



Using Temporal Information in Deep Learning Architectures to Improve Lane Detection Under Adverse Situations

D. Turrado , J. Koloda , and M. Rincón  

Artificial Intelligence Department, UNED, Madrid, Spain
dturrado2@alumno.uned.es, diego.turradoblanco@avl.com, jan@gini.net,
mrincon@dia.uned.es
<http://www.ia.uned.es>

Abstract. One of the fundamental challenges in the field of autonomous driving is the ability to detect dynamic objects, such as vehicles or pedestrians, and static ones, such as lanes, in the surroundings of the vehicle. The accurate perception of the environment under the long tale of driving scenarios is crucial for a safe decision making and motion planning.

Mainly, lane detection approaches still function on single-frame basis and do not exploit the (high) temporal correlation of the signals representing the perceived environment. Single-frame detection networks might work well under circumstances where the lanes are perfectly visible, but show a lack of performance under certain situations, like occlusions, shadows, rain, snow, lane degradation, etc. To address the aforementioned problem, this work proves how adding temporal information for lane binary segmentation improves substantially the performance of single-frame architecture under challenging and adverse situations.

Keywords: Lane detection · Recurrent neuronal networks · Adverse situations

1 Introduction

In recent years, the development of computer vision (i.e. new deep learning architectures), sensor technology (i.e. LIDAR) and processors units (i.e. GPU) has made big advances in the field of autonomous driving possible. The ultimate goal of many of these researches is to have a full picture of the environment around the vehicle, detecting dynamic objects such as vehicles or pedestrians, and static ones such as lanes, and having a semantic understanding of the surroundings which allows a proper motion planning. One of the key and at the same time basic features needed to enable autonomous driving is camera based lane detection. Once the lanes are detected, the vehicle can have a reference to position itself in the surrounding world, so the trajectory planning can calculate where to go, reducing the possibility of collision with vehicles driving in other lanes [4]. Therefore designing robust camera-based real-time lane detection systems is

a key milestone for autonomous vehicles (AD) and advanced driver-assistance systems (ADAS).

Most of these systems are based on the information contained in one single frame, and just a few of them take advantage of the information contained in previous frames [6]. This, of course, is not like human drivers work, being able to extrapolate and infer the position of the lanes under challenging situations, like shadows, light reflections, lane occlusion, etc. referring to the information of the past. This can be done because lanes are static, continuous objects on the street, with a huge overlapping between frames, meaning a highly related and temporal correlation of the signals. So the lanes in the actual frame could be partially inferred from the information in the n -last frames, even though the lanes might not be totally visible anymore.

Under those circumstances the performance of the state-of-the-art methods decreases, detecting the lanes erroneously, in another direction, or even not detecting it at all like in Fig. 1. The reason might be that with the information contained in the actual frame, is very difficult or even impossible to completely infer the position of all lanes. Due to the fact that modern systems have to work under really variable driving scenarios, working under all possible conditions is crucial to develop robust algorithms. This work shows how including temporal information can dramatically increase the performance of a network under challenging situations.



Fig. 1. Lane detection using LaneNet [6] under driving scenarios with shadows, light reflections and lane occlusions from the TuSimple dataset. Green lanes represent the ground truth and red, the ones inferred by the network.

Therefore, the main hypothesis that tries to prove this work is that the use of recurrent neuronal networks (or, in a more general way, the use of temporal information) improves the performance of neuronal networks under challenging and adverse situations.

2 Methodology

To prove this hypothesis we have selected a state-of-the-art neuronal network, LaneNet [5]. LaneNet is a real time lane detection architecture where the lane edge network is based on a light weight encoder/decoder. To reduce computational cost the encoder is based on the combination of depthwise separable convolutions and pointwise convolutions (1×1 convolutions). On the other hand, to recover input image resolution the decoder is based on sub-pixel convolution layers.

Without modifying this encoder/decoder structure we have incorporated a recurrent neuronal network similar as proposed in [6]. This architecture can therefore be understood as a fully convolutional neuronal network with a recurrent intermediate step to process time information. The encoder abstracts n -input images to n -feature maps which can be seen as time sequence information and be fed into the recurrent neuronal network. These feature maps contain the necessary information to detect the lanes and keep the time related information but have the advantage of its reduced size, which means that they can be handled well by the long short-term memory layers. The output of the ConvLSTM [2] is used as input to the convolutional neuronal network decoder which outputs an array of the same size as the input image containing the probability of each pixel belonging to a lane or not. The complete architecture is shown in Fig. 2.

This idea allows us to compare the same basic encoder/decoder network with just the influence of the recurrent neuronal network gathering temporal information.

3 Training

3.1 Dataset

The TuSimple dataset [1] is used for training and validation. It contains 6408 video clips (3626 for training and 2782 for testing) taken at different daytime and traffic conditions on US highways, under good and medium weather situations. Each video clip is a set of 20 frames with until 5 lanes, where just the last frame is labeled, and with a resolution of 1280×720 .

