EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection

Mohsen Ghafoorian(✉), Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann

TomTom, Amsterdam, The Netherlands
{mohsen.ghafoorian,cedric.nugteren,nora.baka,
olaf.booij,michael.hofmann}@tomtom.com

Abstract. Convolutional neural networks have been successfully applied to semantic segmentation problems. However, there are many problems that are inherently not pixel-wise classification problems but are nevertheless frequently formulated as semantic segmentation. This ill-posed formulation consequently necessitates hand-crafted scenario-specific and computationally expensive post-processing methods to convert the per pixel probability maps to final desired outputs. Generative adversarial networks (GANs) can be used to make the semantic segmentation network output to be more *realistic* or better *structure-preserving*, decreasing the dependency on potentially complex post-processing.

In this work, we propose EL-GAN: a GAN framework to mitigate the discussed problem using an *embedding loss*. With EL-GAN, we discriminate based on learned embeddings of both the labels and the prediction at the same time. This results in much more stable training due to having better discriminative information, benefiting from seeing both 'fake' and 'real' predictions at the same time. This substantially stabilizes the adversarial training process. We use the TuSimple lane marking challenge to demonstrate that with our proposed framework it is viable to overcome the inherent anomalies of posing it as a semantic segmentation problem. Not only is the output considerably more similar to the labels when compared to conventional methods, the subsequent post-processing is also simpler and crosses the competitive 96% accuracy threshold.

1 Introduction

Convolutional neural networks (CNNs) have been successfully applied to various computer vision problems by posing them as an image segmentation problem. Examples include road scene understanding for autonomous driving [18,20,23] and medical imaging [2,3,8,13,19,22]. The output of such a network is an image-sized map, representing per-pixel class probabilities. However, in many cases the

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-11009-3_15) contains supplementary material, which is available to authorized users.

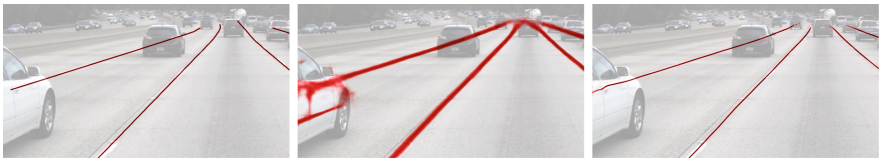
L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11129, pp. 256–272, 2019.
https://doi.org/10.1007/978-3-030-11009-3_15

problem itself is not directly a pixel-classification task, and/or the predictions need to preserve certain qualities/structures that are not enforced with the high degrees of freedom of a per-pixel classification scheme. For instance, if the task at hand is to detect a single straight line in an image, a pixel-level loss cannot easily enforce high-level qualities such as thinness, straightness or the uniqueness of the detected line. The fundamental reason behind this is the way the training loss is formulated (e.g. per-pixel cross entropy), such that each output pixel in the segmentation map is evaluated independently of all others, i.e. no explicit inter-pixel consistency is enforced. Enforcing these qualities often necessitates additional post-processing steps. Examples of post-processing steps include applying a conditional random field (CRF) [14], additional separately trained networks [20], or non-learned problem-specific algorithms [1]. Drawbacks of such approaches are that they require effort to construct, can have many hyper-parameters, are problem specific, and might still not capture the final objective. For example, CRFs need to be trained separately and either only capture local consistencies or are computationally expensive at inference time with long-range dependencies.

A potential solution for the lack of structure enforcement in semantic segmentation problems is to use generative adversarial networks (GANs) [5] to 'learn' an extra loss function that aims to model the desired properties. GANs work by training two networks in an alternating fashion in a minimax game: a *generator* is trained to produce results, while a *discriminator* is trained to distinguish produced data ('fake') from ground truth labels ('real'). GANs have also been applied to semantic segmentation problems to try to address the aforementioned issues with the per-pixel loss [18]. In such a case, the generator would produce the semantic segmentation map, while the discriminator alternately observes ground truth labels and predicted segmentation maps. There are issues with this approach, as also observed by [28]: the single binary prediction of the discriminator does not provide stable and sufficient gradient feedback to properly train the networks.

In prior work, the discriminator in a GAN observes either 'real' or 'fake' data in an alternating fashion (e.g. [18]), due to its inherently unsupervised nature. However, in the case of a semantic segmentation problem, we do have access to the ground truth data corresponding to a prediction. The intuition behind our work is that by feeding both the predictions and the labels at the same time, it is possible for a discriminator to obtain much more useful feedback to steer the training of the segmentation network in the direction of more realistic labels. In other words, the discriminator can be taught to learn a supervised loss function.

