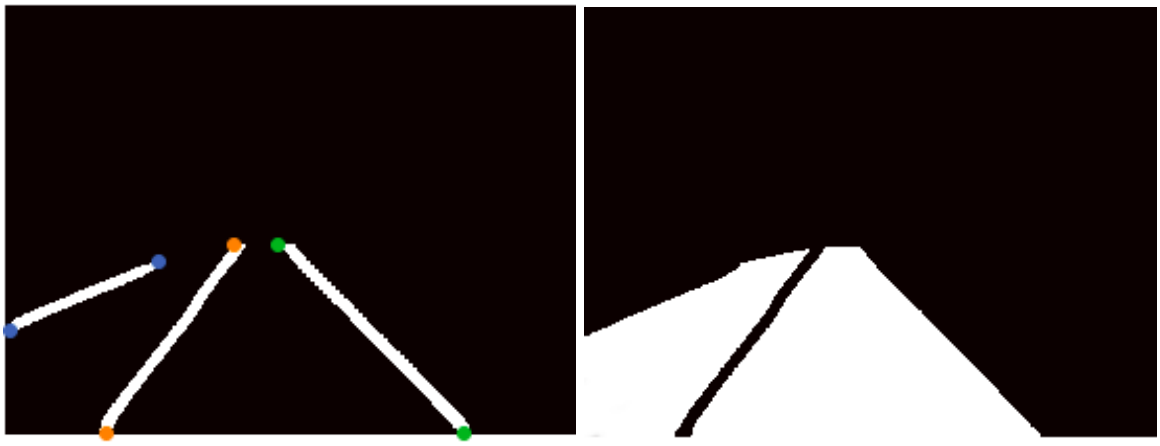


Figure 5: Example of a polygon creation to an ouptut based on the road lane lines. The minimum and maximum of each lane line are found (left) and after the polygons are closed, they are filled (right).

of the combination function that is presented in Fig. 1 as f_c. Each weight is given by:

$$w_i = \frac{1}{N} \times s_i, \tag{2}$$

where N is the number of road representations (or number of processor algorithms) and s_i is the solidity of each region. The weights distribution to the final confidence map construction are explained visually in Fig. 6.

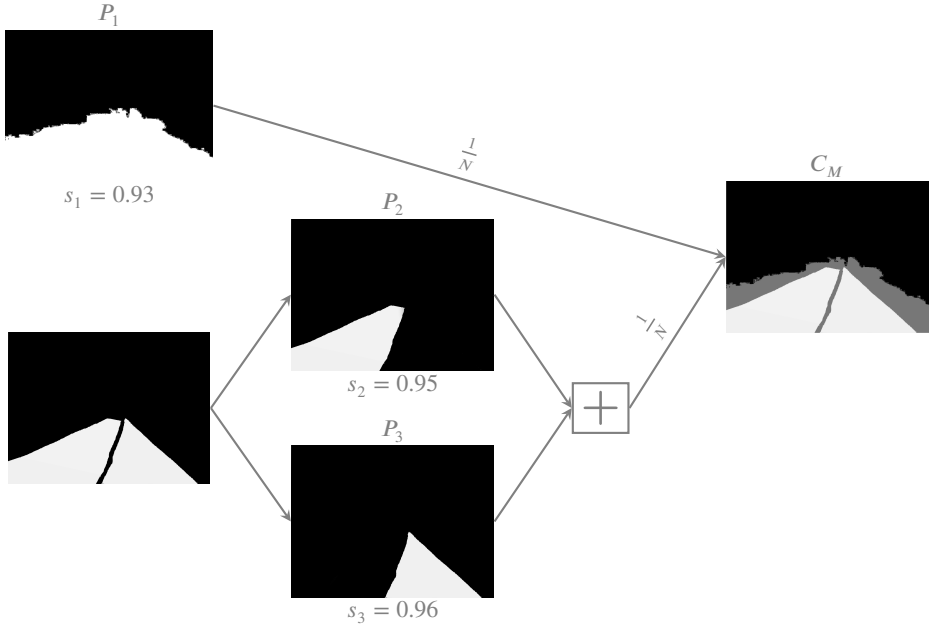


Figure 6: The raw ENet model is a region P_1 that weighs $\frac{1}{N} \times s_1$ on the final confidence map (C_M) and the output that is given by the LaneNet model could be represented with more than one polygon. Thus, in this case those polygons are separated and weigh $\frac{1}{N} \times s_2$ and $\frac{1}{N} \times s_3$, respectively, on the final confidence map.

