Conditional Adversarial Networks Assisted Road Extraction in Remote Sensing Imagery

Jianhua Li[a,*], Hongbing Ma[b]

Department of Electronic Engineering, Tsinghua University, Beijing, China

[a]li-jh16@mails.tsinghua.edu.cn, [b]hbma@tsinghua.edu.cn

*corresponding author

Keywords: Generative adversarial networks, road extraction, remote sensing image, machine learning.

Abstract: In this paper, we narrow down the task for conditional adversarial networks focusing on solving the binary segmentation problem for road extraction on remote sensing imagery. We constrain the objective of the generator of a conditional adversarial network to make it suitable to produce binary road predictions in an effective way. We evaluate our approach on the SpaceNet Roads dataset and our method shows promising results compared to standard segmentation models. Our best pixel level precision score on test set is 76.9%. This paper shows that adversarial networks with detailed images as input and binary mask as output can be optimized with certain adaption and optimization tweaks, showing potentials in solving computer vision problems.

1. Introduction

The extraction of roads on remote sensing imagery plays an important role in information acquisition, verification and update for Geospatial Information System (GIS) and urban road networks. Feature based road extraction methods use both road specific features and common image processing features. Road specific geometric features are used in [2] to improve road extraction accuracy. Salient features of roads are used in [3] to design a multistage framework for road extraction on high-resolution remote sensing imagery.

That being said, it is not until recent years when deep convolutional neural networks (CNN) outperform many other methods on problems concerning machine learning and computer vision that high accuracy road extraction on remote sensing imagery become possible.

Generative adversarial networks (GAN) [4] is a adversarial optimization framework designed to generate “fake” data which ideally has the same distribution as real training data. GANs train two models, a generator and a discriminator, simultaneously to optimize for a “fake” data generator. The discriminator is trained to distinguish between generated “fake” data and real data, which is designed as an adversary of the generator in the framework. In terms of images, methods [8] that conditioned the GAN with input images (conditional GAN) have achieved astonishing results on image-to-image translation problems to generate images real enough to deceive human in visual.

In this paper, we adapt conditional GANs to solving a slightly different problem where the output image is less detailed than the input one. The road extraction problem is treated as a binary segmentation problem, i.e., we want to extract the road of an image to produce a binary mask output. Based on the above idea, we study the objective of GANs, and constrain the objective of GANs to narrow down the generator's task, making it suitable to generate binary road segmentation results on remote sensing images. We show that conditional GANs with detailed input images and desired binary mask output can be optimized with certain adaption and optimization tweaks.

2. Method

Generative Adversarial Networks (GANs) consist of two models: a generator G and a discriminator D. For each sample $y \sim p_y(y)$, the generator aims at estimating the distribution of it,

  DOI: 10.25236/mmmce.2019.143

while the discriminator tries to determine whether its input is "true" or "fake", i.e., whether the input is generated by the generator or it is a real sample. GANs try to solve a minimax problem.