In this work, we propose such an architecture for enforcing structure in semantic segmentation output. In particular, we propose EL-GAN ('Embedding loss GAN'), in which the discriminator takes as input the source data, a prediction map and a ground truth label, and is trained to minimize the difference between embeddings of the predictions and labels. The more useful gradient feedback and increased training stability in EL-GAN enables us to successfully train semantic segmentation networks using a GAN architecture. As a result, our segmentation predictions are structurally much more similar to the training



label regular CNN EL-GAN

Fig. 1. Illustration of using EL-GAN for lane marking segmentation: an example ground truth label (left), its corresponding raw prediction by a conventional segmentation network based on [11] (middle), and a prediction by EL-GAN (right). Note how EL-GAN matches the thin-line style of the labels in terms of certainty and connectivity (Color figure online)

labels without requiring additional problem-specific loss terms or post-processing steps. The benefits of our approach are illustrated in Fig. 1, in which we show an example training label and compare it to predictions of a regular segmentation network and our EL-GAN framework. Our contributions are:

- We propose a novel method to impose structure on problems that are badly posed as semantic segmentation, by using a generative adversarial network architecture with a discriminator that is trained on both predictions and labels at the same time. We introduce EL-GAN, an instance of the above, which uses an L_2 loss on embeddings of the segmentation network predictions and the ground truth labels.
- We show that the embedding loss substantially stabilizes training and leads to more useful gradient feedback compared to a normal adversarial loss formulation. Compared to conventional segmentation networks, this requires no extra engineered loss terms or complex post-processing, leading to better label-like prediction qualities.
- We demonstrate the usefulness of EL-GAN for autonomous driving applications, although the method is generic and can be applied to other segmentation problems as well. We test on the TuSimple lane marking detection dataset and show competitive accuracy scores, but also show that EL-GAN visually produces results more similar to the ground truth labels.

2 Related Work

Quality Preserving Semantic Segmentation. Several methods have proposed to add property-targeted loss terms [2,22] or to use pair-wise or higher-order term CRFs [14,27,31], to enforce neural networks to preserve certain qualities such as smoothness, topology and neighborhood consistency. In contrast to our work, such approaches are mostly only capable of preserving lower-level consistencies and also impose additional costs at inference time. Hand-engineering extra loss terms that target enforcing certain qualities is often challenging as

identifying the target qualities in the first place and then coming up with efficient differentiable loss terms is often not straight-forward.

Adversarial Training for Semantic Segmentation. The principal underlying idea of GANs [5] is to enable a neural network to learn a target distribution for generating samples by training it in a minimax game with a competing discriminator network. Luc et al. [18] employed adversarial training for segmentation to ensure higher-level semantic consistencies. Their work involves using a discriminator that provides feedback to the segmentation network (generator) based on differences between distributions of labels and predictions. This differs from aforementioned works in the sense that the additional loss term is being learned by the discriminator rather than having fixed hand-crafted loss terms. The same mechanism was later applied to image-to-image translation [10], medical image analysis [3,8,13,17,19,24,28,29] and other segmentation tasks [21]. In contrast to our work, this formulation of adversarial training does not use the pairing information of images and labels. Based on this, some works [7,30] suggest using a GAN in a semi-supervised fashion, with the additional assumption that the unlabeled data is coming from the same distribution as the labeled ones. Our work also stems from the same intuition that this formulation does not leverage the pairing information; we instead change the method such that the pairing information is exploited. Another related work is [28], which proposes an L_1 loss term for GAN-based medical image segmentation, but interpretations and extensive ablation studies are not provided. Our method differs in the input the discriminator receives, as well as the loss term that is used to train it. In concurrent work, Hwang et al. [9] uses adversarial training for structural matching between the ground-truth and the predicted image. In contrast to our work, [9] does not condition the discriminator on the input image, nor uses a pixel-level loss to steer the training of the segmenter network. As a consequence, the discriminator representations need to be kept low-level to ensure a segmenter that attends to low-level details. Furthermore, we provide extensive ablation studies in order to better understand, discuss and interpret the characteristics and benefits of the method.

Feature matching, as proposed in [26], also learns features to maximize the difference between the real and fake distributions. However, a difference is that Salimans et al. are matching fake/real distribution features statistics (e.g. mean) rather than matching the embeddings directly, which is not possible in unsupervised image generation.

Perceptual Loss. Several recent works [4,12,25], in particular targeting image super-resolution, are based on the idea that pixel-level objective losses are often not sufficient to ensure high-level semantics of a generated image. Therefore, they suggest to capture higher-level representations of images from the representations of a separate network at a given layer. In image super-resolution, the corresponding ground truth label for a given low-resolution image is often available. Therefore, a difference measure (e.g. L_2) between the high-level representations of the reconstructed and ground truth images is considered as an

extra loss term. Our work is inspired by this idea: similarly, we propose to use the difference between the labels and predictions in a high-level embedding space.

Lane Marking Detection. Since the evaluation of our work focuses on lane marking detection, we also discuss other related approaches for this problem, while we refer the reader to a recent survey for a broader overview [1]. An example of a successful lane marking detection approach is by Pan et al. [23]. In their work, they train a problem-specific spatial CNN and add hand-crafted post-processing. Lee et al. [15] use extra vanishing-point labels to guide the network toward a more structurally consistent lane marking detection. Another recent example is the work by Neven et al. [20], in which a regular segmentation network is used to obtain lane marking prediction maps. They then train a second network to perform a constrained perspective transformation, after which curve fitting is used to obtain the final results. We compare our work in more detail to the studies above [20,23] that are similarly conducted on the Tusimple challenge, in Sect. 6.1.