4. Experimental Infrastructure

This work was developed under the ATLASCAR2 project [1], which is a Mitsubishi i-MiEV equipped with cameras, LiDARs, and other sensors. In this work, we just used one sensor — one PointGrey Flea3 camera — installed mainly on the roof-top of the car, as can be seen in Fig. 7. This camera acquired the images used in the experiments described in section 5. Additionally, it is also important to highlight that the two sets of images used to evaluate this work were acquired from two different camera setups: the main setup (shown in Fig. 7) and a second one located near the windshield, inside the vehicle. Both sets of images were used to test and assess the method. The used software architecture and framework for the development is ROS (Robot Operating System).

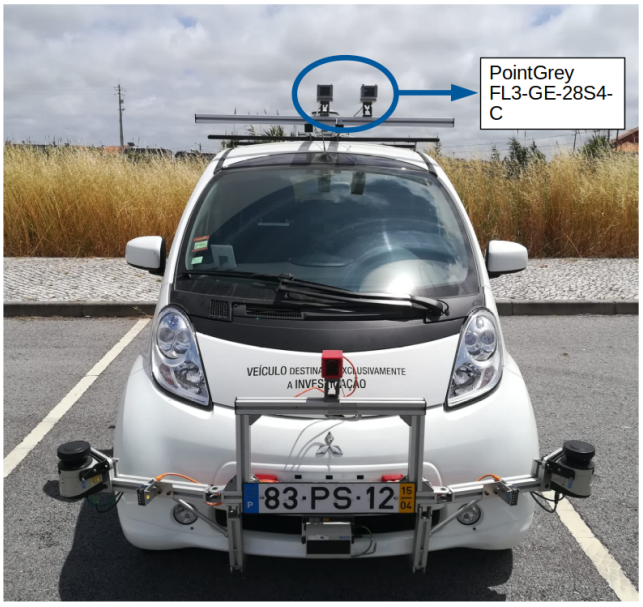


Figure 7: The ATLASCAR2 vehicle. This is one of the camera setups (car roof) used for this work. The other one is inside the vehicle in the windshield.

The training of the deep learning models was done in a single *NVidia RTX2080ti* and with the *PyTorch* framework.

5. Experiments and Results

Regarding the experiments, five examples of different road scenarios and the behavior of each model as well as their combination are shown. These experiments assess qualitatively the confidence map creation as well as each model's performance in challenging environments. After that, we show some statistic results for a full-set of 5000 frames in a usual environment for the ATLASCAR2 vehicle. Here, the number of detected regions provided by each model is presented.

The first experiment is a roundabout entrance (Fig. 8), which is a challenging environment for the LaneNet model since the input image does not contain the right road lane line, which could affect the model performance. Nevertheless, the model detects the difference between the right road lane line and the sidewalk as a road line which is an acceptable approximation to the road lane line. On the final confidence map, the small blobs do not appear due to the post-processing done on the models' outputs.

The second experiment consists of a highway environment (Fig. 9), which is a less-challenging environment because the road lane lines are well represented in the input image. Both models perform adequately, therefore the final confidence map represents the lanes with high accuracy.

The third experiment (Fig. 10) represents a roundabout; the ENet performs accurately, but there are no regions provided by the LaneNet model since the two road lane lines needed to build a polygon do not appear in the input image. Hence, the output of the LaneNet is, as could be expected, an empty set of information. In this case, the