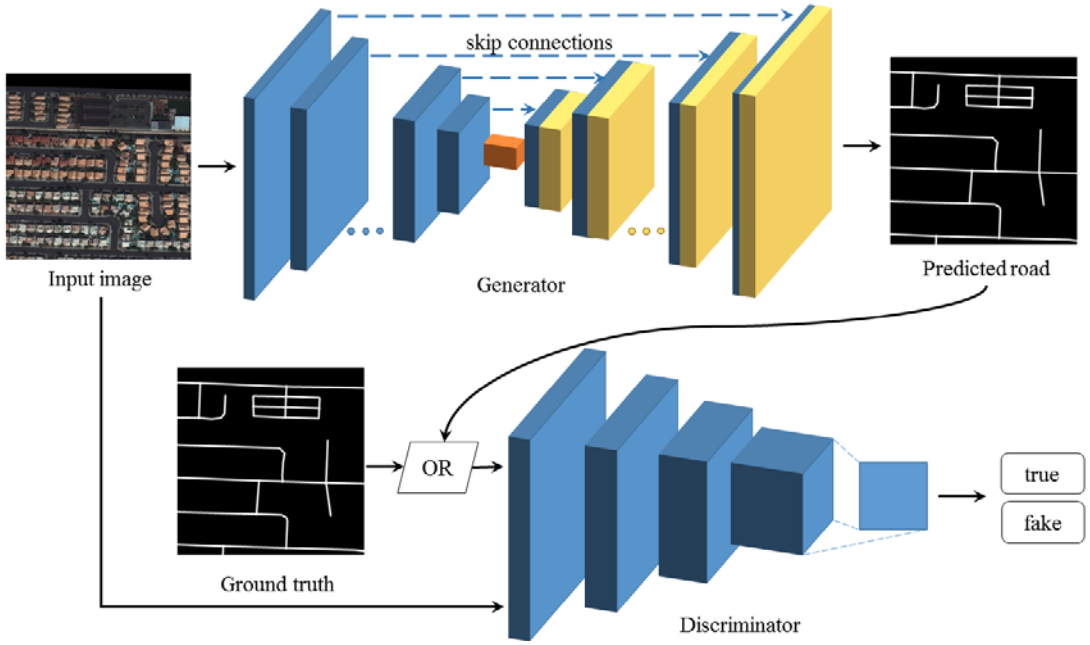


Figure 1 Proposed conditional GAN method for road binary mask extraction.

2.1. Objective

For our road extraction task, we condition the GAN generator with an input remote sensing image $x \sim p_x(x)$ to force a prediction of road binary mask, as illustrated in Figure 1. Our proposed GAN structure follows [8], where the generator is a U-Net [9] stacked up with convolutional layers and mirrored deconvolutional layers along with skip connections between each mirrored layer pair, and the discriminator is a simple convolutional neural network (CNN) with a probability output.

2.1.1. Conditional GAN Loss

For each remote sensing image and its corresponding ground truth road extraction label x, $y \sim p_{\text{data}}(x, y)$, the conditional GAN loss is expressed as:

$$\mathcal{L}_{\text{cgan}}(G, D) = \mathbb{E}[\log D(x, y)] + \mathbb{E}[\log(1 - D(x, G(x)))]. \quad (1)$$

[8] uses L_1, L_2-norm as extra constraints for the image-to-image translation task. Our task to solve the desired problem, however, is more specific. We focus on the road extraction problem, which is considered as a binary segmentation task, and therefore we introduce two commonly used binary segmentation loss functions to constrain the generator to fulfil its purpose.

2.1.2. Binary Segmentation Loss

In road binary segmentation, the ground truth labels are in {0, 1}. We introduce a binary cross entropy loss, as expressed in (2), into the objective of the conditional GAN.

$$\mathcal{L}_{\text{bce}}(G) = -\mathbb{E}[y \cdot \log G(x) + (1 - y) \cdot \log(1 - G(x))], \quad (2)$$

Where the dot represents the dot product of two images treated as vectors, i.e. the sum of products of all corresponding pixel values.

Dice similarity coefficient (DSC) loss is used to measure the similarity of the predicted image and the ground truth image as two sets of binary labels. Using the binary segmentation terms: true positive (TP), false positive (FP), true negative (TN), false negative (FN), DSC is defined as:

$$DSC = 2TP / (2TP + FP + FN),$$

Which is in its discrete form. To make it possible to be trained, we expressed DSC loss in a

continuous form:

$$\mathcal{L}_{DSC} = 1 - \mathbb{E}_{x,y\sim p_{\text{data}}(x,y)}\left[2\sum_{i=1}^{N} y_i p_i \Big/ \left(\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} p_i + \epsilon \right) \right], \quad (3)$$

Where y_i in {0, 1} denotes the ground truth label of the i-th pixel in image x containing N pixels, p_i in [0, 1] denotes the predicted probability output of the i-th pixel.

The overall binary segmentation loss that we use is a linear combination of the binary cross entropy and the DSC loss:

$$\mathcal{L}_{\text{bseg}}(G) = (1-\lambda_{\text{b}})\mathcal{L}_{\text{bce}}(G) + \lambda_{\text{b}}\mathcal{L}_{DSC}(G), \quad (4)$$

Where λ_{b} is a coefficient in the range of (0, 1). The objective of the binary segmentation task of the generator G is to minimize (4), through which the training process simultaneously minimizes the difference of the probability distribution of the predicted labels from that of the ground truth labels and maximizes the intersection-over-union metric between the predicted road segmentations and the ground truths.

