A Novel Framework for Semantic Segmentation with Generative Adversarial Network

Xiaobin Zhu, Xinming Zhang, Xiao-Yu Zhang, Ziyu Xue, Lei Wang

PII: S1047-3203(18)30293-1
DOI: https://doi.org/10.1016/j.jvcir.2018.11.020
Reference: YJVCI 2338

To appear in: *J. Vis. Commun. Image R.*

Revised Date: 23 October 2018
Accepted Date: 13 November 2018

Please cite this article as: X. Zhu, X. Zhang, X-Y. Zhang, Z. Xue, L. Wang, A Novel Framework for Semantic Segmentation with Generative Adversarial Network, *J. Vis. Commun. Image R.* (2018), doi: https://doi.org/10.1016/j.jvcir.2018.11.020

A Novel Framework for Semantic Segmentation with Generative Adversarial Network

Xiaobin Zhu[a,*], Xinming Zhang[a], Xiao-Yu Zhang[b,*], Ziyu Xue[c], Lei Wang[c,*]

[a] *Beijing Technology and Business University*
[b] *Institute of Information Engineering, Chinese Academy of Science*
[c] *Information Technology Institute, Academy of Broadcasting Science*

Abstract

Semantic segmentation plays an important role in a series of high-level computer vision applications. In the state-of-the-art semantic segmentation methods based on fully convolutional neural networks, all label variables are predicted independently from each other, and the restricted field-of-views of the convolutional filters are difficult to capture the long-range information. In this paper, a novel post-processing method based on GAN (Generative Adversarial Network) is explored to reinforce spatial contiguity in the output label maps. With the help of fully connected layers in the discriminator, the GAN can capture the long-range information, and provide an auxiliary higher-order potential loss to the segmentation model, thus the segmentation model has the ability of correcting higher order inconsistencies. Furthermore, the optimization scheme in Wasserstein GAN (WGAN) is adopted to the training process of our model to get better performance and stability. Extensive experiments on public benchmarking database demonstrate the effectiveness of the proposed method.

Keywords: semantic segmentation, generative adversarial network (GAN), wasserstein distance, auxiliary higher-order potential loss

[*] Corresponding authors

Email addresses: brucezhucas@gmail.com (Xiaobin Zhu), izhangxm@foxmail.com (Xinming Zhang), zhangxiaoyu@iie.ac.cn (Xiao-Yu Zhang), xzy_88@126.com (Ziyu Xue), wanglei@abs.ac.cn (Lei Wang)

1. Introduction

Semantic segmentation, as one of the basic and challenging tasks in computer vision, aims at assigning labels to each pixel in an image. For its wide applications in a series of higher-level vision problems, including automatic drive, augmented reality, etc., semantic segmentation has attracted more and more research attention [1][2]. However, it is still very challenging to achieve accurate segmentation results both on localization and classification. The challenges in this task include complex background, high variation in appearance, multiple viewpoints and poses of different objects, etc.

Recently, Convolutional Neural Network(CNN) [3][4][5] based semantic segmentation approaches have achieved dramatic performance improvement, meanwhile those CNN-based segmentation approaches always suffer from higher-order inconsistencies between ground-truth masks and the ones produced by the segmentation model. Higher order inconsistencies can be partially attributed to the independent prediction process of the label variables. To this end, many post-processing approaches are developed to enhance spatial consistency in the predicted label maps. Post-processing can refine the segmentation label mask and remove obvious errors. Fully conditional random fields have been broadly used to combine class scores computed by deep classifiers with the low-level information captured by the local interactions of pixels and edges. Despite these advances in using post-processing method, the segmentation models are still limited to the use of pairwise CRF models. In [6][7][8], an adversarial training approach is proposed to train segmentation models without being limited to a specific class of higher-order potentials, but the work doesn't get attractive performance on public available database.

Another difficulty relates to the low-resolution feature maps and the existence of objects at multiple scales. To achieve numerous transformation invariances, downsampling operations are always adopted, which underpins their abilities to learn hierarchical abstractions of data [9]. As the pioneer of full CNN-based semantic segmentation work, Fully Convolutional Network (FCN)

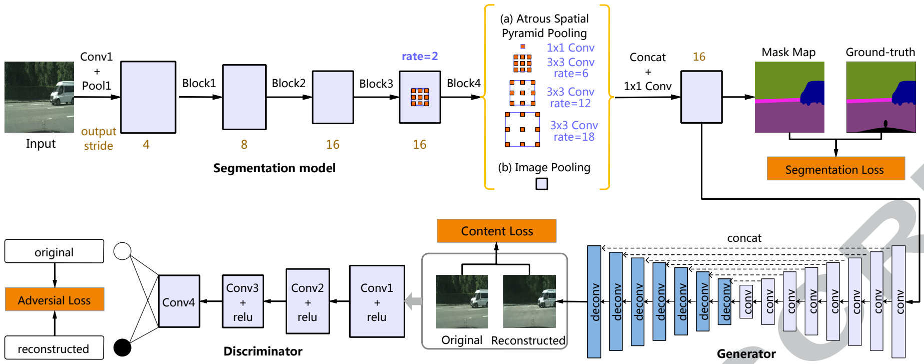

Figure 1: The framework of the proposed semantic segmentation model.

[10] upsampled feature maps at lower layers by consecutive deconvolutional operations to produce dense per-pixel labeled outputs. In [11], an encoder-decoder architecture was proposed to recover the spatial information from the low-resolution feature maps. However, fully connected layers are removed from the last layers to obtain accurate spatial information, while the global information is eliminated.

In this paper, the architecture of DeepLabv3 [12] is employed as the basic segmentation model and the architecture of Pix2pix [13] is employed as the GAN model. Two essential operations, atrous convolution and the spatial pyramid of pooling, are adopted in DeepLabv3. Atrous convolution is able to control the resolution at which feature response are computed with CNNs without requiring learning extra parameters. In order to capture the various object at multiple scales, features at different levels are concatenated by Atrous Spatial Pyramid Pooling (ASPP). Thus the challenge comes from the architecture of CNN can be addressed. Then, we attempt to jointly take advantage of the generator and the discriminator in GAN to correct the higher order inconsistencies between ground truth maps and the ones generated by the segmentation network. And we propose a composite objective function for multiple tasks. In the proposed objective function, the conventional multi-class cross-entropy loss is combined with the content loss provided by the generator and the adversarial loss extracted from the discriminator. Then, we try to optimize the segmentation model by adopting the proposed scheme. Furthermore, the optimization scheme in Wasserstein GAN (WGAN) proposed by [14] is adopted to the training process of our model. The results show that we get higher performance and stability. This paper is an extension of [15]. Compared to the previous version, this version changed GAN to WGAN for better stability and faster training. In addition, experiments were performed on several other data sets, such as StanfordBG, CMP and ADE20K. Compared to existing approaches, the contributions of our work include:

- A novel semantic segmentation framework is proposed, in which GAN is adopted to capture the long-range information and reinforce spatial contiguity in the output label maps.
- The Wasserstein distance is introduced in our GAN system to stabilize the training of GANs and to promote the performance of semantic segmentation.
- Numerous experiments are conducted on five publicly available datasets to demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces and analyzes related existing segmentation works. In Section 3, we elaborate our approach. The experimental evaluation is given in Section 4, and we draw conclusion in Section 5.