[1] http://atlas.web.ua.pt/

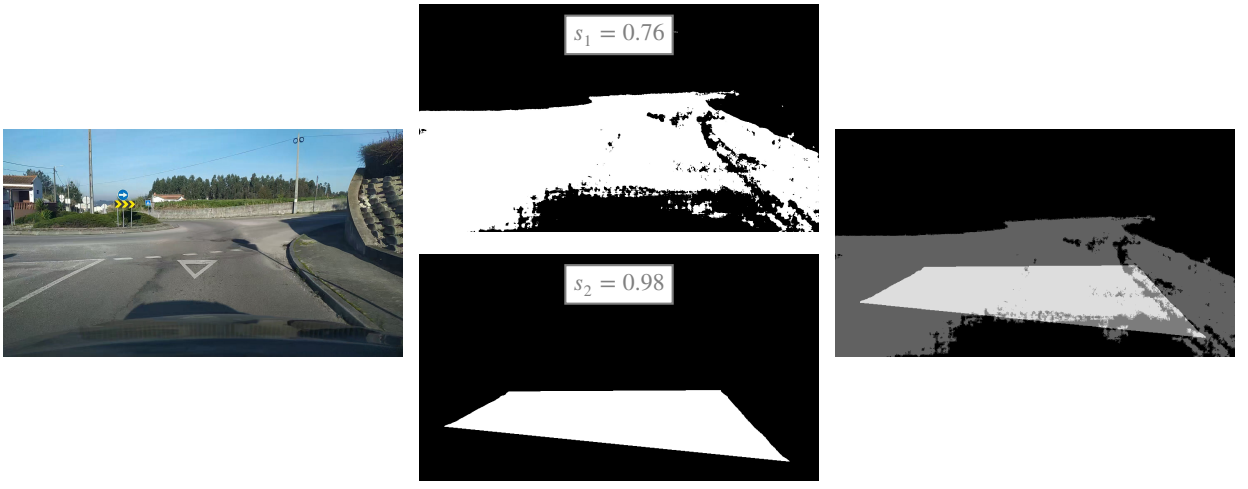


Figure 8: The first example of the experiments consists of a roundabout entrance. This is a challenging road scenario for the models since there is no road lane line on the right side. First, the input (left) is processed in parallel by the two models. Both model outputs are post-processed (middle), and then these regions are combined, yielding the final confidence map (right). Here, it is clear the contribution of the solidity combined with the proportionality between the number of algorithms that return an area as road and the pixel values of the final confidence map. Therefore, if more algorithms return a certain area detected as road and the solidity value of that zone is high, then the pixel values of that zone (confidence) are high on the final confidence map. Moreover, if only one of the algorithms was used, the result would either be limited in terms of road space detection (for the LaneNet model alone) or it would hav less information in terms of road lane (for the ENet model alone).

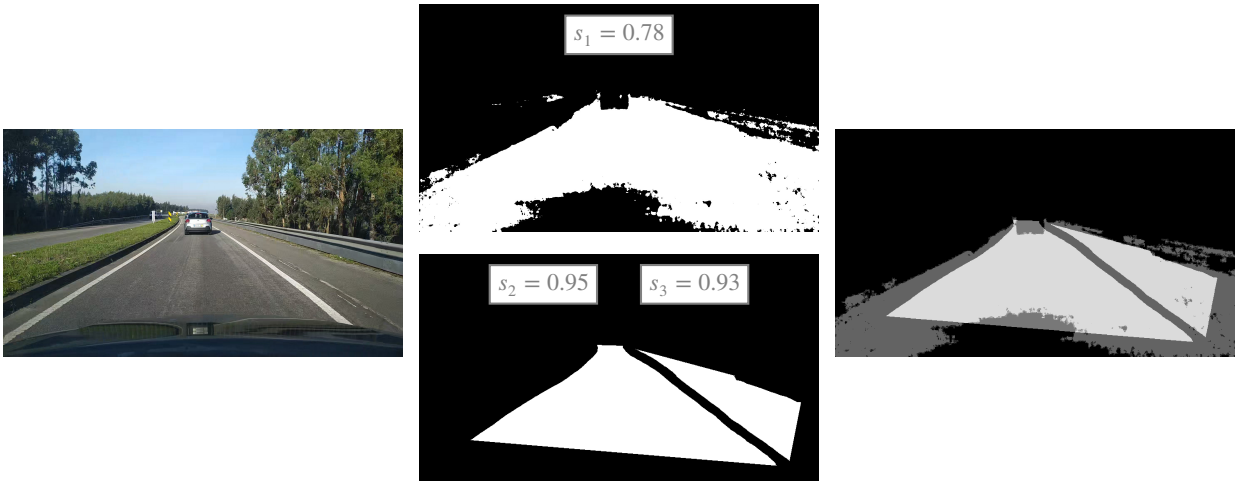


Figure 9: The second example of the experiments consists of highway scenario detection. Here, the road lane lines are well represented in the road scenario, thus, the models do not have problems with each feature detection (road and lines). First, the input (left) is processed in parallel providing two regions (middle). After post-processing each model's output, these regions are combined, taking into account the respective solidity (right).

representation of the road is made exclusively with the semantic segmentation component of the method. The more challenging the road scenario, the more this combination technique makes sense, because in this case, it is so difficult to detect road/lanes that the final confidence map is a grey area (one of the algorithms could not return anything). Thus, the certainty/confidence of the final road representation depends also on the difficulty of the assessed road scenario. This implies that if one of the models is not performant in one type of environment then it will not have a major impact on the final confidence map (due to either its low solidity value or an empty set of information).