Our final objective is expressed as follows:

$$\min_G \max_D \mathcal{L}(G,D) = \mathcal{L}_{\text{cgan}}(G,D) + \lambda\mathcal{L}_{\text{bseg}}(G), \quad (5)$$

Where $\lambda > 0$ is a hyperparameter to control the training balance between the conditional GAN loss and the binary segmentation loss.

2.2. Architecture

Our generator and discriminator, as illustrated in Figure 1, are adapted from those in [8]. As our generator we use U-Net [9] consisting of encoders and mirrored decoders following the encoder-decoder [1] architecture with skip connections between each pair of mirrored encoder and decoder. To produce a probability output, the discriminator is simply designed as a convolutional neural network.

2.2.1. The Generator

The input size of the generator is 256×256, with 3-channel RGB images as input. The generator consists of 8 encoders and 8 decoders. The output size of each encoder starts at 128×128 and is halved successively, producing 1×1 features eventually after the final 8th encoder. In the meantime, the number of features output for each encoder starts at 64, and is doubled successively, until reaching a ceiling of 512. Each of the decoder produces the same output size and number of features as those of its mirrored encoder (except the last decoder), and its output is concatenated with the input of its mirrored encoder by a skip connection. The output of the last decoder is 256×256×1 in size, which is then activated by a sigmoid activation and is the predicted binary mask desired.

The structure of the encoder is ReLU-convolution-BatchNorm, where in the convolutional unit the kernel size is 4×4 and the moving stride is 2, and BatchNorm is a batch normalization unit [7]. Similarly, the structure of the decoder is ReLU-deconvolution-BatchNorm-dropout, where in the deconvolutional unit the kernel size is 4×4 and the moving stride is 2, and the dropout unit drops 50% parameters.

2.2.2. The Discriminator

The input of the discriminator is 256×256 in size, and 4 in number of channels, which is concatenated by a RGB input image and a road extraction binary mask (either a ground truth or a predicted “fake” one). There are 5 convolutional layers, the output sizes of which are 128×128, 64×64, 32×32, 31×31 and 30×30 respectively, while the number of output features are 64, 128, 256, 512 and 1 respectively.

In each layer of the discriminator, the structure is convolution-BatchNorm-ReLU, where in each convolutional unit the kernel size is 4×4 and the moving stride is 2.

2.3. Optimization

To optimize our conditional GAN, we use minibatch SGD method on the generator *G* and the discriminator *D* in an alternate way as done in [8]. Several gradient ascent steps are progressed on *D* with *G* fixed, and after that several gradient descent steps are progressed on *G* with *D* fixed. Specifically, we use a step size of 1 for optimizing both *G* and *D* and a minibatch size of 16.

3. Experiments and Evaluation

In this section, we briefly describe the dataset and evaluation metrics we used for all experiments conducted, specifically give training and testing details of our proposed conditional GAN models and finally provide evaluation and comparison results for our method to U-Net as a baseline.

3.1. Dataset

We use the SpaceNet Roads [10] dataset for our experiments. The dataset contains fusion RGB images, with a spatial resolution 0.31m and an image size of 1300×1300. There are four regions of interest in the dataset, including Vegas, Paris, Shanghai and Khartoum. Both satellite images and road centerline vectors annotations are provided by the dataset.

In our experiments, we randomly crop the input images of size 1300×1300 into 256×256 to fit our designed input size for our conditional GAN. 80% of the data are separated into training dataset, and the rest into testing dataset.

The annotations of the dataset we used in this section are road centerline vectors instead of road binary masks. In order to prepare the dataset for experiments, we take a preprocessing step over the image annotations, where a simple boundary fill method with fixed width of 4m is applied on road centerlines to produce binary masks.

3.2. Optimization Details

In all experiments, we choose $\lambda_b = 0.5$ and $\lambda = 100$. As for the neural network training, we choose Adam optimizer with an initial learning rate of 2.0×10^{-4}, a momentum of 0.5 and a minibatch size of 16. We train 100–200 epochs for each model to get a best result. We use TensorFlow as the deep learning framework.

During testing, we crop a full size image into 1280×1280 to feed into the trained generator.