2. Related Work

Semantic segmentation is an active topic of research in the domain of computer vision tasks, and numerous methods have been proposed to solve the aforementioned basic problems. Before the arrival of deep networks, the semantic segmentation methods mostly relied on hand engineered features to classify pixels independently. Specifically, the features of a patch are fed into a elaborately designed classifier, e.g. Random Forest [16] or Boosting [17] etc., to predict the class of its center pixel. With the rapid development Convolutional

Neural Networks (CNNs) in image classification, the extracted deep features were also adopted to conduct semantic segmentation. In those pioneer works based on deep features, patchwise training is common [18, 19, 20]. And numerous pre-processing and post-processing complications, including superpixels [19, 21, 22], proposals [23, 24], or post-hoc refinement by random fields or local classifiers [19, 21] were adopted to achieve accurate boundaries prediction.

Nowadays, the model based on Fully Convolutional Networks (FCN) [10], as the milestone, has become the mainstream in the semantic segmentation task, for its elegant architectures, high efficiency and good performance. However, it also faces some inherent or even inevitable shortcomings, e.g. the low-resolution feature maps, restricted field-of-views and independent predictions, etc. The successors proposed a variety of variants to solve those problems. Directly, they adopted more powerful CNN features, VGG-16 [25], GoogLeNet [26], and ResNet [27], etc. They designed delicate structures to leverage accurate spatial information from low-level feature maps, e.g. the skip architectures [10], atrous contributions [12], etc., or tried to recover the spatial information using learned deconvolution operations [28] or upsampling operations [29][30]. As for the problems caused by the restricted field-of-views and existence of objects in multi-scales, numerous methods were proposed. In [31, 32, 33, 34], image pyramid was proposed with inputs in multiple scales. In [12], Atrous Spatial Pyramid Pooling (ASPP) was proposed, in which atrous convolutions with multiple rates were combined to extract features. Although, the aforementioned solutions led to better performance, they can only alleviate the low-resolution and multi-scales problems. They are still open issues in semantic segmentation, and there are still a long way to go in the future.

Besides, the FCN-based methods also face the problem caused by the independent prediction of each variable. In another words, in the output masks, all label variables are predicted independently from each other, which may neglect the close correlations between pixels and is difficult to capture the long-range information. All those problems cause the higher-order inconsistencies in the pixel-wise class label prediction. Various post-processing approaches have been explored to reinforce spatial contiguity in the output label maps. Conditional random fields (CRFs) are one of the most effective approaches to enforce spatial contiguity in the output label maps. In [35], a fully connected CRF, i.e. DenseCRF, was proposed to establish pairwise potentials on all pairs of pixels in the image, and a highly efficient approximate inference algorithm was proposed for fully connected CRF models. DenseCRF was broadly adopted in the semantic segmentation task as a kind of postprocessing method [12]. Furthermore, Zheng et al. [36] formulated Conditional Random Fields with Gaussian pairwise potentials and mean-field approximate inference as Recurrent Neural Networks. Those fully connected CRF models can capture the long-range information, and dig out the correlations between any nonadjacent pixels, reinforcing spatial contiguity in the output label maps. Despite these advantages, the work discussed above is limited to the use of pairwise CRF models. In [37], a convolutional semantic segmentation network along with an adversarial network was proposed to discriminate segmentation maps coming either from the ground truth or from the segmentation network. The fully-connected layers in the discriminator can exactly capture the long-range information. In [38], proposed a semi-supervised framework based on Generative Adversarial Networks (GANs) which consists of a generator to provide extra training examples, and the discriminator to assign labels to the samples. In [39], they also proposed a method for semi-supervised semantic segmentation using the adversarial network. In this paper, we still take advantage of GANs, and concentrate on reinforcing spatial contiguity, and do not involve any semi-supervised scheme.

Generative Adversarial Network (GAN) is a framework introduced by [40] to train deep generative models. It consists of a generator network, G, aiming at learning a distribution p_z of the data, and a discriminator network D, which tries to distinguish between real data (from true distribution) $p_{data}(x)$ and fake data (produced by the generator). A discriminator network can be consider as a loss function which is used to sidesteps the need to explicitly evaluate loss, in the sense that the loss function of the generative model is dened by auxiliary parameters that are not part of the generative model. Besides the conventional

GANs, we also investigate the adoption of Wasserstein GAN(WGAN) [14] in the semantic segmentation task to stabilize the training procedures and promote the segmentation performance.

3. The Proposed Framework

In the proposed framework, WGAN is adopted and constructed as the losses with auxiliary parameters to supervise the optimizing of the fundamental semantic segmentation network in an adversarial way. As shown in Figure 1. Our framework is composed of three building blocks: (1) the fundamental semantic segmentation model, specifically DeepLabv3[12], (2) the generator which is formulated to reconstruct images from generated masks, (3) the discriminator which is constructed to discriminate the reconstructed images and the real images. The generator and the discriminator formulate the GAN, and are firstly pre-trained using ground truth masks and original images. In the pretraining phase, utilizing the ground truth masks as the inputs, the generator is driven to generate reconstructed images which are hard to discriminate from the real images. Then, the pretrained generator and discriminator are adopted as the auxiliary losses to supervise the optimizing of the segmentation model. Specifically, WGAN is adopted to construct our generator and discriminator.

3.1. GAN Model

Firstly, we introduce our generator and discriminator which are formulated as a GAN model. The GAN model can be interpreted as variational losses with auxiliary parameters which are not part of the the segmentation model. Referring to Pix2pix [13], we define generator network G and discriminator network D in such a way that D can act as supervisor to G in the min-max optimization process. This process aims at the training the generator G to generate a reconstructed image I^R, while the discriminator D tries to discriminate the original image and the reconstructed image I^R. This training process of GAN can be defined as follows:

$$\min_{G}\max_{D}\mathbb{E}_{I\sim P_{data}(I)}[\log D(I)]+\mathbb{E}_{I^R\sim P_g(I)}[\log(1-D(I^R)] \tag{1}$$

The generator takes simultaneously output of predictions layer p_{seg} and original image I as input, then try to generate the same image with original image I. Highly motivated by Pix2pix [13], our G network is composed of 4 convolutional layers and 4 de-convolutional layers, and a dropout layer is followed with the rate as 0.5 to avoid overfitting. In addition, the D network is constructed by 4 convolutional layers with the ReLU method as the activation function followed every layer but excludes the last layer.

As above described, the generator is constructed as an encoder-decoder network with convolutional and deconvolutional layers. In the encoder-decoder network, the input is passed through a series of layers that progressively downsample, until a bottleneck layer, at which point the process is reversed. In order to avoid the information loss in the bottleneck, skip connections are added between each layer i and layer $n-i$, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer $n-i$.

L1 loss and L2 loss may produce blurry results on image generation problems. Despite these losses fail to encourage high-frequency crispness, in many cases they nonetheless accurately capture the low frequencies. Because the L1 loss will enforce correctness at the low frequencies, we do not need an entirely new framework to enforce correctness. In order to model high-frequencies, it is sufficient to restrict our attention to the structure in local image patches. For this end, *Patch*GAN [13] is adopted as our discriminator architecture. This discriminator tries to classify if each $N\times N$ patch in an image is real or fake. All responses of this discriminator is averaged to provide the final output of D.

Our pre-trained GAN network can be interpreted as learned higher-order losses, which sidestep the need to design higher-order loss terms explicitly since the pre-trained GAN network has access to large portions or entire output images. The higher order loss will be backed to the segmentation model, then it

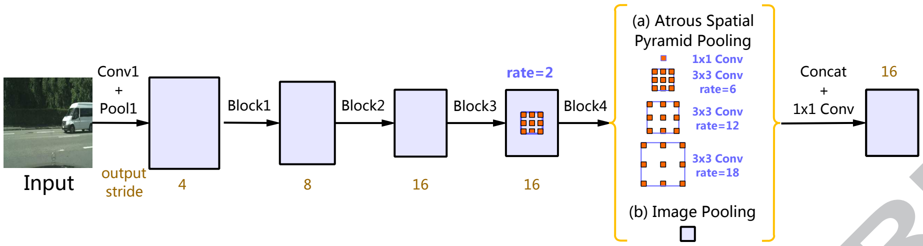

Figure 2: The model details of DeepLabv3.

has the ability of correcting higher order inconsistencies.