The fourth example (Fig. 11) shows another challenging situation to create a reliable confidence map. Both models

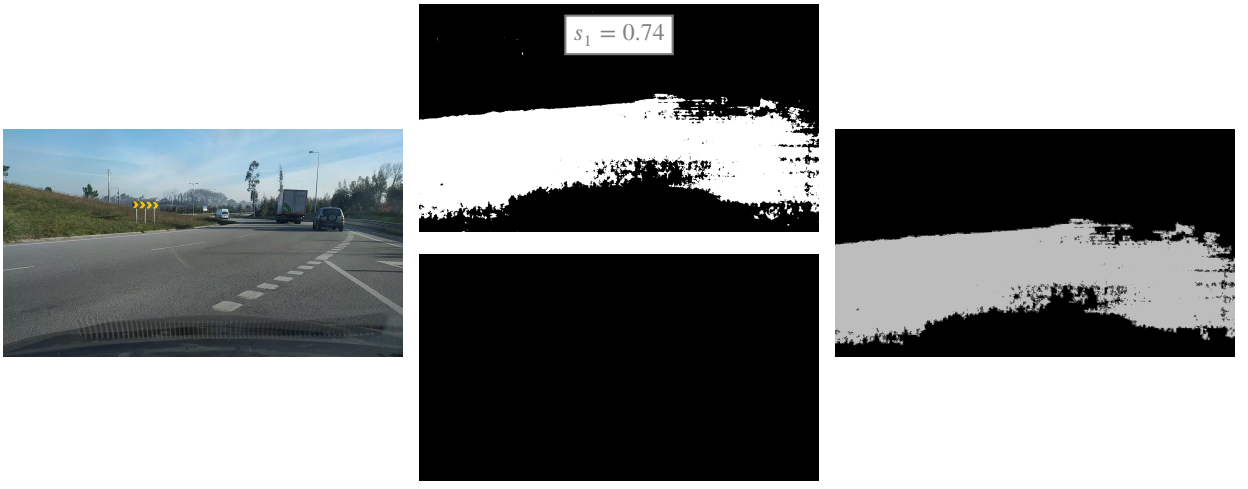


Figure 10: The third example of the experiments consists of a scenario inside a roundabout. On this scenario, the road lane lines are difficult to detect. First, the input (left) is processed in parallel by the two models contributing with two regions (middle). The road lane lines are not detected by the LaneNet model as can be seen in the bottom image. The final confidence map (right) is solely based on the ENet model contribution.

perform well in an overshadowed environment, which demonstrates the generalization and precision of these DL-based architectures.

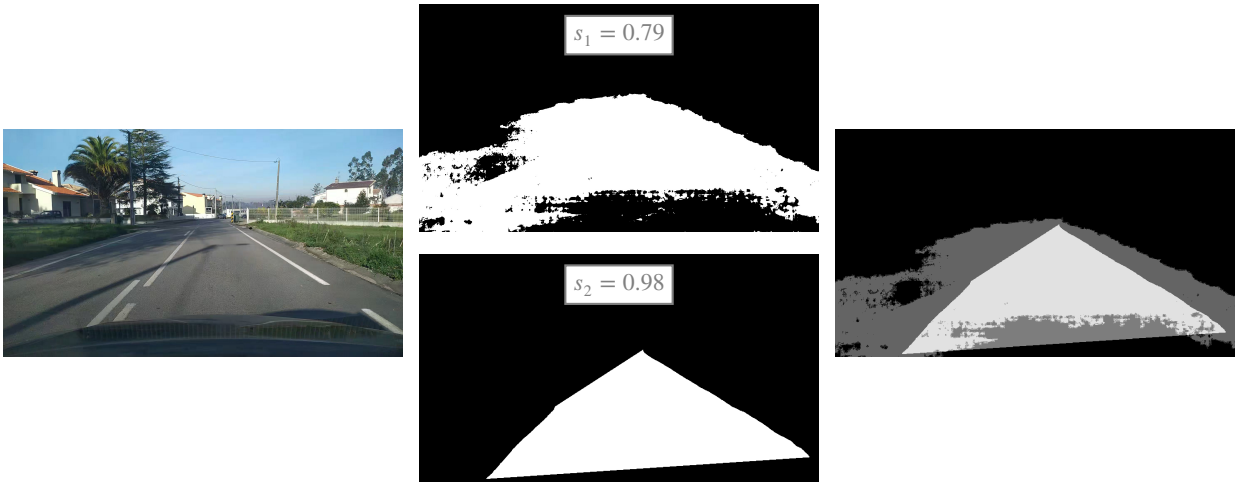


Figure 11: The fourth example of the experiments consists of another challenging road scenario because of the overshadowed road lanes. First, the input (left) is processed in parallel by the two models outputting the respective road features. After that, these outputs are post-processed (middle), they give rise to the final confidence map (right). Once again, if just one of the models was used, either there would be no distinction between the *ego lane* and parallel lanes (exclusive usage of ENet model) or there would be no access to information from parallel lanes (exclusive usage of LaneNet model).

The last experiment consists of an occlusion of a road line (Fig. 12). This could have had a negative impact on the detection of the road line. However, the contribution to the final confidence map was not affected, which demonstrates the effectiveness of LaneNet. Likewise, ENet architecture also shows no problems in distinguishing these two classes (vehicles and road). Finally, the confidence map illustrates accurately the real road scenario.