The annotation of each lane is done using polylines, defined by the intersection points between evenly horizontal distributed lines and each lane. On the other hand, because the proposed architecture is conceived to work at pixel level, classifying each one as lane or not lane (binary segmentation). Therefore, using these polylines as basis, ground truth frames have been generated where each pixel belongs to the class 0 (background) or 1 (lane). To do so, each lane has been segmented with a width of 5 pixels.

Since the proposed deep learning architecture is designed to work at pixel level, classifying each pixel as lane or non-lane (binary segmentation), these polylines have been used as a basis to generate a ground truth in which each pixel belongs to class 0 (background) or 1 (lane). For this purpose, each lane has been segmented with a width of 5 pixels.

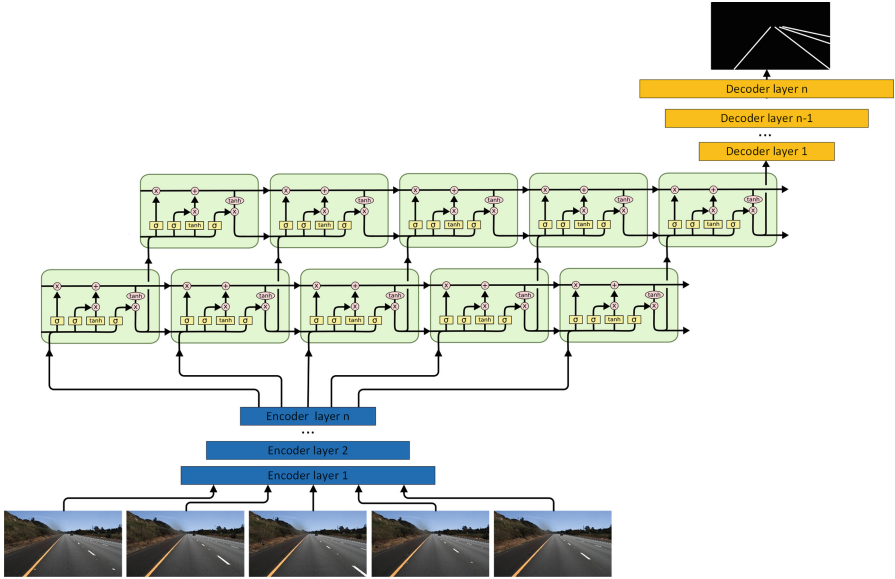


Fig. 2. Network architecture for the special case of 5 frames and ConvLSTM with 2 layers. The basic idea of this architecture for lane detection was proposed in [6], however the encoder/decoder has been adapted to match LaneNet structure.

Table 1. Characteristic of TuSimple dataset [1]. *Just the frame 20th is labeled.

Dataset	Train clips	Test clips	Frames per clip*	Resolution	Lanes
TuSimple	3626	2782	20	1280 × 720	≤5

3.2 Data Augmentation

In order to avoid over-fitting problems and increase performance, data augmentation has been used to train all models. Brightness shifting, gamma correction, adding small amount of noise and random horizontal flip and translation has been implemented.

3.3 Loss and Optimizer

To train the proposed model a binary cross entropy loss function and an ADAM optimizer with a learning rate of $3e-4$ has been used. The number of epochs is 500 with a batch size of 15 and 6 for LaneNet and LaneNet-ConvLSTM respectively.

$$\text{binary cross entropy} = \sum_{i=1}^n y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i) \quad (1)$$

where y_i is the ground truth value and \hat{y}_i the predicted one.

4 Results

4.1 Metrics

Lane detection is an imbalanced binary classification problem, where the amount of ones, which represents lanes, is much less than the amount of zeros, which represents the background. In general, ones are just 4% of the dataset, this means that classifying all pixels as no-lane gives an accuracy of 96%. Therefore, accuracy is just a reference index, and it should not be used as key performance indicator. Therefore precision, recall and false positive rate are metrics which can indicate with higher precision the performance of the system:

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall \text{ or true positive rate} = \frac{TP}{TP + FN} \quad (3)$$

$$false \text{ positive rate} = \frac{FP}{FP + TN} \quad (4)$$

$$F1 = 2 \frac{precision \cdot recall}{precision + recall} \quad (5)$$

where TP stands for true positive, TN for true negative, FP for false positive and FN for false negative. Evaluating these metrics under different thresholds, ROC (receiver operating characteristic curve) and PRC (precision recall curve) can be generated. These curves offer the advantage of analyzing the complete classifier behavior instead of selecting an arbitrary threshold, and therefore give a better idea of the overall network performance.

4.2 Performance

The performance of the networks have been analyzed using two different datasets. The first one containing the whole TuSimple validation dataset, so mostly ideal conditions (without occlusions, shadows, etc.). The second one containing exclusively the subset of the original TuSimple validation dataset with challenging situations, adding a further step of complexity to the inference.

Performance Under General Situations: in Table 2, the results for the two different architectures in terms of area under receiver operating characteristic curve, area under precision recall curve, maximum accuracy and its associate F1 are presented. These metrics are the average of five training runs to be able to make better statistical conclusions.