3.2. Segmentation Model

The goal of segmentation model is to generate confidence maps $p_{seg} \in R^{c*w*h}$, where the c is dataset class number and w, h are the width and height of prediction maps respectively. Then, the argmax operation is performed to the predictions layer to get the final prediction mask. In detail, the deeplab models are constructed based on fully convolutional neural networks, so that they can deal with different input sizes.

In our framework, DeepLabv3 [12] is selected as the basic segmentation model. The segmentation model aims to generate confidence maps $p_{seg} \in R^{c*w*h}$, where c is the dataset class numbers and w, h are the width and height of prediction maps, respectively. Then, the argmax operation is adopted to get the final prediction mask, in which each value shows the label of the response pixel of the input image.

In DeepLabv3, as shown in Figure 2, two essential operations are adopted: atrous convolution and Atrous Spatial Pyramid Pooling (ASPP). With the help of atrous convolution, one is able to control the resolution at which feature response are computed with CNNs without learning extra parameters. With ASPP, feature maps extracted by atrous convolution filters with different rates are concatenated to capture various object at multiple scales. More details of the atrous convolution and ASPP module can be found in [41].

3.3. Loss Functions

The hybrid loss function ℓ in our framework consists of three main key components: the segmentation loss ℓ_S, the content loss ℓ_C and the adversarial loss ℓ_A. In detail, the loss function ℓ can be formulated as follows:

$$\ell = \ell_S + \lambda_1 \ell_C + \lambda_2 \ell_A \tag{2}$$

where the λ_1 and λ_2 are two emprical weight parameters.

In our method, the multi-class cross-entropy loss[42] is adopted to evaluate the performance of segmentation performance. The detail segmentation loss function is defined as:

$$\ell_S = -\frac{1}{M} \sum_{j=1}^{M} \sum_{x}^{N} \sum_{i}^{C} Y_{xi}^{(j)} \log(P_{xi}^{(j)}) \tag{3}$$

where the P_{xi} is computed by the segmentation model which indicates the probability of assigning label i to pixel x, and Y_{xi} indicates the probability of the ground-truth label. M denotes the batch size, N is the pixel number of the mask, and C denotes the category number of the data set.

The content loss is used to calculate quality of the reconstructed image I^R generated by G network. The pixel-wise loss is computed as:

$$\begin{aligned} \ell_C &= \mathcal{L}_{\ell_1}(G) \\ &= \mathbb{E}_{Y,I \sim P_{data}(Y,I), z \sim P_z(z)}[\|I - G(Y,z)\|_1] \end{aligned} \tag{4}$$

The adversarial loss reflects the quality of the reconstructed image I^R reconstructed by G. Here, we test two kinds of formulation, namely the G loss of the conventional GAN, and the G loss of the Wasserteins GAN. The conventional loss is formulated as:

$$\ell_A = -\mathbb{E}_{Y \sim P_{data}(Y), z \sim P_z(z)} \log(D(G(Y,z))) \tag{5}$$

The loss formulated as WGAN is computed as:

$$\ell_A = -\mathbb{E}_{Y \sim P_{data}(Y), Z \sim P_z(z)}[D(G(Y,Z))] \tag{6}$$

The objective function of the conditional GAN is also testified in two forms, namely the conventional form:

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) =& \mathbb{E}_{Y,z\sim P_{data}(Y,z)}[\log D(Y,z)]+ \\ & \mathbb{E}_{Y\sim P_{data}(Y),I\sim P_z(I)}[\log(1-D(Y,G(Y,I))]\end{aligned} \tag{7}$$

and the WGAN form:

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) =& \mathbb{E}_{Y,z\sim P_{data}(Y,z)}[D(Y,z)]- \\ & \mathbb{E}_{Y\sim P_{data}(Y),I\sim P_z(I)}[(1-D(Y,G(Y,I))]\end{aligned} \tag{8}$$

where G tries to minimize this loss against an adversarial D that tries to maximize it, i.e. $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D)$. Previous work has found it beneficial to mix the GAN objective loss function with another traditional loss, such as L1 distance. The discriminator's goal remains unchanged, while the generator needs to not only fool the discriminator but also to be near the ground truth label mask in an L1 sense. Our final objective is

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G). \tag{9}$$

Without the noise z, the net could still learn a mapping from x to y, but would produce deterministic outputs, and therefore fail to match any distribution other than a delta function. The existing GANs have acknowledged this and provided Gaussian noise z as an input to the generator, in addition to x. Since the generator simply learned to ignore the noise, this strategy is not effective. For our final models, the form of dropout is regard as noise, applied on several layers of our generator at both training and test time. Despite the dropout noise, we observe only minor stochasticity in the output of the proposed method. Designing conditional GANs that produce highly stochastic output, and thereby capture the full entropy of the conditional distributions they model, is an important question left open by the present work.

3.4. Training Process

The framework is optimized by the loss function in Eq.2. The forward propagation is as follows. Fistly, the GAN model is trained to learn the relation ship between the ground truth and the original image. The ground-truth mask $\mathbf{I}_i^{GT}$ is the input for the generator $\mathbf{G}$ to reconstruct an image $\mathbf{I}_i^R$:

$$\mathbf{I}_i^R = \phi(\mathbf{I}_i^{GT}; \pi) \tag{10}$$

where π stands for the parameters of the generator $\mathbf{G}$. Then, the discriminator D assigns probability:

$$p = \mathbf{D}(\mathbf{I}_i, \mathbf{I}_i^R; \psi) \tag{11}$$

where $\mathbf{I}_i$ is the original image and ψ stands for the parameters of the discriminator D. In this step, the parameters of π and ψ need to learn. In particular, network parameters is initialized and forward propagation is conducted to obtain the value of each loss (ℓ_C, ℓ_A). In each iteration, we sample a mini-batch of images from the training set, and then update each parameter:

$$\pi \leftarrow \pi - \tau \nabla_\pi \left(\ell_C + \ell_A\right), \tag{12}$$

$$\psi \leftarrow \psi + \tau \nabla_\psi \ell_A \tag{13}$$

230 where τ is the learning rate. The GAN model finish the training until it converges.

After the GAN model finish its training, then we train our segmentation model under the proposed framework. First of all, the parameters of GAN model are initialized from pre-trained weights file. Then, the mask image $\mathbf{I}_i^M$ are obtained from the segmentation model:

$$\mathbf{I}_i^M = \varphi(\mathbf{I}_i; \theta) \tag{14}$$

where θ is the parameters of the segmentation model. In this step, parameters of the GAN model are fixed and will not be updated. We only have parameters of θ to learn. We use back-propagation (BP) for learning and stochastic gradient descent (SGD) to minimize the loss. In particular, we use forward propagation to obtain the value of each loss (ℓ_S, ℓ_C, ℓ_A). In each iteration, we sample a mini-batch of images from the training set, and then update the parameter θ:

$$\theta \leftarrow \theta - \tau \nabla_\theta \left(\ell_S + \ell_C + \ell_A\right) \tag{15}$$

This step is stoped until the segmentation model is converged.

4. Experiments

To evaluate the effectiveness and efficiency of the proposed method, we conduct experiments on five publicly available datasets. Firstly, datasets will be introduced briefly. Then, the experimental settings and important notations are detailed. Finally, the experimental results are discussed.

