

Article

# Aerial Image Road Extraction Based on an Improved Generative Adversarial Network

Xiangrong Zhang <sup>1,\*</sup>, Xiao Han <sup>1</sup>, Chen Li <sup>2</sup>, Xu Tang <sup>1</sup>, Huiyu Zhou <sup>3</sup> and Licheng Jiao <sup>1</sup>

<sup>1</sup> Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China; xiaohan@stu.xidian.edu.cn (X.H.); tangxu128@xidian.edu.cn (X.T.); lchjiao@mail.xidian.edu.cn (L.J.)

<sup>2</sup> Computer Science Department, Xi'an Jiaotong University, Xi'an 710049, China; cli@xjtu.edu.cn

<sup>3</sup> Department of Informatics, University of Leicester, Leicester LE1 7RH, UK; hz143@le.ac.uk

\* Correspondence: xrzhang@mail.xidian.edu.cn

Received: 19 February 2019; Accepted: 9 April 2019; Published: 17 April 2019



**Abstract:** Aerial photographs and satellite images are one of the resources used for earth observation. In practice, automated detection of roads on aerial images is of significant values for the application such as car navigation, law enforcement, and fire services. In this paper, we present a novel road extraction method from aerial images based on an improved generative adversarial network, which is an end-to-end framework only requiring a few samples for training. Experimental results on the Massachusetts Roads Dataset show that the proposed method provides better performance than several state of the art techniques in terms of detection accuracy, recall, precision and F1-score.

**Keywords:** deep learning; road extraction; generative adversarial network

---

## 1. Introduction

Roads act as a fundamental unit for many geographic information system applications, such as vehicle navigation, traffic management, and emergency response. It is also an important element of military surveying and mapping. In addition, for ensuring dynamics and accuracy, rapid city development requires frequent road updating, and the growing development of aerial technologies also provides an efficient, low-cost and reliable solution to receive dynamical road information. Besides aerial images, there are also other kinds of remote sensing data can be used for road extraction, such as hyperspectral images (HSI) [1,2], synthetic aperture radar (SAR) data [3–5], airborne laser scanning (ALS) data [6–8] and mobile laser scanner (MLS) data [9–11]. In this paper, we only focus on aerial images.

Traditional road network data mainly comes from manual extraction, which consumes intensive human resources. Aerial images provide abundant information about the ground covers. With the improvement of spatial resolution, it becomes an increasingly important data source for extracting road network information from aerial images. With the continuous updating of road information, traditional manual operation has been unable to meet the demand. Combining remote sensing technology with computer vision to extract road information from aerial images helps automate and accelerate road monitoring.

The research for road extraction has a long history before deep learning methods become widely used. We here summarize the traditional road extraction methods on three levels: feature, object, and knowledge levels.

(1) Road extraction methods on feature level: In previous work, roads were often extracted using spectral, geometric, topological and textural features of the roads. For example, a template matching

method was used to extract a certain number of seed pixels, or specific templates [12–14] and then generate roads based on the extracted seed points or templates. Using the characteristic edges of roads (e.g. parallelism), edges and parallel lines of roads were extracted using a distance function [15–19]. Using established mathematical models, such as Snake model [20–23] and Markov models, the edges of roads were examined. Some specific filters can be used to enhance road pixels for better road extraction [24]. Hierarchical feature level algorithms were usually performed based on spectral features of images [24,25], and the design of these algorithms was slightly simple but had impressive efficiency. However, these algorithms cannot produce satisfactory performance in complex environments and may produce ‘salt and pepper’ noise [24].

(2) The extraction methods based on object hierarchy: These object-based road extraction methods usually cluster image into a number of small areas (objects) or first segment the image, and then take a small area as a unit for road extraction. One example is the multi-resolution analysis method [26–29] based on the different resolutions of the aerial image or a single image with different scales, which can improve the accuracy of road extraction through two combination operations. Regional statistical analysis methods [30,31] were based on a probability model as well as the widely used road unit trimming [32–34] and joining [35] methods. These algorithms have achieved good performance in complex situations by merging pixels into homogeneous regions, which helps to reduce the influence of noise. However, these algorithms usually require initial segmentation or clustering of images, which has significantly influenced the final extraction precision, and is prone to the “sticky” phenomenon [29].