3 Method

In this section we introduce EL-GAN: adversarial training with embedding loss for semantic segmentation. This method is generic and can be applied to various segmentation problems. The detailed network architecture and training set-up is discussed in Sect. 4.

3.1 Baseline: Adversarial Training for Semantic Segmentation

Adversarial training can be used to ensure a higher level of label resembling qualities such as smoothness, preserving neighborhood consistencies, and so on. This is done by using a discriminator network that learns a loss function for these desirable properties over time rather than formulating these properties explicitly. A typical approach for benefiting from adversarial training for semantic segmentation [10,18] involves formulating a loss function for the segmentation network (generator) that consists of two terms: one term concerning low-level pixel-wise prediction/label fitness ($\mathcal{L}_{\text{fit}}$) and another (adversarial) loss term for preserving higher-level consistency qualities ($\mathcal{L}_{\text{adv}}$), conditioned on the input image:

$$\mathcal{L}_{\text{gen}}(x, y; \theta_{\text{gen}}, \theta_{\text{disc}}) = \mathcal{L}_{\text{fit}}(G(x; \theta_{\text{gen}}), y) + \lambda \mathcal{L}_{\text{adv}}(G(x; \theta_{\text{gen}}); x, \theta_{\text{disc}}), \quad (1)$$

where x and y are the input image and the corresponding label map respectively, θ_{gen} and θ_{disc} are the set of parameters for the generator and discriminator networks, $G(x; \theta)$ represents a transformation on input image x, imposed by the generator network parameterized by θ, and λ indicates the relative importance of the adversarial loss term. The loss term $\mathcal{L}_{\text{fit}}$ is often formulated with a pixel-wise categorical cross entropy loss, $\mathcal{L}_{\text{cce}}(G(x; \theta_{\text{gen}}), y)$, where $\mathcal{L}_{\text{cce}}(\hat{y}, y) = \frac{1}{wh} \sum_i^{wh} \sum_j^c y_{i,j} \ln(\hat{y}_{i,j})$ with c representing the number of target classes and w and h being the width and height of the image.

The adversarial loss term, $\mathcal{L}_{\text{adv}}$ indicates how successful the discriminator is in rejecting the (fake) dense prediction maps produced by the generator and is often formulated with a binary cross entropy loss between zero and the binary prediction of the discriminator for a generated prediction map: $\mathcal{L}_{\text{bce}}(D(G(x;\theta_{\text{gen}});\theta_{\text{disc}}),0)$, where $\mathcal{L}_{\text{bce}}(\hat{z},z) = -z\ln(\hat{z}) - (1-z)\ln(1-\hat{z})$ and D is the transformation imposed by the discriminator network.

While the generator is trained to minimize its adversarial loss term, the discriminator tries to maximize it, by minimizing its loss defined as:

$$\mathcal{L}_{\text{disc}}(x,y;\theta_{\text{gen}},\theta_{\text{disc}}) = \mathcal{L}_{\text{bce}}(D(G(x;\theta_{\text{gen}});\theta_{\text{disc}}),0) + \mathcal{L}_{\text{bce}}(D(y;\theta_{\text{disc}}),1). \quad (2)$$

By alternating between the training of the two networks, the discriminator learns the differences between the label and prediction distributions, while the generator tries to change the qualities of its predictions, similar to that of the labels, such that the two distributions are not distinguishable. In practice, it is often observed that the training of the adversarial networks tends to be more tricky and unstable compared to training normal networks. This can be attributed to the mutual training of the two networks involved in a minimax game where the training dynamics of each affect the training of the other. The discriminator gives feedback to the generator based on how plausible the generator images are. There are two important issues with the frequently used adversarial training framework for semantic segmentation:

1. The notion of plausibility and fake-ness of these prediction maps comes from the discriminator's representation of these concepts and how its weights encode these qualities; This encoding is likely to be far from perfect, resulting in gradients in directions that are likely not improving the generator.
2. The conventional adversarial loss term does not exploit the valuable piece of information on image/label pairing that is often available for many of the supervised semantic segmentation tasks.

3.2 Adversarial Training with Embedding Loss

Given the two issues above, one can leverage the image/label pairing to base the plausibility/fake-ness not only on the discriminator's understanding of these notions but also on a true plausible label map. One way to utilize this idea is to use the discriminator to take the prediction/label maps into a higher-level description and define the adversarial loss as their difference in embedding space:

$$\mathcal{L}_{\text{gen}}(x,y;\theta_{\text{gen}},\theta_{\text{disc}}) = \mathcal{L}_{\text{fit}}(G(x;\theta_{\text{gen}}),y) + \lambda\mathcal{L}_{\text{adv}}(G(x;\theta_{\text{gen}}),y;x,\theta_{\text{disc}}), \quad (3)$$

where we suggest to formulate $\mathcal{L}_{\text{adv}}(G(x;\theta_{\text{gen}}),y;x,\theta_{\text{disc}})$ with embedding loss $\mathcal{L}_{\text{emb}}(G(x;\theta_{\text{gen}}),y;x,\theta_{\text{disc}})$ defined as a distance over embeddings (e.g. L_2):

$$\mathcal{L}_{\text{emb}}(\hat{y},y;x,\theta_{\text{disc}}) = \|D_e(y;x,\theta_{\text{disc}}) - D_e(\hat{y};x,\theta_{\text{disc}})\|_2, \quad (4)$$

where $D_e(\hat{y};x,\theta)$ represents the embeddings extracted from a given layer in the network D parameterized with θ, given the prediction $\hat{y}$ and x as its inputs.