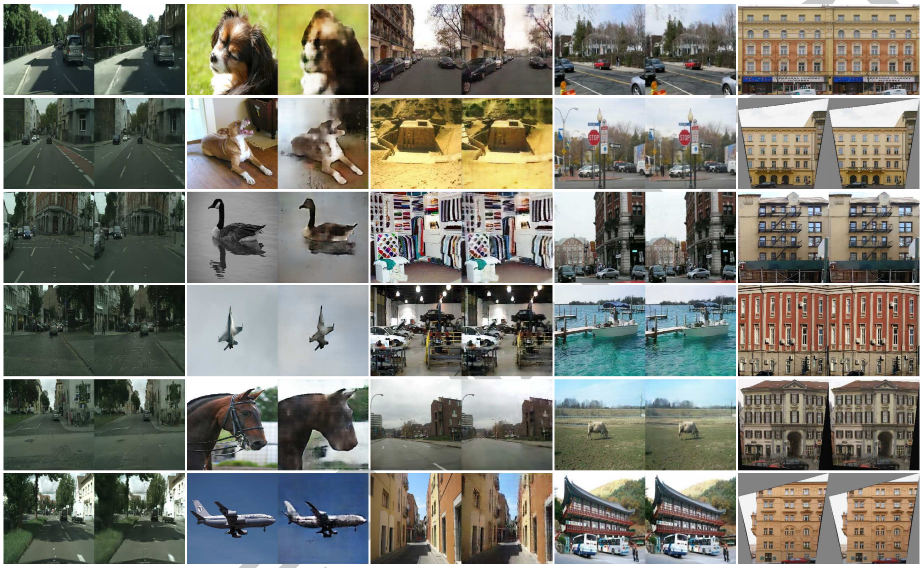

Figure 3: The samples of images reconstructed by the GAN model on the five databases. These images are divided into 5 groups according to the columns. From left to right, the five groups are Cityscapes Dataset, Pascal VOC 2012, ADE20K, StanfordBG, and CMP Facades respectively. For each group, the images on the left is the original images and the images on the right are generated from the GAN model.

4.1. Datasets

- Cityscapes [43]. Cityscapes dataset is a large-scale database which focuses on semantic understanding of urban street scenes. In Cityscapes Dataset, there are 19 foreground object classes and one background class for segmentation task, containing both stuff and objects. It contains 5000 high quality and high resolution pixel-level finely annotated images collected from 50 cites in different seasons. The dataset contains 2975, 500, and

1525 images for training, validation, and testing, respectively. It was originally recorded as video so the frames were manually selected to have the following features: large number of dynamic objects, varying scene layout, and varying background.

- PASCAL VOC 2012 [44]. PASCAL VOC 2012 consists of a ground-truth annotated dataset of images and five different competitions: classification, detection, segmentation, action classification, and person layout. The segmentation one is specially interesting since its goal is to predict the object class of each pixel for each test image. There are 21 classes categorized into vehicles, household, animals, and other. Background is also considered if the pixel does not belong to any of those classes. The dataset is divided into two subsets: training and validation with 1464 and 1449 images respectively. The testing set is private for the challengers. This dataset is arguably the most popular for semantic segmentation so most of remarkable methods in literatures have been submitted to its performance evaluation server to validate against their private test set. Methods can be trained either using only the dataset or either using additional information.

- ADE20K [45]. ADE20K has 20,210 images in the training set, 2,000 images in the validation set, and 3,000 images in the testing set. This dataset is interested in having a diverse set of scenes with dense annotations of all the objects present. Images come from the LabelMe, SUN datasets [46], and Places and were selected to cover the 900 scene categories defined in the SUN database [46].

- Stanford Background (StanfordBG) [47]. The images of Stanford Background dataset are imported from existing public datasets: LabelMe, MSRC, PASCAL VOC and Geometric Context. This dataset contains 715 images (with size of 320 x 240 pixels) with at least one foreground object and having the horizon position within the image. The dataset is pixel-wise annotated (horizon location, pixel semantic class, pixel geometric class and image region) for evaluating methods for semantic scene understanding.

- CMP Facades [48]. CMP Facades includes 606 rectified images of facades from various sources, which have been manually annotated. The facades are from different cities around the world and diverse architectural styles. The dataset has 12 classes: facade, molding, cornice, pillar, window, door, sill, blind, balcony, shop, deco, background and the labelId's range is from 1 to 12. For convenient reason, we obtain the final label mask by performing minus 1 operation to the original label mask. So that the labelId's range is from 0 to 11.

4.2. Experimental Settings and Metrics

All the networks are implemented based on the TensorFlow framework [49]. The proposed framework is trained in two steps: Firstly, the generative adversarial network is trained to learn the distribution of the ground truth masks. Limited to the GPU memory, the batch size is set to 8. As for the training of the generator, the Adam optimizer is adopted with isotropic Gaussian weights. The loss function applied to this optimization process can be formulated as:

$$\ell_G = 100 * \ell_C + \ell_A \tag{16}$$

We use fix-size input image by preprocessing operations during training, such as the random crop, the horizontal flip or resize. The procedure of GAN model training will be finished after a large number of iterations. This training protocol discussed above is named "GAN". If the GAN model is combined with our segmentation model, it is name "SegGAN". The optimization scheme proposed by [14], named "WGAN", is adopted to the training process of the proposed model. Compared with the "GAN", the difference is training protocol using in GAN model. If the WGAN model is combined with our segmentation model, it is name "SegWGAN".

In terms of the training segmentation model on Cityscapes Dataset, the pre-trained parameters provided by the project of TensorFlow [50] are directly adopted. The method , as shown in Table 1 and 2, marked as '†' presents its performance information on Cityscapes testing set. Here, we simply use the small batch size during training for the purpose of demonstration.

After the GAN model and the segmentation model have been fine-tuned, we combine the segmentation model, the generator model and the discriminator model as a whole framework. Note that, the weights in the layers of the segmentation network are initialized in the same way as the previous step. The standard Adam optimizer is utilized for the optimization of the semantic segmentation model, and the adversarial networks are initialized using pre-trained weights from the first step. As for the training settings on other four databases, we follow the above training setting.

As for the loss function formulated in formula (2), the λ_1 and λ_2 are both set as 0.1. Then, the optimization of the segmentation model is conducted based on the adopted loss function. Due to the limited GPU memory at hand, we fine-tune from the provided checkpoints whose batch norm parameters have been trained, and use smaller learning rate 0.0005 with fine_tune_batch_norm is set as False. Note that, the parameters of the GAN are not update. Similar as the training process of the generator, the proposed framework is trained util it converges. In the testing phase, we only adopted the semantic segmentation network to conduct mask prediction. The train/test scheme discussed above is applied to the training and testing process on other datasets.

The performance of different models are measured by the mean Intersection over Union (IoU) proposed in [10], and it is calculated as:

$$S_{mIoU} = (1/n_{\mathrm{cl}}) \sum_i n_{ii} / \left(\sum_j n_{ij} + \sum_j n_{ji} - n_{ii} \right) \tag{17}$$

where the n_{ii} is the number of pixels of class i predicted to class i, n_{cl} is the number of the dataset classes. Since some methods have no mIoU data, we use the pixel accuracy and class accuracy defined in [10] to measure the performance.