Table 2. Area under ROC (receiver operating characteristic curve), under PRC (precision recall curve), maximum accuracy and its corresponding F1 for the compared state-of-the-art networks.

Network	AUROC	AUPRC	Accuracy	F1
LaneNet	99.276	87.616	98.291	78.832
LaneNet-ConvLSTM	99.274	87.685	98.300	78.965

Although, it seems that LaneNet-ConvLSTM outperforms LaneNet in most of the metrics, their performance is quite similar and applying the Student’s t-test with 5% significance level reveals that there is no statistical difference between both (the p-value for the null hypothesis $H_0 : \mu_1 = \mu_2$ is 0.40517).

Performance Under Challenging Situations: one important point of this analysis is that the TuSimple dataset has been taken mostly under favorable conditions, with no shadows, no lane occlusions, etc. On the other hand, the discussed methods should outperform single-frame detection algorithms specially under challenging situations, therefore the proposed metrics have been recalculated just for these situations (to do so a manual classification of the validation set has been done). The results of this analysis are presented in Table 3 and show that multi-frame detection algorithms clearly outperform single-frame ones under challenging situations.

Table 3. Area under ROC (receiver operating characteristic curve), under PRC (precision recall curve), maximum accuracy and its corresponding F1 for the compared state-of-the-art networks for only the challenging situations contained in the TuSimple validation set.

Network	AUC-ROC	AUC-PRC	Accuracy	F1
LaneNet	96.487	67.234	96.724	57.437
LaneNet-ConvLSTM	97.282	70.445	96.877	60.943

Semantic segmentation architectures should work robustly at a coarse level, identifying the total number of lanes correctly, and at a fine level, detecting solid and robust lanes with a high overlapping with the ground truth. With these two requirements in mind, the experimental results show visually that, in general, multi-frame architectures outperform single-frame ones under adverse situations. Figure 3 shows three examples of the predictions of the above mentioned architectures (LaneNet and LaneNet-ConvLSTM) under challenging situations.

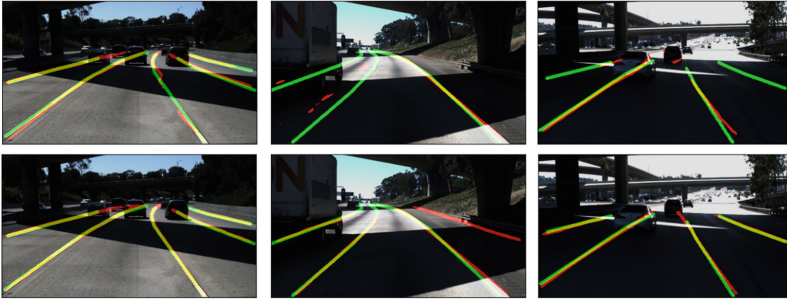


Fig. 3. Three examples of images in challenging situations. Top row: LaneNet. Bottom row: LaneNet-ConvLSTM. Green indicates ground truth and red indicates inference.

Multi-frame architectures show a more robust behavior, inferring more solid and robust lanes specially when the contrast is changing rapidly (e.g. entrance of a tunnel). Under these challenging situations, they also show higher capabilities in detecting the lines of the neighboring lanes. The proposed LaneNet-ConvLSTM architecture is also even able to correctly infer complex side lines, which are not in the ground truth, decreasing erroneously the quantitative performance.

5 Conclusions

One key conclusion is that most of the well established datasets in the field of autonomous driving just consider “ideal” conditions, and although two different architectures could show similar performance under those, the long tail influence has to definitely be taken into account. In the analyzed case, the performance of single-frame networks drops more abruptly than multi-frame architectures under challenging situations, leading to the conclusion that temporal information helps under these adverse situations and validating the initial hypothesis.

In future works, the ConvLSTM can be replaced by other types of architectures like transformers [3], which have outperformed RNN in recent years. Another important aspect is that in this work we have focused on 2D segmentation, but this information has to be transformed to 3D in order to be used for motion planning. The use of temporal information for this conversion might be also investigated in further projects.

References

1. https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection. Accessed 15 Mar 2022
2. Shi, X., et al.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. arXiv preprint [arXiv:1506.04214](https://arxiv.org/abs/1506.04214) (2015)

3. Vaswani, A., et al.: Attention Is All You Need. arXiv preprint [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) (2017)
4. Wang, W., et al.: A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Trans. Vehicul. Technol.* **67**(10), 9145–9157 (2018). <https://doi.org/10.1109/TVT.2018.2854406>
5. Wang, Z., Ren, W., Qiu, Q.: LaneNet: Real-Time Lane Detection Networks for Autonomous Driving. arXiv preprint [arXiv:1807.01726](https://arxiv.org/abs/1807.01726) (2018)
6. Zou, Q., et al.: Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks. arXiv preprint [arXiv:1903.02193](https://arxiv.org/abs/1903.02193) (2019)