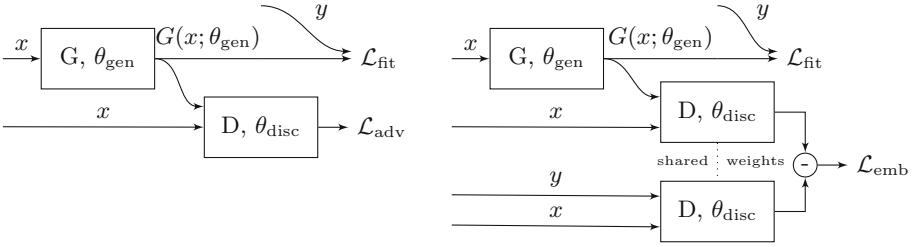


Fig. 2. Illustration of the novel training set-up for the generator loss: left for a conventional GAN (Eq. 1), right when using the embedding loss (Eqs. 3 and 4)

We name this the EL-GAN architecture, in which the adversarial loss and the corresponding gradients are computed based on a difference in high-level descriptions (embeddings) of labels and predictions. While the discriminator learns to minimize its loss on the discrimination between real and fake distributions, and likely learns a set of discriminative embeddings, the generator tries to minimize this embedding difference. This formulation of generator training is illustrated in Fig. 2 on the right-hand side, in which we also show the regular generator training set-up on the left-hand side for comparison.

Apart from the mentioned change in computing the adversarial loss for the generator updates, Eq. 2 for discriminator updates can optionally be rewritten with a similar idea as:

$$\mathcal{L}_{\text{disc}}(x, y; \theta_{\text{gen}}, \theta_{\text{disc}}) = -\mathcal{L}_{\text{emb}}(G(x, \theta_{\text{gen}}), y; x, \theta_{\text{disc}}). \tag{5}$$

However, in our empirical studies we have found that using the cross entropy loss for updating the discriminator parameters gives better results.

4 Experimental Setup

In this section we elaborate on the datasets and metrics used for evaluating our method, followed by details of the network architectures and training methods.

4.1 Evaluation Datasets and Metrics

We focus our evaluation on the application domain of autonomous driving, but stress that our method is generic and can be applied to other semantic segmentation problems as well. One of the motivations of this work is to be able to produce predictions resembling the ground truth labels as much as possible. This is in particular useful for the TuSimple lane marking detection data set with thin structures, reducing the need for complicated post-processing.

The TuSimple lane marking detection dataset[1] consists of 3626 annotated 1280×720 front-facing road images images on US highways in the San Diego

[1] TuSimple dataset details: http://benchmark.tusimple.ai/#/t/1.

area divided over four sequences, and a similar set of 2782 test images. The annotations are given in the form of polylines of lane markings: those of the ego-lane and the lanes to the left and right of the car. The polylines are given at fixed height-intervals every 20 pixels. To generate labels for semantic segmentation, we convert these to segmentation maps by discretizing the lines using smooth interpolation with a Gaussian with a sigma of 1 pixel wide. An example of such a label is shown in red in the left of Fig. 1.

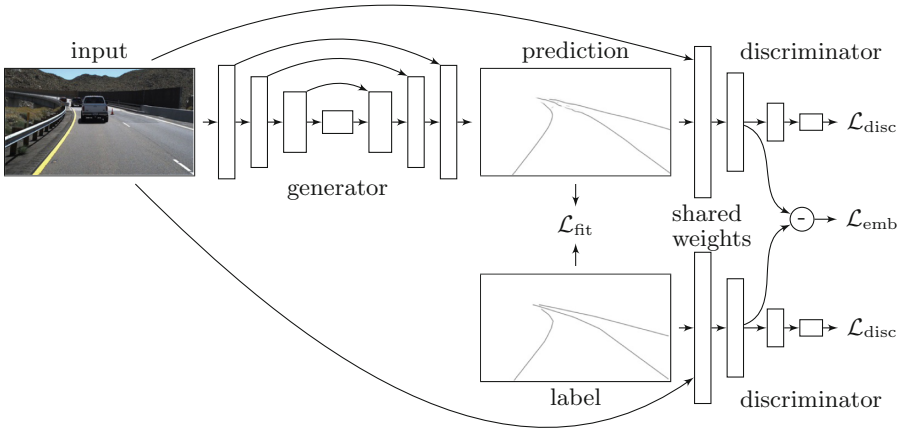


Fig. 3. Overview of the EL-GAN architecture, illustrating both the training of the generator and discriminator with examples from the TuSimple lane marking challenge

The dataset is evaluated on results in the same format as the labels, namely multiple polylines. For our evaluation we use the official metrics as defined in the challenge (See footnote 1), namely accuracy, false positive rate, and false negative rate. We report results on the official test set as well as on a validation set which is one of the labeled sequences with 409 images ('0601'). We note that performance on this validation set is perhaps not fully representative, because of its small size. A different validation sequence also has its drawbacks, since the other three are much larger and will considerably reduce the size of the already small data set.