The pixel accuracy can be formulated as:

$$S_{pix\_acc} = \sum_i n_{ii} / \sum_i t_i \tag{18}$$

and the class accuracy can be computed as:

$$S_{cls\_acc} = (1/n_{\mathrm{cl}}) \sum_i n_{ii}/t_i \tag{19}$$

There are some different in Cityscapes Dataset. The official provides some extra indexes including IoU_{cla}, IoU_{cat}, $iIoU_{cla}$ and $iIoU_{cat}$. In order to clearly understand the results on Cityscapes Dataset, some symbols should be explained. Let C be the number of dataset classes, and let G be the number of data set categories. Every category contains one or more classes. Note that the categories are defined by the official of Cityscapes dataset. The IoU_{cla} means intersection-over-union on class, it is calculated as:

$$IoU_{cla} = \frac{1}{C} \sum_i^C \frac{TP_i}{TP_i + FP_i + FN_i} \tag{20}$$

where TP_i, FP_i and FN_i are the numbers of true positive, false positive, and false negative pixels of class i respectively. For the same reason, the IoU_{cat} means intersection-over-union on category, it is calculated as:

$$IoU_{cat} = \frac{1}{G} \sum_i^G \frac{TP_i}{TP_i + FP_i + FN_i} \tag{21}$$

where TP_i, FP_i and FN_i are the numbers of true positive, false positive, and false negative pixels of category i respectively. Besides, the official provides $iIoU_{cla}$ and $iIoU_{cat}$ score to address the problem that the global IoU measure is biased toward object instances that cover a large image area. The $iIoU_{cla}$ means instance-level intersection-over-union on class and the $iIoU_{cat}$ means instance-level intersection-over-union on category. The $iIoU_{cla}$ is defined as:

$$iIoU_{cla} = \left(\sum_k^C w_k \right)^{-1} \sum_i^C \frac{w_i * iTP_i}{iTP_i + FP_i + iFN_i} \tag{22}$$

where iTP_i, FP_i and iFN_i denote the numbers of true positive, false positive, and false negative pixels of class i respectively. The w is the ratio of the class average instance size to the size of the respective ground truth instance. And the $iIoU_{cat}$ means instance-level intersection-over-union on category, it is calculated as:

$$iIoU_{cat} = \left(\sum_{k}^{G} w_k\right)^{-1} \sum_{i}^{G} \frac{w_i * iTP_i}{iTP_i + FP_i + iFN_i} \tag{23}$$

where iTP_i, FP_i and iFN_i denote the numbers of true positive, false positive, and false negative pixels of category i, respectively.

320 ### *4.3. Experimental Results*

4.3.1. Cityscapes Dataset

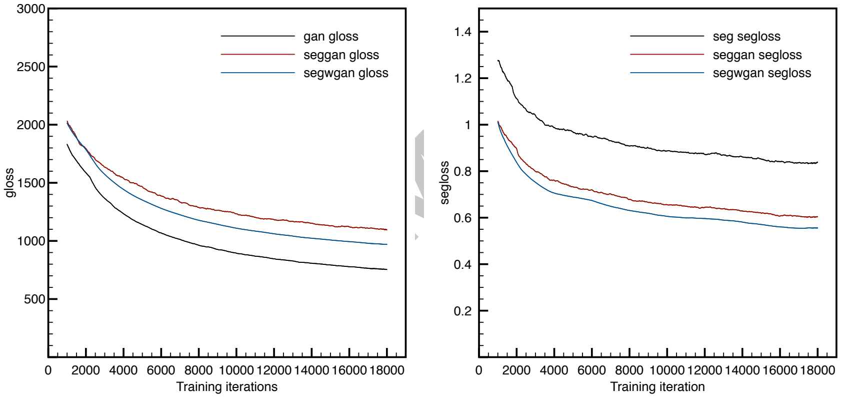

Figure 4: The gloss and segloss values across the training iterations on Cityscapes Dataset.

The IoU scores of all classes are shown in Table 1 and other performance index could be found at Table 2. By adopting GAN in DeepLabv3, the proposed method can achieve a higher mIoU score than the original DeepLabv3. It indicates that more present classes in the images are identified correctly. Note that, not all evaluation criterions of performance are promoted, such as instance-level intersection-over-union metric of category. This may caused by overfitting on some categories. Furthermore, Deeplabv3 combined with the SegWGAN not only can achieve higher scores, but also is more stability. The differences could be found in Figure 4. The figure describes the *gloss* value and the *segloss* value decreasing trends across as training iterations.

Table 1: Per-class results on Cityscapes testing set.

Method	**mIoU**	road	swalk	build	wall	fence	pole	tlight	sign	veg
CRF-RNN [36]	62.5	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6
FCN [10]	65.3	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4
DPN [51]	66.8	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5
Dilation10 [52]	67.1	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8
LRR [53]	69.7	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5
DeepLabv2 [41]	70.4	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9
Piecewise [32]	71.6	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0
TuSimple-DUC[54]	76.1	98.4	84.8	92.4	54.3	54.3	62.8	70.2	75.9	93.2
PSPNet[55]	78.4	98.6	**86.2**	**92.9**	50.8	**58.8**	64.0	**75.6**	**79.0**	93.4
DeepLabv3†	76.1	98.5	85.3	92.6	54.6	51.7	63.5	69.8	76.1	93.2
SegGAN	78.9	98.5	85.7	92.9	55.9	58.5	65.5	71.2	77.7	93.2
SegWGAN	**79.1**	**98.8**	85.1	92.8	**56.4**	58.7	**65.7**	71.4	77.1	**93.6**

Method	terrain	sky	person	rider	car	truck	bus	train	mbike	bike
CRF-RNN [36]	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6
FCN [10]	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8
DPN [51]	69.4	94.5	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8
Dilation10 [52]	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0
LRR [53]	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7
DeepLabv2 [41]	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8
Piecewise [32]	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6
TuSimple-DUC[54]	70.9	94.7	84.1	66.5	95.3	68.3	78	63.9	64.8	73.6
PSPNet[55]	72.3	95.4	**86.5**	**71.3**	**95.9**	68.2	79.5	73.8	**69.5**	**77.2**
DeepLabv3†	71.6	95.5	84.7	66.9	95.3	66.0	75.3	66.4	65.8	73.8
SegGAN	72.1	**95.5**	85.6	69.3	95.4	**72.9**	85.7	79.9	67.8	74.8
SegWGAN	**72.4**	94.0	85.6	69.5	95.2	71.1	**85.9**	**80.9**	67.8	74.8

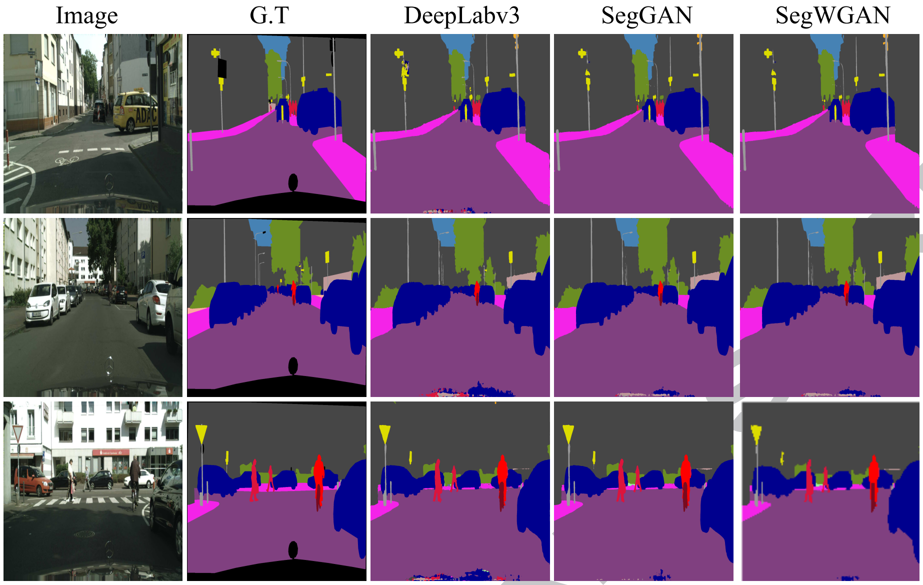

Figure 5: Visual comparisons on the Cityscapes Database.