Finally, a study was performed to evaluate the quality of each method versus the joint performance of the combination method. As mentioned before, this study was performed for a full set of 5000 frames which contains several types of road scenarios (highways, roundabouts, regular roads). One of the most interesting highlights of the study was the creation of a confidence map for all image frames, which means that every frame generated a confidence map

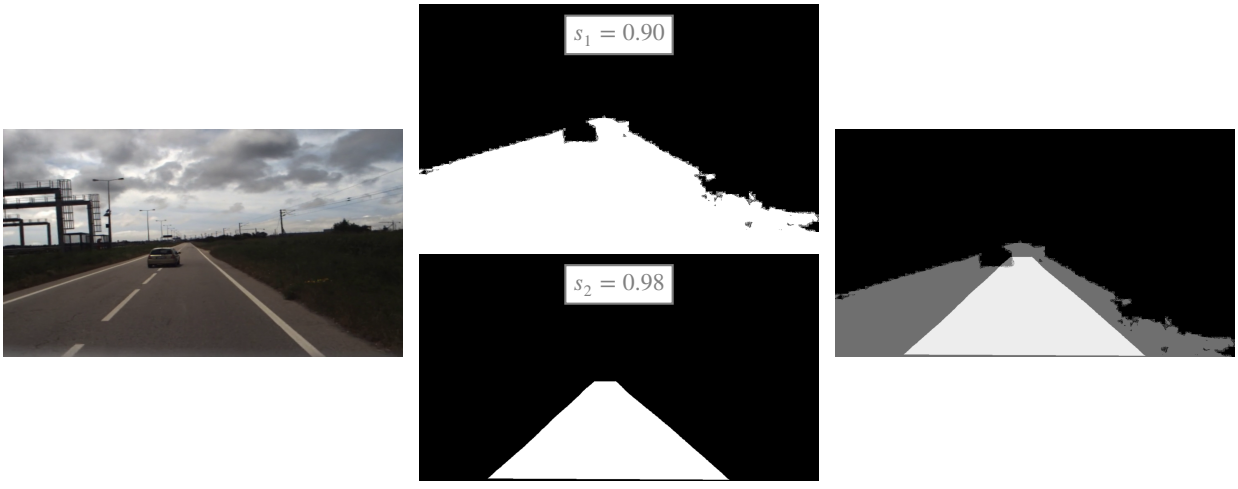


Figure 12: The fifth experiment consist of a defying road scenario (left), since there is a road lane line occlusion. The post-processed outputs (middle) show the robustness of both models. Finally, the confidence map (right) is composed by the two road representations with higher confidence values since both solidity values are higher than in the other four cases.

based on at least one of the outputs of both models. This study also allows to compare the number of regions that each model contributed to the final confidence map (Fig. 13). By observing this figure, we can conclude that the ENet model always contributes at most with one region, which is expected because every frame is a road representation. On the other hand, this does not happen on the LaneNet architecture (especially in the range between frames 3500 and 4500), since in some frames there are no road lines, namely when the vehicle is driving in roundabouts. It is also possible to observe that, sometimes, the LaneNet model returns more lines than expected since the maximum number of lanes of this test set is 3. This could be overcome by training this model with a more generalized dataset since the one that was used is limited.