Since our network still outputs segmentation maps rather than the required polylines, we do apply post-processing, but keep it as simple as possible: after binarizing, we transform each connected component into a separate polyline by taking the mean x-index of a sequence of non-zero values at each y-index. We refer to this method as 'basic'. We also evaluate a 'basic++' version which also splits connected components in case it detects that multiple sequences of non-zero values occur at one sampling location.

4.2 Network Architectures and Training

In this section we discuss the network and training set-up used for our experiments. A sketch of the high-level network architecture with example data is shown in Fig. 3, which shows the different loss terms used for either the generator or discriminator training, or both.

For the generator we use a fully-convolutional U-Net style network with a downwards and an upwards path and skip connections. In particular, we use the Tiramisu DenseNet architecture [11] for lane marking detection, configured with 7 up/down levels for a total of 64 3×3 convolution layers.

For the discriminator we use a DenseNet architecture [6] with 7 blocks and a total of 32 3×3 convolution layers, followed by a fully-convolutional *patch-GAN* classifier [16]. We use a two-headed network for the first 2 dense blocks to separately process the input image from the labels or predictions, after which we concatenate the feature maps. We take the embeddings after the final convolution layer, but explore other options in Sect. 5.2.

We first pre-train the generator models until convergence, which we also use as our baseline non-GAN model for Sect. 5. Using a batch size of 8, we then pre-train the discriminator for 10k iterations, after which alternate between 300 and 200 iterations of generator and discriminator training, respectively. The generator is trained with the Adam optimizer, while the discriminator training was observed to be more stable using SGD. We train the discriminator using the regular cross entropy loss (Eq. 2), while we train the generator with the adversarial embedding loss with $\lambda = 1$ (Eqs. 3 and 4). We did not do any data augmentation nor pre-train the model on other data.

5 Results

In this section we report the results on the TuSimple datasets using the experimental set-up as discussed in Sect. 4. Additionally, we perform three ablation studies: evaluating the training stability, exploring the options for the training losses, and varying the choice for embedding loss layer.

5.1 TuSimple Lane Marking Challenge

In this section we report the results of the TuSimple lane marking detection challenge and compare them with our baseline and the state-of-the-art.

We first evaluated EL-GAN and our baseline on the validation set using both post-processing methods. The results in Table 1 show that the basic post-processing method is not suitable for the baseline model, while the improved basic++ method performs a lot better. Still, EL-GAN outperforms the baseline, in particular with the most basic post-processing method.

Some results on the validation set are shown in Fig. 4, which compares the two methods in terms of raw prediction maps and post-processed results using the basic++ method. Clearly, EL-GAN produces considerably thinner and more label-like output with less noise, making post-processing easier in general.

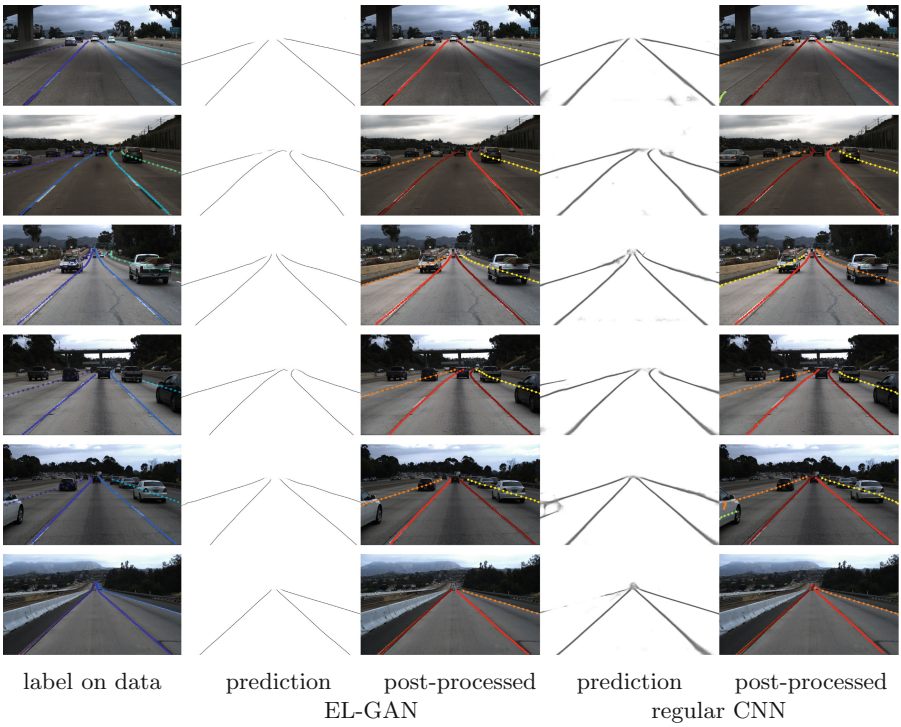


Fig. 4. Example results for lane marking segmentation: the labels on top of the data (left column), the prediction and final results of EL-GAN (next two columns), and results of the regular CNN baseline [11] using the same post-processing (right two columns). The colors of the lines have no meaning other than to distinguish them from each other. Details are best viewed on a computer screen when zoomed in (Color figure online)