The proposed method achieves the best result under this train/test protocol for Cityscapes Dataset. The visual comparisons produced on validation set are provided in Figure 5. The original images, the ground truth, the results of DeepLabv3, the results of the SegGAN, and the SegWGAN are shown from the first column to the five column. These images clearly indicate that our GAN model is able to learn hidden structures, and can be adopted to enhance the performance of our segmentation model. In addition, images reconstructed using the label mask on the Cityscapes Dataset during training are present in Figure 3 at the first column. The original images and the reconstructed images by the GAN model are presented in left side and right side respectively. The almost completely reconstructed images suggest that the GAN model have learned the ability of accessing higher-order potentials.

4.3.2. Pascal VOC 2012

Experiments are also conducted on PASCAL VOC 2012 [44], and the experimental results are shown in Table 3. And the training process could be found in Figure 4. The figure shows the changing trends of *gloss* and *segloss* values

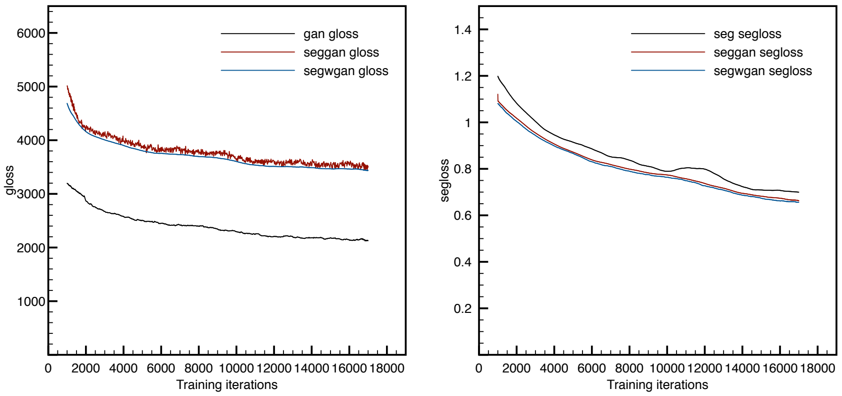

Figure 6: The gloss and segloss values across the training iterations on Pascal VOC 2012.

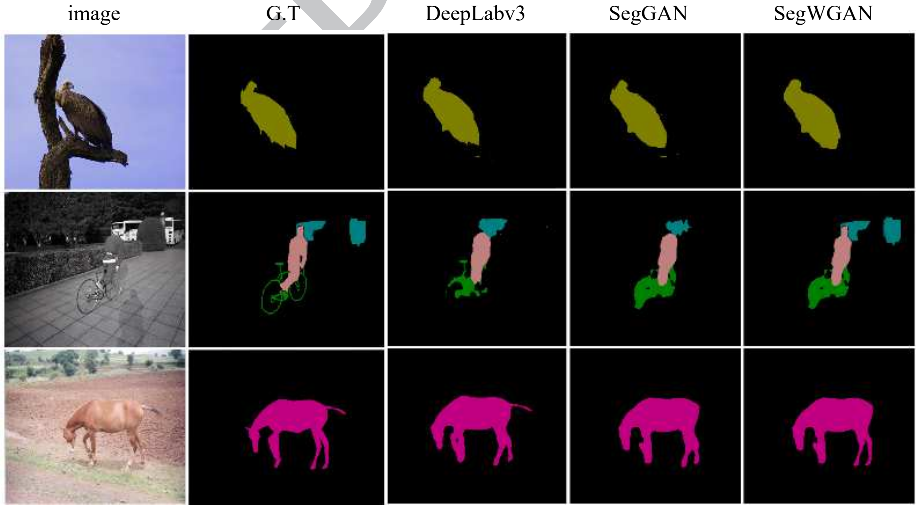

Figure 7: Samples of images produced by different segmentation models on Pascal VOC 2012.

Table 2: Results on Cityscapes testing set.

Method	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
CRF-RNN [36]	62.5	34.4	82.7	66.0
FCN [10]	65.3	41.7	85.7	70.1
DPN [51]	66.8	39.1	86.0	69.1
Dilation10 [52]	67.1	42.0	86.5	71.1
LRR [53]	69.7	48.0	88.2	74.7
DeepLabv2 [41]	70.4	42.6	86.4	67.7
Piecewise [32]	71.6	51.7	87.3	74.1
PSPNet[55]	78.4	**56.7**	**90.6**	**78.6**
DeepLabv3†	76.1	52.5	90.0	76.8
SegGAN	78.9	52.9	90.4	76.5
SegWGAN	**79.1**	53.9	90.5	76.6

across the training iterations. PASCAL VOC 2012 is a challenging database for the objects since the big difference between each other on scale, surface, and texture. By adopting GAN and WGAN based on DeepLabv3, the proposed method can achieve a higher mIoU score than the original DeepLabv3. It indicates that the GAN model can effectively detect the higher order potentials and more present classes in the images are identified correctly. It demonstrates that GAN and WGAN model could promote the performance of the segmentation model. Some visual comparisons are shown in Figure 7. In Figure 7, three different types of instances are selected for sufficient and fair comparisons. For the instance at the first line, all the compared methods can achieve satisfied results. And for the middle instance, the proposed method can detect the entire bus while the others miss some parts. However, our method cannot achieve better performance on all the images. For example, our method cannot detect the horsetail as shown at the bottom line of Figure 7. In conclusion, synthesizing the scores in Table 3 and the selected visual comparisons in Figure 7, the

Table 3: Per-class results on PASCAL VOC 2012 testing set. The reference of the method marked the '†' can be found at [55]

Method	**mIoU**	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
FCN [10]	62.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1
Zoom-out †	69.6	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3
CRF-RNN †	72.0	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1
DeconvNet [28]	72.5	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0
GCRF [56]	73.2	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1
DPN †	74.1	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0
Piecewise †	75.3	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3
DeepLabv2 [41]	79.7	92.6	**60.4**	91.6	63.4	76.3	95.0	88.4	**92.6**	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5
DeepLabv3 [12]	82.7	94.7	44.7	95.3	74.1	85.7	96.0	89.1	92.1	47.1	90.4	**74.9**	88.6	95.5	92.4	89.5	65.9	90.4	**67.6**	88.5	80.1
SegGAN	83.1	95.5	45.6	96.3	**74.8**	88.3	96.6	90.0	92.2	**46.6**	91.9	69.8	89.9	95.9	**92.5**	**89.9**	66.8	**91.9**	62.1	90.0	81.5
SegWGAN	**83.4**	**95.9**	45.9	**96.5**	74.7	**88.7**	**96.8**	**90.4**	92.5	46.0	**92.8**	69.8	**90.5**	**96.2**	92.2	89.5	**67.8**	91.7	62.7	**90.1**	**81.6**

proposed method can achieve the best performance.

4.3.3. ADE20K

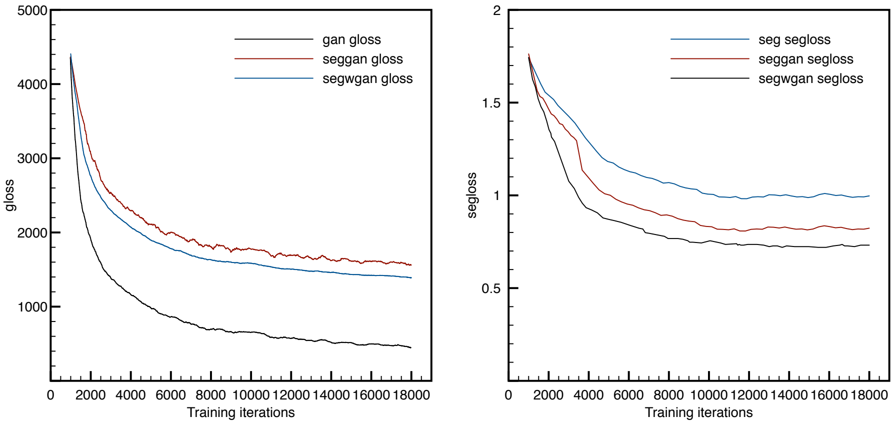

Figure 8: The gloss and segloss values across the training iterations on ADE20K dataset.