The conditional GAN architecture and the training procedure in our experiments are adapted from Hesse's implementation of [8]. The output of the generator is in an continuous range of [0, 1], whereas the ground truth labels are binary in {0, 1}, which makes it too easy for the discriminator to discriminate between “true” road binary masks and “fake” predictions. Therefore, we discretize the output of the generator into {0, 1} for the use as input for the discriminator.

3.3. Results

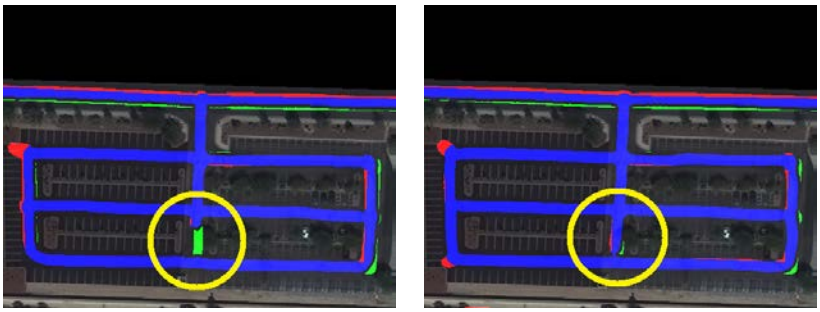
In this section, we evaluate the performance of our conditional GAN based road extraction model using pixel level metrics including intersection-over-union (IoU), precision and recall.

$$IoU = \frac{TP}{TP + FP + FN}, Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{FP + FN},$$

All three metrics are evaluated over the test dataset and then averaged.

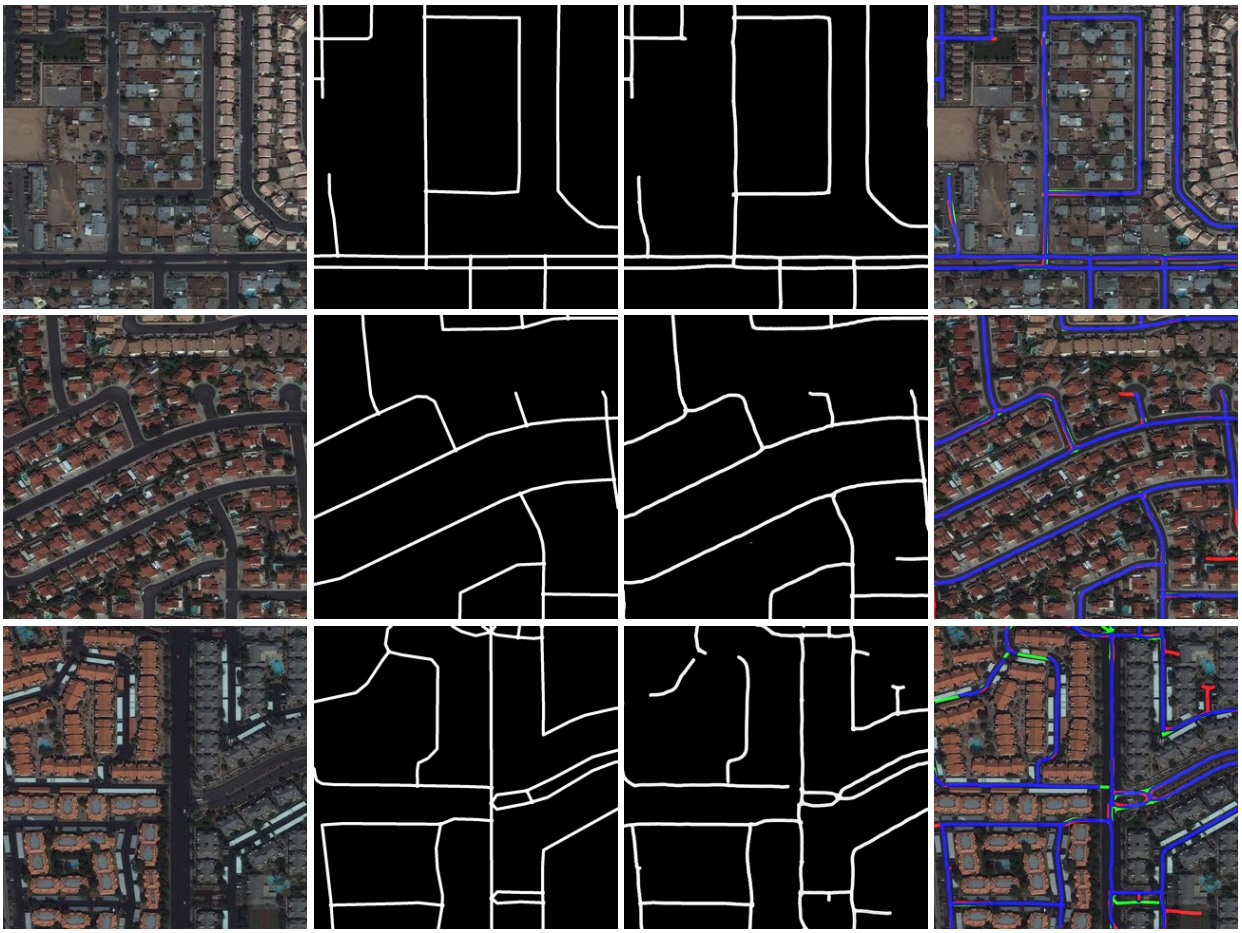
Table 1 Quantitative results of road binary segmentation metrics for different methods (%).