(3) Road information extraction methods based on knowledge level: Knowledge-based road extraction methods usually use the previous knowledge about roads, or the supplementary information to extract the targets. Methods such as multi-source data analysis based on existing road databases to guide or assist the extraction of road networks [36,37] are commonly used. They also exploited the self-characteristics of roads, such as spectrum and context [38–44]. These methods have achieved satisfactory results in complex situations [45–49]. However, these algorithms are not efficient. It is reasonable to combine multi-source to distinguish “different bodies with the same spectrum” or “same body with different spectra” but the acquisition of multi-resources data is relatively difficult, which limits the applications of the method.

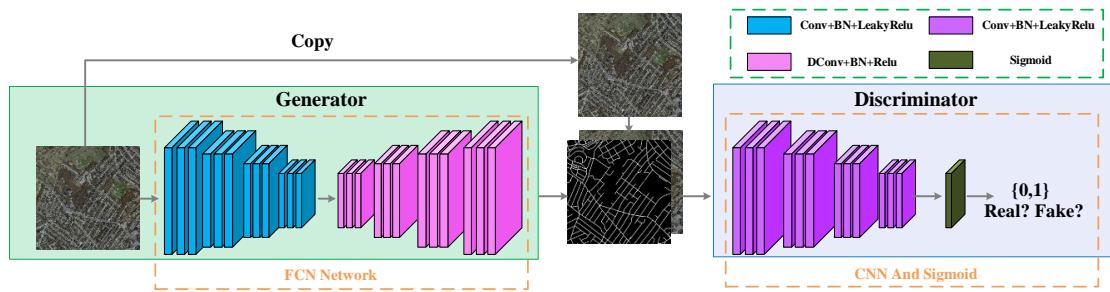
In recent years, deep learning has made successful applications in image analysis [50] and natural language processing [51]. Combining with low-level features, high-level representations of images can be formed, which have the capability of mining the distributions of data. The essence of deep learning is to learn more meaningful features from massive training data by constructing networks with multiple hidden layers, and its destination is to improve the accuracy of classification or prediction. Therefore, deep learning can be regarded as a case of feature learning. Methods based on deep learning transform features from one layer to another, which makes classification or prediction much easier.

There are some available deep learning based road extraction methods. Convolutional Neural networks (CNNs) have achieved excellent performance in image classification. Different from the classification of the whole image, road extraction is considered as a binary classification problem at the pixel level. It is needed to classify each pixel in the input aerial image as road or background. Therefore, the extraction method based on CNN usually uses a sliding window [52,53], by which the category of the central pixel of the window is obtained. Recently, a cascaded end-to-end convolutional neural network called CasNet [54] was proposed to detect roads and extract the centerline of an aerial image, ARCNet [55] combine CNN with attention mechanism to classify scene in very high resolution remote sensing images including roads. For some 3D data, such as HSI and ALS data, some end-to-end 3D CNN [56–58] has been proposed to detection and classification. A full convolutional network (FCN) can output pixel level classification information with the same size as the input image, which is suitable for road extraction in aerial images [59–62]. FCN includes down- and up-sampling. The up-sampling process in FCN uses the features obtained from the down-sampling process to increase the dimension by deconvolution layers, and obtains the same dimension of the classification map as

the input image. They can be divided into FCN-32s, FCN-16s, FCN-8s, FCN-4s, corresponding to the FCN networks with different upper sampling steps 32, 16, 8, and 4 respectively.

Some works also use Generative Adversarial Network (GAN) [63] model to extract roads from aerial images [64,65]. GAN as a kind of deep learning model is inspired by the zero-sum game theory. It contains a generator model  $G$  and a discriminator model  $D$  where the generator  $G$  can capture the distribution of the sample data, and the discriminator  $D$  is a binary classifier used to check whether the input is the real data or the generated sample. Generally speaking, GAN based methods regard road extraction as a task of image-to-image translation, so in [64,65] FCN was used as the architecture of the generator, and CNN as the discriminator. [64] used an encoder-decoder architecture in generator, and added a term of entropy loss in loss function; [65] used a two-stage framework to extract roads, in which two GANs were first used to detect roads and intersections and then the best covering road graph was found by applying a smoothing-based graph optimization procedure. Both methods chose encoder-decoder architecture in generator, which makes the generator be of poor ability to generate finer images.