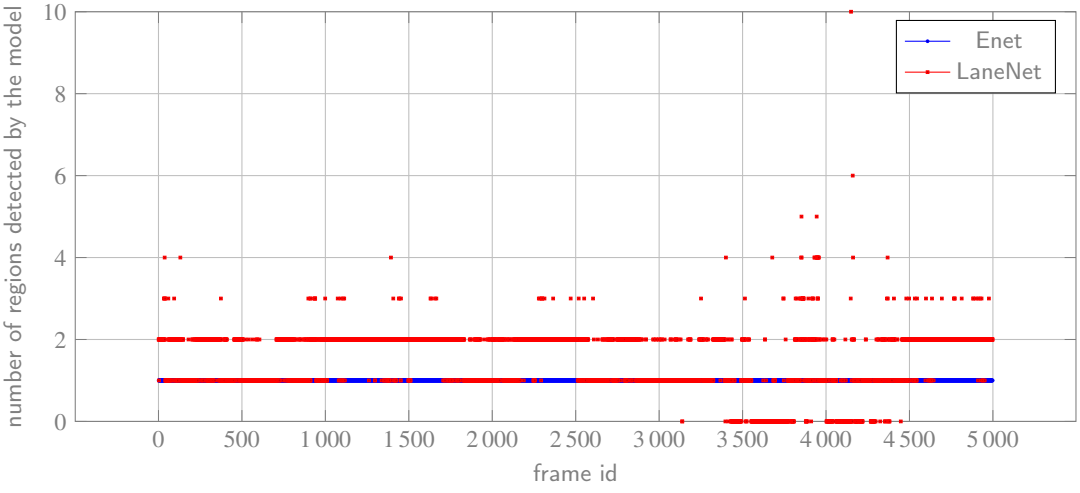


Figure 13: Number of regions detected by each model for the acquired set. As can be seen, on average, the number of regions detected by the LaneNet is two, which matches the real number of lanes. There are, however, some outliers, on a set of frames that were taken on a roundabout, where the LaneNet struggles to detect valid regions. The segmentation model detects, as expected, a single region on all frames.

Another analysis concerns the number of detected lanes on each frame, as shown in Fig. 14; as can be seen in this figure, in 38% of the frames the LaneNet model contributed with one lane to the final confidence map, while in 51%

of the times it contributed with two lanes. Compared to the real values, i.e. the real values of road lanes in each frame, there is a higher detection of one road lane versus two road lanes. Nonetheless, in all observed cases, when only one lane was detected, it was the *ego lane*, that is, the lane where the vehicle is navigating; this allows to conclude that this disparity of the detection versus the real number of lanes does not compromise the solution presented for most navigation purposes. One example of these cases is shown in Fig. 15. In summary, if the situations of one or two lanes are added up together, a total of 82% frames with one or two lanes in the real road is obtained, and the proposed method detected a total of 89% frames in those circumstances. This difference of 7% in the total number of frames where no lanes are detected is small enough to be compensated by a continuous road perception system that can integrate and obviate potential isolate road detection failures without consequences in a normal human timescale during the driving action.

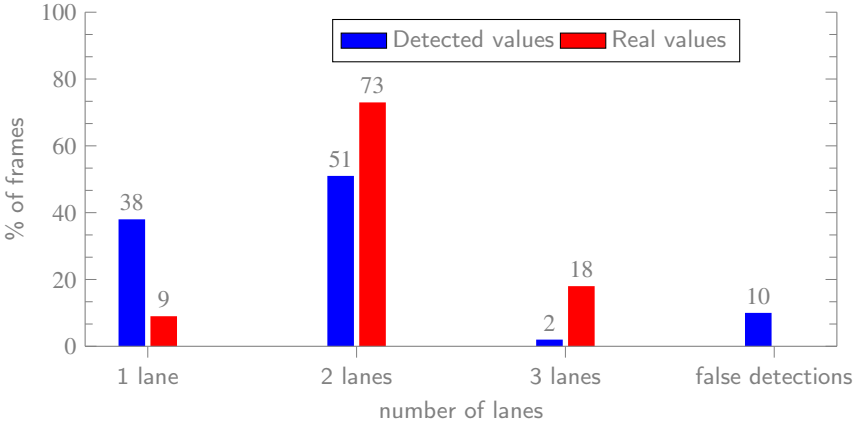


Figure 14: Histogram of the number of lanes detected by the LaneNet model. The false detections occur when the number is greater than 3 since in the used set the maximum number of lanes is also 3.

Figure 15: An example that illustrates the detection of only one road lane, when there are two road lanes. When this occurs in the tests made, the ego lane is always detected, which does not adversely affect the final confidence map. In fact, the system is just not providing all the existing information, but the one provided is reliable and, for most circumstances, enough for navigation purposes.

6. Conclusion

This work presents a new method for the combination of the output of multiple Deep Learning networks for road detection. This combination produces a confidence map that encodes the probability of each pixel in the image to be part of a road. This confidence map is paramount for further procedures, such as the navigation planners. By its intrinsic definition, this approach is naturally able to achieve a better overall performance than each individual algorithm because the best of each technique is always favored. When one technique fails or produces less reliable or less consistent results, the other technique that runs in parallel prevails.

Another advantage of the approach is that the number of simultaneous algorithms can be increased, and each algorithm can be updated or replaced with more confidence, because the final detection does not depend entirely on the new algorithm. This could be an effective test-bed for experimentation in autonomous driving. The underlying ROS based architecture that supports the implementation of multiple sources and multiple algorithms simultaneously provides a strong framework for concurrent and complementary techniques that allow scalabilty and redundancy, hence, robustness and safety in autonomous driving or driving assistance.

The Deep Leaning approaches used in the work have proved to perform better than most classic techniques, but it is not yet definitive that one single DL network is able to perform universally and this work will continue to exploit future developments for robust road detection.

This work used a simple descriptor (region solidity) to weigh the importance of each detected region for the creation of the final confidence map. Other descriptors can be explored and other weighting techniques can be developed to reach a robust later fusion technique.