	mean IoU	Precision	Recall
U-Net	65.2	74.1	**81.9**
Howe et al.	89.0	-	-
Our cGAN	63.7	73.0	80.8
cGAN + postprocessing	65.6	76.9	80.1



U-Net conditional GAN

Figure 2 Detailed comparisons between road extraction using our conditional GAN and U-Net as baseline: connectivity. Main differences are shown in circle, and true positive in blue, false positive in red, false negative in green. Better view in color.



Input image Ground truth Predicted output Visual result

Figure 3 Visual results of our conditional GAN method on road extraction. On the last column: true positive in blue, false positive in red, false negative in green. Better view in color.

Table 1 lists mean IoU, precision and recall for four different methods evaluated on test set. We train a U-Net of the same structure as our conditional GAN generator independently as a baseline model. Howe et al. [6] uses a model adapted from U-Net and ResNet [5], and we list the authors' results in the table as comparison. For our cGAN method, we apply certain postprocessing steps to retrieve a slightly better result. In postprocessing, we firstly Gaussian blur the road binary mask predicted by the generator of our conditional GAN, secondly skeletonize the mask using morphology algorithms and finally apply boundary fill again to reproduce a road binary mask of width 4m. All postprocessing steps use the scikit-image library.

From Table 1, we find our cGAN method to have comparable performance to the baseline U-Net. Our method reaches a road precision of ~75% and tends to have a higher recall rate. Figure 2 shows

comparison of our method to U-Net with two sample images. From this figure, we find that the connectivity of our method can be better than that of U-Net.

Visual results of our method is shown in Figure 3, where the sample images are taken from the test set. This figure shows various extraction results: good ones, ones with false positives and ones with false negatives. Meanwhile, there are some defects of our method shown on the sample images. It is still difficult to obtain accurate extraction result where there are shadows of plants or buildings, as illustrated on the 2nd and the last row in Figure 3.

4. Conclusion

The road extraction method proposed in this paper, which adapts a conditional GAN in its objective, is able to generate comparable results on remote sensing imagery. The experiments show promising performances both in evaluation metrics and in visualization, especially in the recall of road pixels. Conditional GANs have a great potential in solving vision problems.

References

[1] Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12), 2481-2495 (2017)

[2] Cao, G., Wang, S., Liu, Y.: An improved algorithm for automatic road detection in high-resolution remote sensing images by means of geometric features and path opening. In: Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. pp. 1861-1864. IEEE (2015)

[3] Das, S., Mirnalinee, T., Varghese, K.: Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images. IEEE transactions on Geoscience and Remote sensing 49(10), 3906-3931 (2011)

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

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

[6] Howe, J., Casterline, M., Brown, A.: Solving spacenet road detection challenge with deep learning, https://spacenetchallenge.github.io/datasets/spacenetRoads-summary.html

[7] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)

[8] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)

[9] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234-241. Springer (2015)

[10] SpaceNet on Amazon Web Services (AWS): The spacenet roads dataset, https://spacenetchallenge.github.io/datasets/spacenetRoads-summary.html