In this paper, we propose an improved GAN model to extract roads from aerial images, and the overall framework of the proposed model is shown in Figure 1. In comparison to the available road extraction methods based on GAN, our proposed method has a simpler architecture than two stages method [65] and an easier loss function than [63]. In addition, Since GAN can produce promising results with a small amount of samples, which overcomes the scarcity in quantity of remote sensing images compared to natural images when using deep learning methods. According to the characteristics of GANs, we train a model to automatically generate a binary image of roads and the background. For the specific road extraction task, we enhance the original GAN loss function, adding a content-based loss item to ensure that the generated image is more accurate. Our approach has improved the extraction outcome compared to the other methods based on deep learning.



**Figure 1.** The framework of our method.

The remainder of this paper is organized as follows: Section 2 introduces the standard generative adversarial networks and the network structure we use for road extraction. Section 3 mainly shows our experimental results and comparisons. In Section 4, summary and future expectations are given.

## 2. Generative Adversarial Network for Road Extraction

### 2.1. Generative Adversarial Network

GAN has received much attention in recent years due to its ability to generate new samples similar to the training samples by learning the given samples' probability distributions. Different from the other deep learning models, GAN consists of two networks, i.e., generative and discriminate networks. As the name suggests, GAN can learn probability distributions from the given dataset and generate new samples similar to the given samples by using random noise. The training process of GAN can be seen as that the two networks optimize themselves against each other. Briefly, the generative network needs to generate more realistic samples to 'fool' the discriminate network, on the contrary, discriminate networks need to learn a better way to detect fake images generated by the

generative network. For the trained GAN, we can use the generative network to generate new samples, and also use the discriminate network for feature extraction or classification. The loss function of GAN is defined as follows:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

where  $p_{data}$  denotes the distribution of the real data (usually real images),  $p_z$  denotes the distribution of the input noise,  $D$  denotes the discriminate network,  $G$  denotes the generative network,  $x$  denotes the input from the real data, and  $z$  denotes the input from the random noise. The discriminate network needs to detect fake samples so as to make  $D(x) \rightarrow 1$  and  $D(G(z)) \rightarrow 0$ , whilst the generative network needs to generate realistic samples, leading to  $D(G(z)) \rightarrow 1$ .

In the last few years, the original GAN has a number of variants, such as Deep Convolutional Generative Adversarial Network (DCGAN) which combines CNN with GAN [66]; Conditional Generative Adversarial Network (CGAN) which takes the inputs with random noise or certain conditions [67]; Wasserstein GAN [68] which uses Wasserstein distance to define the loss function for solving the vanishing gradient problem [68,69]; CycleGAN [70] that has outstanding performance on the task of image translation.

DCGAN replaces the multilayer perceptron in GAN with CNN. In our approach, in the generative network, we use fractional-stride convolutions for up-sampling, which indicates that we map the input vector of low dimension into the image of high dimension. In the discriminate network, we use stride convolutions for down-sampling, mapping the input image into (0,1), which denotes the probability of the input belonging to the real data.

CGAN's input contains random noise  $z$ . Given condition  $y$ , the loss function of CGAN can be defined as follows:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (2)$$

For different tasks and datasets, condition  $y$  is different, but for a certain task and dataset, condition  $y$  should be the same for different samples in the same category.

Different from the previous methods, CycleGAN has outstanding performance on the task of image translation where we need pair datasets (the original and the transformed images). We treat image translation as a reversible process, in other words, when we use a GAN to translate image  $a \in A$  into image  $b \in B$ , images  $a$  and  $b$  have different styles of A and B, and we can also use another GAN to translate image  $b$  into image  $a$ . CycleGAN uses a cycle consistency loss function to guarantee the reversible process:

$$L_{cyc}(G_{AtoB}, G_{BtoA}, A, B) = E_{a \sim A}[\|G_{BtoA}(G_{AtoB}(a)) - a\|_1] + E_{b \sim B}[\|G_{AtoB}(G_{BtoA}(b)) - b\|_1] \quad (3)$$

where  $G_{AtoB}$  and  $G_{BtoA}$  denote two generators respectively,  $a$  and  $b$  denote images belonging to style A and B respectively, and  $\|\cdot\|_1$  denotes L1-norm. The loss function of CycleGAN can be defined as follows:

$$L(G_{AtoB}, G_{BtoA}, D_A, D_B) = L_G(G_{AtoB}, D_B, A, B) + L_G(G_{BtoA}, D_A, B, A) + L_{cyc}(G_{AtoB}, G_{BtoA}, A, B) \quad (4)$$

where  $D_A$  and  $D_B$  denote the discriminators corresponding to the generators  $G_{AtoB}$  and  $G_{BtoA}$  respectively,  $L_G$  denotes the GAN loss and  $L_{cyc}$  denotes the cycle consistency loss.