Future perspectives are then wide open, both in the tuning of other Networks and detection algorithms, and also in the weighting technique to merge or fuse the results of those algorithms. Perhaps a specialized network to perform the fusion will be the next major challenge or the usage of an RNN (Recurrent Neural Network) to compute a time-based fusion instead of a frame-based combination.

Acknowledgements

This work was partially supported by Project SeaAI-FA_02_2017_011 and Project PRODUTECH II SIF- POCI-01-0247-FEDER-024541.

References

[1] , . Tusimple competitions for cvpr2017. http://github.com/TuSimple/tusimple-benchmark. Accessed: 2020-05-02.

[2] Almeida, T., Santos, V., Lourenço, B., 2020. Scalable ros-based architecture to merge multi-source lane detection algorithms, in: Silva, M.F., Luís Lima, J., Reis, L.P., Sanfeliu, A., Tardioli, D. (Eds.), Robot 2019: Fourth Iberian Robotics Conference, Springer International Publishing, Cham. pp. 242–254.

[3] Aly, M., 2008. Real time detection of lane markers in urban streets, in: Intelligent Vehicles Symposium, 2008 IEEE, IEEE. pp. 7–12.

[4] Assidiq, A.A., Khalifa, O.O., Islam, M.R., Khan, S., 2008. Real time lane detection for autonomous vehicles, in: 2008 Int. Conf. on Computer and Communication Engineering, pp. 82–88.

[5] Badrinarayanan, V., Kendall, A., Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR abs/1511.00561. URL: http://arxiv.org/abs/1511.00561, arXiv:1511.00561.

[6] Bounini, F., Gingras, D., Lapointe, V., Pollart, H., 2015. Autonomous vehicle and real time road lanes detection and tracking, in: 2015 IEEE Vehicle Power and Propulsion Conference (VPPC), pp. 1–6. doi:10.1109/VPPC.2015.7352903.

[7] Brabandere, B.D., Neven, D., Gool, L.V., 2017. Semantic instance segmentation with a discriminative loss function. CoRR abs/1708.02551. URL: http://arxiv.org/abs/1708.02551, arXiv:1708.02551.

[8] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR abs/1606.00915. URL: http://arxiv.org/abs/1606.00915, arXiv:1606.00915.

[9] Chen, P., Hang, H., Chan, S., Lin, J., 2019. Dsnet: An efficient CNN for road scene segmentation. CoRR abs/1904.05022. URL: http://arxiv.org/abs/1904.05022, arXiv:1904.05022.

[10] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. CoRR abs/1604.01685. URL: http://arxiv.org/abs/1604.01685, arXiv:1604.01685.

[11] Hou, C., Hou, J., Yu, C., 2016. An efficient lane markings detection and tracking method based on vanishing point constraints, in: 2016 35th Chinese Control Conf. (CCC), pp. 6999–7004.

[12] Hou, Y., Ma, Z., Liu, C., Loy, C.C., 2019. Learning lightweight lane detection cnns by self attention distillation. arXiv:1908.00821.

[13] Jiang, J., Zheng, L., Luo, F., Zhang, Z., 2018. Rednet: Residual encoder-decoder network for indoor RGB-D semantic segmentation. CoRR abs/1806.01054. URL: http://arxiv.org/abs/1806.01054, arXiv:1806.01054.

[14] Kluge, K., Lakshmanan, S., 1995. A deformable-template approach to lane detection. IEEE , 54–59.

[15] Liu, S., Lu, L., Zhong, X., Zeng, J., 2018. Effective road lane detection and tracking method using line segment detector, in: 2018 37th Chinese Control Conf. (CCC), pp. 5222–5227.

[16] Lo, S., Hang, H., Chan, S., Lin, J., 2019. Multi-class lane semantic segmentation using efficient convolutional networks. CoRR abs/1907.09438. URL: http://arxiv.org/abs/1907.09438, arXiv:1907.09438.

[17] Long, J., Shelhamer, E., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. CoRR abs/1411.4038. URL: http://arxiv.org/abs/1411.4038, arXiv:1411.4038.

[18] Lyu, Y., Huang, X., 2018a. Road segmentation using CNN with GRU. CoRR abs/1804.05164. URL: http://arxiv.org/abs/1804.05164, arXiv:1804.05164.

[19] Lyu, Y., Huang, X., 2018b. Roadnet-v2: A 10 ms road segmentation using spatial sequence layer. CoRR abs/1808.04450. URL: http://arxiv.org/abs/1808.04450, arXiv:1808.04450.