Table 4: The segmentation performance on ADE20K validation dataset.

Method	Class Acc	Pix Acc	mIoU
FCN [10]	-	71.32	29.4
SegNet [57]	-	71.00	21.6
DilatedNet[52]	-	73.55	32.3
CascadeNet [45]	-	74.52	34.9
DeepLabv3 [58]	53.9	72.5	36.0
SegGAN	54.6	73.7	36.7
SegWGAN	**54.9**	**74.1**	**36.9**

Experiments are also conducted on ADE20K [45], and the experimental results are shown in Table 4. The training process is shown in Figure 8. Comparing the original DeepLabv3, the SegWGAN is higher 0.9% on the mIoU score. Moreover, the SegGAN is higher 0.7%. Besides, it is clear that the SegWGAN get more stability on training. In the end, the SegWGAN obtains the best score on the ADE20K dataset. For example, the results of SegWGAN on the mIoU, the “Class ACC” and the “Pix Acc” are higher than the original DeepLabv3 by 1.0%, 0.6%, 0.9%. The results could be reflected by the Figure 8. In the Figure 8, the gloss SegWGAN and the segloss of SegWGAN obtain best performance simultaneously. In conclusion, all the results show the good performance and effectiveness of the proposed method.

4.3.4. StanfordBG

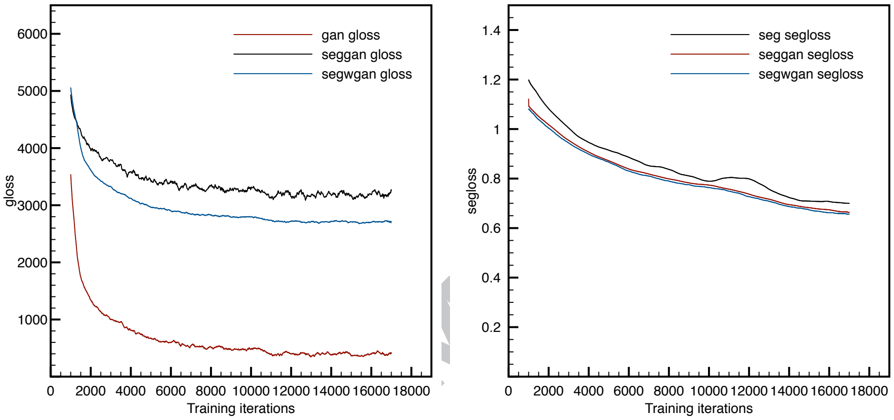

Figure 9: The gloss and segloss values across the training iterations on StanfordBG dataset.

In this section, we present the experimental results on the StanfordBG dataset, as shown in Table 5. Figure 9 describes the *gloss* value and the *segloss* value changed information as training iterations incrementing. It shows that DeepLabv3 with WGAN obtains extra stability. All the results of the compared methods (shown in the upper portion of Table 5) are directly borrowed from their papers. The symbol “-” means that they have not provided this score. In the lower portion of Table 5, different variants of the proposed method are compared. It can be clearly seen that all the variants can achieve higher scores than those compared methods. In this paper, we aims at improving the performance of the segmentation model by adopting GAN or WGAN. The experimental results definitely support our proposition. For example, the mIoU of SegGAN has been improved 0.3% than the original DeepLabv3. The “Class Acc” and “Pix

Table 5: The segmentation performance on StanfordBG dataset. The reference of the method marked the '†' can be found at [59].

Method	Class Acc	Pix Acc	mIoU
Gould et al. 2009†	-	76.4	-
Munoz et al. 2010†	66.2	76.9	-
Tighe et al. 2010†	-	77.5	-
Socher et al. 2011†	-	78.1	-
Kumar et al. 2010†	-	79.4	-
Lempitzky et al. 2011†	72.4	81.9	-
multiscale convnet†	72.4	78.8	-
INRIA et al. 2016[37]	68.7	75.2	54.3
DeepLabv2 [58]	75.9	87.0	63.4
DeepLabv3 [12]	76.5	87.4	73.2
SegGAN	76.8	87.6	73.8
SegWGAN	**79.0**	**87.7**	**73.9**

Acc" are also higher than the original DeepLabv3. The same trends are also shown in the adoption of WGAN on DeepLabv3. The mIoU of SegWGAN has been improved 2.5% than the original DeepLabv3. And SegWGAN can achieve the highest scores among all the variants and compared methods, showing the effectiveness and advancements of WGAN. In conclusion, all the results show the good performance and effectiveness of the proposed method.

4.3.5. CMP Facades

The per-class mIoU scores are shown in Table 6 and the training details are shown in Figure 10. Similar to the results of Cityscapes, the training details show that the proposed method is more stable. The performance of DeepLabv3 is promoted highly. Comparing the original DeepLabv3, the mIoU of SegWGAN is 10.2% higher than DeepLabv3. The "Class Acc" and "Pix Acc" are also higher than the original DeepLabv3 by 4.7% and 5.6%. In conclusion, the results suggest the good performance and effectiveness of the proposed method.

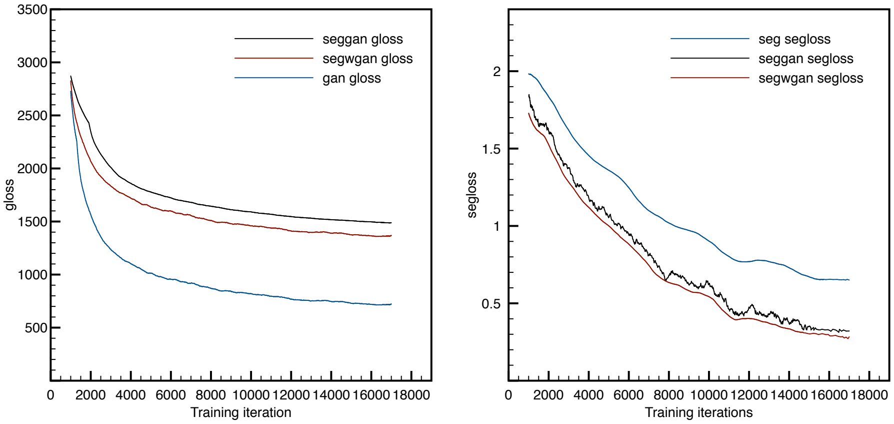

Figure 10: Training details of proposed method on the CMP Facades Dataset.

Table 6: The segmentation performance on CMP Facades validation dataset.

Method	Class Acc	Pix Acc	Mean IoU
Efficient[60]	48.9	68.0	37.4
Efficient2D[61]	48.7	68.1	37.5
DeepLabv3[58]	67.9	72.1	43.3
SegGAN	71.7	76.1	49.6
SegWGAN	72.6	77.8	53.5

5. Conclusion

In this paper, we propose a novel framework that a semantic segmentation model is combined with a GAN model to optimize the generated masks. In this work, the GAN is adopted to measure the loss of higher order inconsistencies by digging out the relationship between the masks and images, and then the learned GAN is treated as an auxiliary loss to fine-tune the semantic segmentation model. Numerous experiments on five publicly datasets show that our proposed method can steadily improve the performance of the semantic segmentation model.

Acknowledgement

This work was supported by National Key R&D Program of China (2017YFB1401000) and National Natural Science Foundation of China (61501457, 61602517). The corresponding authors are Xinming Zhang and Xiao-Yu Zhang, who contribute equally to this paper.