Furthermore, we train EL-GAN and the baseline on the entire labeled dataset, and evaluate using the basic++ post-processing on the official test set of the TuSimple challenge. Table 2 shows the results, which includes all methods in the top 6 (only two of which are published, to the best of our knowledge) and their rank on the leaderboard as of March 14, 2018. We rank 4th based

Table 1. Results on TuSimple lane marking validation set

Method	Post-processing	Accuracy (%)	FP	FN
Baseline (no GAN)	basic	86.2	0.089	0.213
Baseline (no GAN)	basic++	94.3	0.084	0.070
EL-GAN	basic	93.3	0.061	0.104
EL-GAN	basic++	**94.9**	**0.059**	**0.067**

Table 2. TuSimple lane marking challenge leaderboard (test set) as of March 14, 2018

Rank	Method	Name on board	Extra data	Accuracy (%)	FP	FN
#1	Unpublished	leonardoli	?	**96.87**	0.0442	0.0197
#2	Pan et al. [23]	XingangPan	Yes	96.53	0.0617	**0.0180**
#3	Unpublished	aslarry	?	96.50	0.0851	0.0269
#5	Neven et al. [20]	DavyNeven	No	96.38	0.0780	0.0244
#6	Unpublished	li	?	96.15	0.1888	0.0365
#14	Baseline (no GAN)	N/A	No	94.54	0.0733	0.0476
#4	EL-GAN	TomTom EL-GAN	No	96.39	**0.0412**	0.0336

on accuracy with a difference less than half a percent to the best, and obtain the lowest false positive rate. Compared to the baseline, our adversarial training algorithm improves ∼2% on the accuracy (decrease of error by 38%), decreases the FPs by more than 55% and FNs by 30% on the private challenge test set. These improvements take the baseline from 14th rank to 4th.

5.2 Ablation Studies

Table 3 compares the use of embedding/cross entropy as different choices for adversarial loss term for training of the generator and the discriminator networks. To compare the stability of the training, statistics over validation accuracies are reported. Figure 5 furthermore illustrates the validation set F-score mean, and standard deviation over 5 training runs. These results show that using the embedding loss for the generator makes GAN training stable. We observed similar behavior when training with other hyper-parameters.

Table 3. TuSimple validation set accuracy statistics over different training iterations (every 10K), comparing the stability of different choices for adversarial losses

Loss		Accuracy statistics			Equations
Generator	Discriminator	Mean	Var	Max	
Cross entropy	Cross entropy	33.84	511.71	58.11	1 and 2
Cross entropy	Embedding	0.00	**0.00**	0.02	1 and 5
Embedding	Cross entropy	93.97	0.459	94.65	3, 4 and 2
Embedding	Embedding	**94.17**	0.429	**94.98**	3, 4 and 5

The features used for the embedding loss can be taken at different locations in the discriminator. In this section we explore three options: taking the features either after the 3rd, 5th, or 7th dense block. We note that the 3rd block contains the first shared convolution layers with both the image input and the predictions or labels, and that the 7th block contains the final set of convolutions before

Fig. 5. A comparison of training stability for using different adversarial loss terms (embedding/cross entropy) on the validation f-score. For each method the central point represents the mean f-score and the bars on each side illustrate the standard deviation. It should be noted that in the *g:emb/d:ce* and *g:emb/d:emb* cases the std bars are not visible due to tiny variations among different runs.

the classifier of the network. Results for the TuSimple lane marking detection validation set are given in Table 4 and in Fig. 6. From the results, we conclude that the later we take the embeddings, the better the score and the more similar the predictions are to the labels.

Table 4. Ablation study on embedding extraction layer

Embedding loss after block #	Accuracy (%)	FP	FN
Dense block 3 (first block after joining)	93.91	0.1013	0.1060
Dense block 5	94.01	0.0733	0.0878
Dense block 7 (before classifier)	**94.94**	**0.0592**	**0.0673**

6 Discussion

6.1 Comparison with Other Lane Marking Detection Methods

In Table 2 we showed the results on the TuSimple lane marking data set with EL-GAN ranking 4th on the leaderboard. In this section, we compare our method in more detail to the other two published methods: Pan et al. [23] (ranking 2nd) and Neven et al. [20] (ranking 5th).

Neven et al. [20] argue in their work that post-processing techniques such as curve fitting are preferably not done on the output of the network, but rather in a birds-eye perspective. To this extent they train a separate network to learn a homography to find a perspective transform for which curve fitting is easier. In our work we show that it is possible to achieve comparable accuracy results without having to perform curve fitting at all, thus omitting the requirement for training and evaluating a separate network for this purpose.