[20] Neven, D., Brabandere, B.D., Georgoulis, S., Proesmans, M., Gool, L.V., 2018. Towards end-to-end lane detection: an instance segmentation approach. CoRR abs/1802.05591. URL: http://arxiv.org/abs/1802.05591, arXiv:1802.05591.

[21] Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. CoRR abs/1505.04366. URL: http://arxiv.org/abs/1505.04366, arXiv:1505.04366.

[22] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., 2016. Enet: A deep neural network architecture for real-time semantic segmentation. CoRR abs/1606.02147. URL: http://arxiv.org/abs/1606.02147, arXiv:1606.02147.

[23] Poudel, R.P.K., Bonde, U., Liwicki, S., Zach, C., 2018. Contextnet: Exploring context and detail for semantic segmentation in real-time. CoRR abs/1805.04554. URL: http://arxiv.org/abs/1805.04554, arXiv:1805.04554.

[24] Poudel, R.P.K., Liwicki, S., Cipolla, R., 2019. Fast-scnn: Fast semantic segmentation network. CoRR abs/1902.04502. URL: http://arxiv.org/abs/1902.04502, arXiv:1902.04502.

[25] Pouyanfar, S., Chen, S., 2016. Semantic event detection using ensemble deep learning, in: 2016 IEEE International Symposium on Multimedia (ISM), pp. 203–208.

[26] Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597. URL: http://arxiv.org/abs/1505.04597, arXiv:1505.04597.

[27] Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. CoRR abs/1707.03237. URL: http://arxiv.org/abs/1707.03237, arXiv:1707.03237.

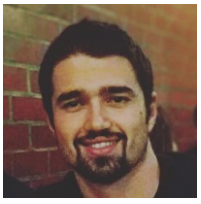
[28] TaSci, E., Ugur, A., 2018. Image classification using ensemble algorithms with deep learning and hand-crafted features, in: 2018 26th Signal Processing and Communications Applications Conf. (SIU), pp. 1–4.

[29] Wang, Y., Zhou, Q., Wu, X., 2019. Esnet: An efficient symmetric network for real-time semantic segmentation. CoRR abs/1906.09826. URL: http://arxiv.org/abs/1906.09826, arXiv:1906.09826.

[30] Yu, Z., Ren, X., Huang, Y., Tian, W., Zhao, J., 2020. Detecting lane and road markings at a distance with perspective transformer layers. arXiv:2003.08550.

[31] Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J., 2017. Icnet for real-time semantic segmentation on high-resolution images. CoRR abs/1704.08545. URL: http://arxiv.org/abs/1704.08545, arXiv:1704.08545.

[32] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2016. Pyramid scene parsing network. CoRR abs/1612.01105. URL: http://arxiv.org/abs/1612.01105, arXiv:1612.01105.



Tiago Almeida graduated in Mechanical Engineering, obtaining a Master Degree in 2019. Currently, he is a research fellow at IEETA, University of Aveiro, in the fields of Computer Vision and Robotic Systems. His Master Degree allowed the writing of his first article and the respective presentation at the 4th Iberian Conference (ROBOT2019). He is now working on one of the tasks of the Produtech project, whose aim is to build a low-cost autonomous guided vehicle that is capable of knowing its localization in a factory installation, through Artificial Vision. His research interests include Deep Learning and Machine Learning applied to Computer Vision.

Bernardo Lourenço graduated in Msc. of Mechanical Engineering in 2018 at the University of Aveiro. Currently, he is a research fellow in the fields of Computer Vision and Robotic Systems at the Department of Mechanical Engineering at the University of Aveiro. He has participated in two conferences: the 2019 ICARSC Conference and the ROBOT2019 Conference. His research interests are deep learning, computer vision, programming and robotic systems.

Vítor Santos graduated in Electronics Engineering and Telecommunications in 1989 and obtained a Ph.D. in Electrical Engineering in 1995. He was a researcher in mobile robotics at the Joint Research Center, Italy. Currently, he is an Associate Professor at the Department of Mechanical Engineering, lecturing courses related to advanced perception and robotics. He has managed research activity on mobile robotics, advanced perception, and humanoid robotics, with the supervision or co-supervision of more than 100 graduate and post-graduate students, and more than 140 publications. He has been in the program committee of several national and international conferences and acts regularly as a reviewer for several international conferences and journals. At the University of Aveiro, he has coordinated the ATLAS project for mobile robot competition that achieved 6 first prizes in the annual Autonomous Driving competition and the development of ATLASCAR and ATLASCAR2, the first car with autonomous navigation capabilities in Portugal that won the first prize in the Freebots competition in 2011. He is one of the founders of Portuguese Robotics Open and co-founder of the Portuguese Society of Robotics and a researcher at IEETA, in the Intelligent Robotics and Systems Group, focused on autonomous driving and driver assistance.