References

[1] L. Nie, L. Zhang, L. Meng, X. Song, X. Chang, X. Li, Modeling disease progression via multisource multitask learners: A case study with alzheimers disease, IEEE Trans. Neural Netw. Learning Syst 28 (7) (2017) 1508–1519.

[2] L. Nie, X. Wang, J. Zhang, X. He, H. Zhang, R. Hong, Q. Tian, Enhancing micro-video understanding by harnessing external sounds, in: Proceedings of the 2017 ACM on Multimedia Conference, ACM, 2017, pp. 1192–1200.

[3] X. Song, F. Feng, X. Han, X. Yang, W. Liu, L. Nie, Neural compatibility modeling with attentive knowledge distillation, arXiv preprint arXiv:1805.00313.

[4] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, J. Ma, Neurostylist: Neural compatibility modeling for clothing matching, in: Proceedings of the 2017 ACM on Multimedia Conference, ACM, 2017, pp. 753–761.

[5] J. Chen, X. Song, L. Nie, X. Wang, H. Zhang, T.-S. Chua, Micro tells macro: predicting the popularity of micro-videos via a transductive model, in: Proceedings of the 2016 ACM on Multimedia Conference, ACM, 2016, pp. 898–907.

[6] P. Luc, C. Couprie, S. Chintala, J. Verbeek, Semantic segmentation using adversarial networks, arXiv preprint arXiv:1611.08408.

[7] S. Qi, X. Wang, X. Zhang, X. Song, Z. L. Jiang, Scalable graph based non-negative multi-view embedding for image ranking, Neurocomputing 274 (2018) 29–36.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[9] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. J. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, Vol. 8689 of Lecture Notes in Computer Science, Springer, 2014, pp. 818–833.

[10] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[11] J. Wang, Z. Wang, D. Tao, S. See, G. Wang, Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V, Vol. 9909 of Lecture Notes in Computer Science, Springer, 2016, pp. 664–679.

[12] L. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, CoRR abs/1706.05587. arXiv:1706.05587.

[13] P. Isola, J. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 5967–5976.

[14] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875.

[15] X. Z. e. Xinming Zhang, A novel framework for semantic segmentation with generative adversarial network, Beijing, China, 2018.

[16] J. Shotton, M. Johnson, R. Cipolla, Semantic texton forests for image categorization and segmentation, in: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA, 2008.

[17] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, P. H. S. Torr, What, where and how many? combining object detectors and crfs, in: Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV, 2010, pp. 424–437.

[18] D. C. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Deep neural networks segment neuronal membranes in electron microscopy images, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., 2012, pp. 2852–2860.

[19] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1915–1929.

[20] P. H. O. Pinheiro, R. Collobert, Recurrent convolutional neural networks for scene labeling, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, 2014, pp. 82–90.

[21] B. Hariharan, P. A. Arbeláez, R. B. Girshick, J. Malik, Simultaneous detection and segmentation, in: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII, 2014, pp. 297–312.

[22] X. Song, X. Wang, L. Nie, X. He, Z. Chen, W. Liu, A personal privacy preserving framework: I let you know who can see what.

[23] P. Jing, Y. Su, C. Xu, L. Zhang, Hyperssr: A hypergraph based semi-supervised ranking method for visual search reranking, Neurocomputing 274 (2018) 50–57.

[24] S. Gupta, R. B. Girshick, P. A. Arbeláez, J. Malik, Learning rich features from RGB-D images for object detection and segmentation, in: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII, 2014, pp. 345–360.

[25] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.

[29] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation.

[30] Y. Liu, L. Nie, L. Liu, D. S. Rosenblum, From action to activity: sensor-based activity recognition, Neurocomputing 181 (2016) 108–115.

[31] L. Chen, Y. Yang, J. Wang, W. Xu, A. L. Yuille, Attention to scale: Scale-aware semantic image segmentation, in: 2016 IEEE Conference on Com-

puter Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 3640–3649.

[32] G. Lin, C. Shen, A. van den Hengel, I. D. Reid, Efficient piecewise training of deep structured models for semantic segmentation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 3194–3203.

[33] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, M. Wang, Low-rank multi-view embedding learning for micro-video popularity prediction, IEEE Transactions on Knowledge and Data Engineering.

[34] P. Jing, Y. Su, L. Nie, H. Gu, J. Liu, M. Wang, A framework of joint low-rank and sparse regression for image memorability prediction, IEEE Transactions on Circuits and Systems for Video Technology.

[35] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, in: Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain., 2011, pp. 109–117.

[36] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. H. S. Torr, Conditional random fields as recurrent neural networks, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015, pp. 1529–1537.

[37] P. Luc, C. Couprie, S. Chintala, J. Verbeek, Semantic Segmentation using Adversarial Networks., CoRR.

[38] N. Souly, C. Spampinato, M. Shah, Semi supervised semantic segmentation using generative adversarial network, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, 2017, pp. 5689–5697.

[39] W. Hung, Y. Tsai, Y. Liou, Y. Lin, M. Yang, Adversarial learning for semi-supervised semantic segmentation, CoRR abs/1802.07934. arXiv:1802.07934.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

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

[42] H. He, R. Xia, Joint binary neural network for multi-label learning with applications to emotion classification.

[43] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 3213–3223.

[44] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International journal of computer vision 111 (1) (2015) 98–136.

[45] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[46] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE, 2010, pp. 3485–3492.

[47] S. Gould, R. Fulton, D. Koller, Decomposing a scene into geometric and semantically consistent regions, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1–8.

[48] R. Tyleček, R. Šára, Spatial pattern templates for recognition of objects with regular structure, in: Proc. GCPR, Saarbrucken, Germany, 2013.

[49] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning., in: OSDI, Vol. 16, 2016, pp. 265–283.

[50] Tensorflow, Deeplab: Deep labelling for semantic image segmentation, https://github.com/tensorflow/models/tree/master/research/deeplab (2018).

[51] Z. Liu, X. Li, P. Luo, C. C. Loy, X. Tang, Semantic image segmentation via deep parsing network, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, IEEE Computer Society, 2015, pp. 1377–1385.

[52] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, CoRR abs/1511.07122. arXiv:1511.07122.

[53] G. Ghiasi, C. C. Fowlkes, Laplacian pyramid reconstruction and refinement for semantic segmentation, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III, Vol. 9907 of Lecture Notes in Computer Science, Springer, 2016, pp. 519–534.

[54] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. W. Cottrell, Understanding convolution for semantic segmentation, CoRR abs/1702.08502. arXiv:1702.08502.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, CoRR abs/1612.01105. arXiv:1612.01105.

[56] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, in: Advances in neural information processing systems, 2011, pp. 109–117.

[57] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, CoRR abs/1511.00561. arXiv:1511.00561.

[58] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE transactions on pattern analysis and machine intelligence 40 (4) (2018) 834–848.

[59] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1915–1929.

[60] V. Jampani, R. Gadde, P. V. Gehler, Efficient facade segmentation using auto-context, in: Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on, IEEE, 2015, pp. 1038–1045.

[61] R. Gadde, V. Jampani, R. Marlet, P. Gehler, Efficient 2d and 3d facade segmentation using auto-context, IEEE transactions on pattern analysis and machine intelligence.

Highlights:

1、A novel semantic segmentation framework is proposed, in which GAN is adopted to capture the long-range information and reinforce spatial contiguity in the output label maps.

2、The Wasserstein distance is introduced in our GAN system to stabilize the training of GANs and to promote the performance of semantic segmentation.

3、Numerous experiments are conducted on five publicly available datasets to demonstrate the effectiveness of the proposed method.