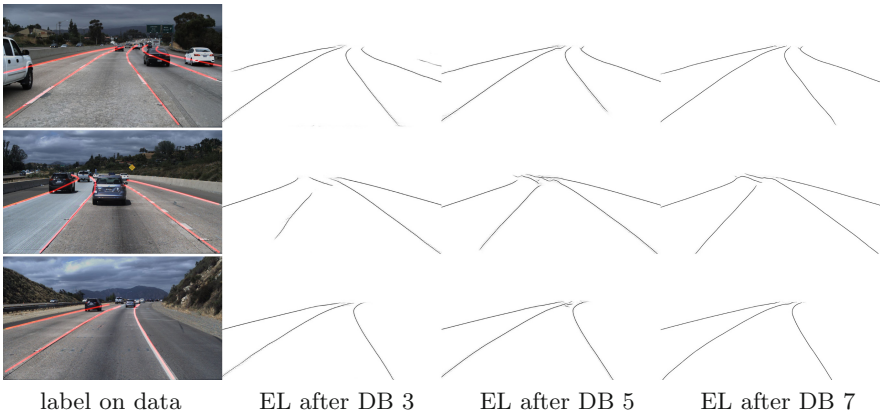


Fig. 6. Comparison of taking the embedding loss (EL) after a particular dense block (DB): the data and the label (left) and the prediction results of the different settings (right three images). Details are best viewed on a computer screen when zoomed in

Pan et al. [23] use a multi-class approach to lane marking detection, in which each lane marking is a separate class. Although this eases post-processing, it requires more complexity in label creation and makes the task more difficult for the network: it should now also learn which lane is which, requiring a larger field of view and yielding ambiguities at lane changes. In contrast, with our GAN approach, we can learn a simpler single-class problem without requiring complex post-processing to separate individual markings. Pan et al. [23] also argue that problems such as lane marking detection can benefit from spatial consistency and message passing before the final predictions are made. For this reason they propose to feed the output of a regular segmentation network into a problem specific 'spatial CNN' with message passing convolutions in different directions. This does indeed result in a better accuracy on the TuSimple data set compared to EL-GAN, however, it is unclear how much is attributed to their spatial CNN and how much to the fact that they train on a non-public data set which is 20 times larger than the regular TuSimple data set.

6.2 Analysis of the Ablation Study

As we observed in the comparison of the different adversarial loss terms as presented in Table 3 and Fig. 5, using the embedding loss for the generator makes the training more stable and prevents collapses. The embedding loss, in contrast to the usual formulation with the cross entropy loss, provides stronger signals as it leverages the existing ground-truth rather than basing it only on the discriminator's internal representations of fake-ness and plausibility.

Therefore, using a normal cross entropy loss can result in collapses, in which the generator starts to explore samples in the feature space where the discriminator's fake/real comprehension is not well formed. In contrast, using the embed-

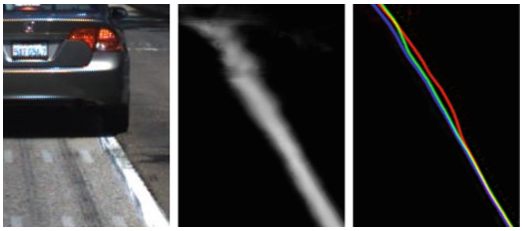


Fig. 7. Example detail of input data (left), a regular semantic segmentation output (center), and three different EL-GAN models trained with the same settings shown as red, green, and blue channels (right) (Color figure online)

ding loss, such noise productions result in high differences in the embedding space and is strictly penalized by the embedding loss. Furthermore, having an overwhelming discriminator that can perfectly distinguish the fake and real distributions results in training collapses and instability. Hence, using an embedding loss with better gradients that flow back to the generator likely results in a more competent generator. Similarly, it is no surprise that using an embedding loss for the discriminator and not for the generator results in a badly diverging behavior due to a much more dominating discriminator and a generator that is not penalized much for producing noise.

In the second ablation study, as presented in Table 4 and Fig. 6, we observed that using deeper representations for extracting the embeddings results in better performance. This is perhaps due to a larger receptive field of the embeddings that better enables the generator to improve on the higher-level qualities and consistencies.

6.3 GANs for Semantic Segmentation

Looking more closely at the comparison between a regular CNN and EL-GAN (Fig. 4), we see a distinct difference in the nature of their output. The non-GAN network produces a probabilistic output with a probability per class per pixel, while EL-GAN's output is similar to a possible label, without expressing any uncertainty. One might argue that the lack of being able to express uncertainty hinders further post-processing. However, the first step of commonly applied post-processing schemes is removing the probabilities by thresholding or applying argmax (e.g. [20,23]). In addition, the independent per-pixel probabilistic output of the regular CNN might hide inter-pixel correlation necessary for correct post-processing. The cross entropy loss pushes the network to output a segmentation distribution that does not lie on the manifold of possible labels.

In EL-GAN and other GANs for semantic segmentation, networks are trained to output a sample of the distribution of possible labels conditioned on the input image. An example is shown in Fig. 7, from which we clearly see the selection of a

sample once the lane marking is occluded and the network becomes more uncertain. Although this sacrifices the possibility to express uncertainty, we argue that the fact that it lies on, or close to, the manifold of possible labels, it can make post-processing easier and more accurate. For the task of lane marking detection we indeed have shown that the semantic segmentation does not need to output probabilities. However, for other applications this might not be the case. A straightforward approach to re-introduce expressing uncertainty by a GAN, would be to simply run it multiple times conditioned on extra random input or use an ensemble of EL-GANs. The resulting samples which model the probability on the manifold of possible labels would then be the input to post-processing.

7 Conclusions

In this paper, we proposed, studied and compared EL-GAN as a method to preserve label-resembling qualities in the predictions of the network. We showed that using EL-GAN results in a more stable adversarial training process. Furthermore, we achieved state-of-the-art results on the TuSimple challenge, without using any extra data or complicated hand-engineered post-processing pipelines, as opposed to the other competitive methods.

Acknowledgments. The authors would like to thank Nicolau Leal Werneck, Stefano Secondo, Jihong Ju, Yu Wang, Sindi Shkodrani and Bram Beernink for their contributions and valuable feedback.

References

1. Bar Hillel, A., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. Mach. Vis. Appl. **25**(3), 727–745 (2014). https://doi.org/10.1007/s00138-011-0404-2
2. BenTaieb, A., Hamarneh, G.: Topology aware fully convolutional networks for histology gland segmentation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 460–468. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_53
3. Dai, W., et al.: Scan: structure correcting adversarial network for chest x-rays organ segmentation. arXiv preprint arXiv:1703.08770 (2017)
4. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: NIPS: Advances in Neural Information Processing Systems, pp. 658–666 (2016)
5. Goodfellow, I., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014). http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
6. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR: Computer Vision and Pattern Recognition (2017)
7. Hung, W.C., Tsai, Y.H., Liou, Y.T., Lin, Y.Y., Yang, M.H.: Adversarial learning for semi-supervised semantic segmentation. arXiv e-prints, February 2018

8. Huo, Y., et al.: Splenomegaly segmentation using global convolutional kernels and conditional generative adversarial networks. In: Proceedings of SPIE, vol. 10574, pp. 10574–10574-7 (2018). https://doi.org/10.1117/12.2293406
9. Hwang, J.J., Ke, T.W., Shi, J., Yu, S.X.: Adversarial structure matching loss for image segmentation. arXiv preprint arXiv:1805.07457 (2018)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR: Computer Vision and Pattern Recognition (2017)
11. Jegou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y.: The one hundred layers tiramisu: fully convolutional DenseNets for semantic segmentation. In: CVPRW: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1175–1183, July 2017. https://doi.org/10.1109/CVPRW.2017.156
12. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
13. Kohl, S., et al.: Adversarial networks for the detection of aggressive prostate cancer. arXiv preprint arXiv:1702.08014 (2017)
14. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in Neural Information Processing Systems, pp. 109–117 (2011)
15. Lee, S., et al.: VPGNet: vanishing point guided network for lane and road marking detection and recognition. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1965–1973. IEEE (2017)
16. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 702–716. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_43
17. Li, Z., Wang, Y., Yu, J.: Brain tumor segmentation using an adversarial network. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 123–132. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9_11
18. Luc, P., Couprie, C., Chintala, S., Verbeek, J.: Semantic segmentation using adversarial networks. In: NIPS Workshop on Adversarial Training, Barcelona, Spain, December 2016. https://hal.inria.fr/hal-01398049
19. Moeskops, P., Veta, M., Lafarge, M.W., Eppenhof, K.A.J., Pluim, J.P.W.: Adversarial training and dilated convolutions for brain MRI segmentation. In: Cardoso, M.J., et al. (eds.) DLMIA/ML-CDS -2017. LNCS, vol. 10553, pp. 56–64. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67558-9_7
20. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. arXiv e-prints, February 2018
21. Nguyen, V., Vicente, T.F.Y., Zhao, M., Hoai, M., Samaras, D.: Shadow detection with conditional generative adversarial networks. In: ICCV: IEEE International Conference on Computer Vision, pp. 4520–4528. IEEE (2017)
22. Oktay, O., et al.: Anatomically constrained neural networks (ACNN): application to cardiac image enhancement and segmentation. IEEE Trans. Med. Imaging **37**(2), 384–395 (2017)
23. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: spatial CNN for traffic scene understanding. In: AAAI Conference on Artificial Intelligence, February 2018

24. Sadanandan, S.K., Karlsson, J., Whlby, C.: Spheroid segmentation using multi-scale deep adversarial networks. In: ICCVW: IEEE International Conference on Computer Vision Workshops, pp. 36–41, October 2017. https://doi.org/10.1109/ICCVW.2017.11
25. Sajjadi, M.S., Scholkopf, B., Hirsch, M.: EnhanceNet: single image super-resolution through automated texture synthesis. In: CVPR: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4491–4500 (2017)
26. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
27. Schwing, A.G., Urtasun, R.: Fully connected deep structured networks. arXiv e-prints, March 2015
28. Xue, Y., Xu, T., Zhang, H., Long, R., Huang, X.: SegAN: adversarial network with multi-scale L_1 loss for medical image segmentation. arXiv e-prints, June 2017
29. Yang, D., et al.: Automatic liver segmentation using an adversarial image-to-image network. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10435, pp. 507–515. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66179-7_58
30. Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D.P., Chen, D.Z.: Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10435, pp. 408–416. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66179-7_47
31. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: ICCV: International Conference on Computer Vision, pp. 1529–1537 (2015)