There are also some other methods based on GAN to conduct image translation, or called image-to-image translation such as DiscoGAN [71], DualGAN [72], and pix2pix [73].

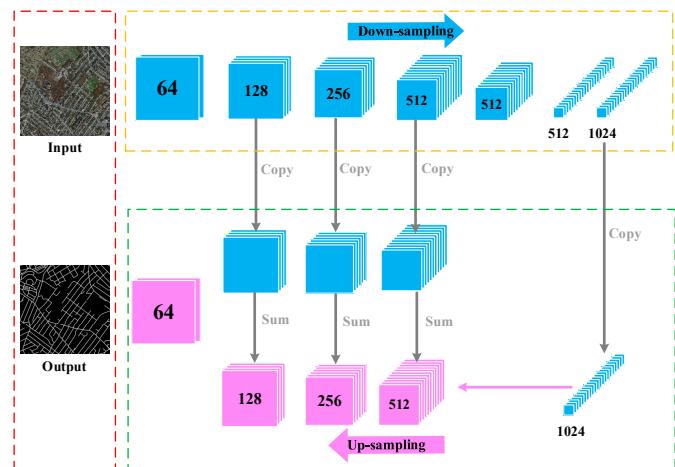
## 2.2. The Structure of Generative Adversarial Network for Road Extraction

Inspired by the outstanding performance of CGAN to image translation tasks, we use CGAN to extract roads in aerial images. Using CycleGAN and some other CGAN models, we can easily

translate images into another style, such as horse into zebra, day into night, summer into winter and so on, we want to use this idea translate aerial images into label images.

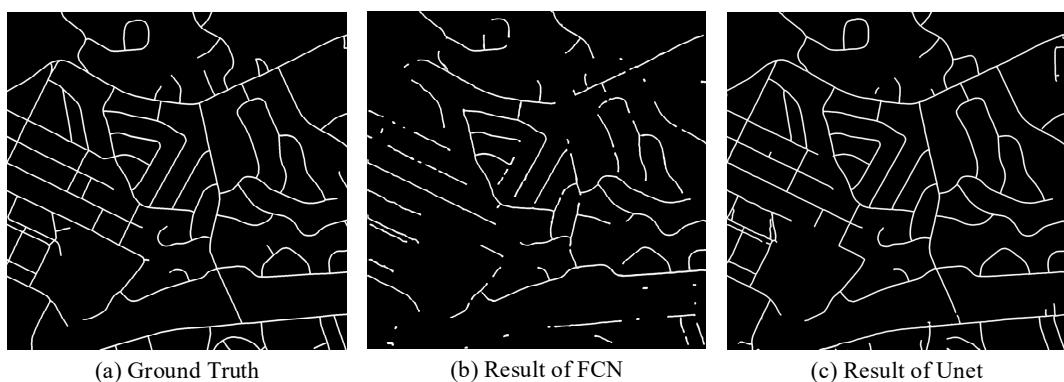
Road extraction can be regarded as a problem of binary classification at the pixel level, in which we predict whether a pixel belongs to roads or background. When targeting a binary image regardless of process details, we can regard this task as an image translation, where we want to translate the aerial image into the binary image that depicts the road and background.

We combine DCGAN and CGAN in our model to extract roads from aerial images, in short, we use a structure of DCGAN with certain conditions, and here we just input aerial images as our condition without any random noise. Due to the particularity of our task where the input and the output are images with the same size, we replace the deconvolution layers with FCN. The structure of the discriminator is the same as that of the discriminator of DCGAN. Our framework is shown in Figure 1. The structure of FCN is shown as Figure 2, where the blue blocks denote the down-sampling layers and the pink ones denote the up-sampling. The numbers on the blocks equal to the numbers of the feature maps of each layer. Our FCN model inherits the traditional FCN-4s structure. We also collect low level features by adding them to the feature maps of up-sampling. Different from the traditional FCN-4s model, we remove the pooling layers in terms of down-sampling and have different numbers of layers and feature maps in our FCN network.

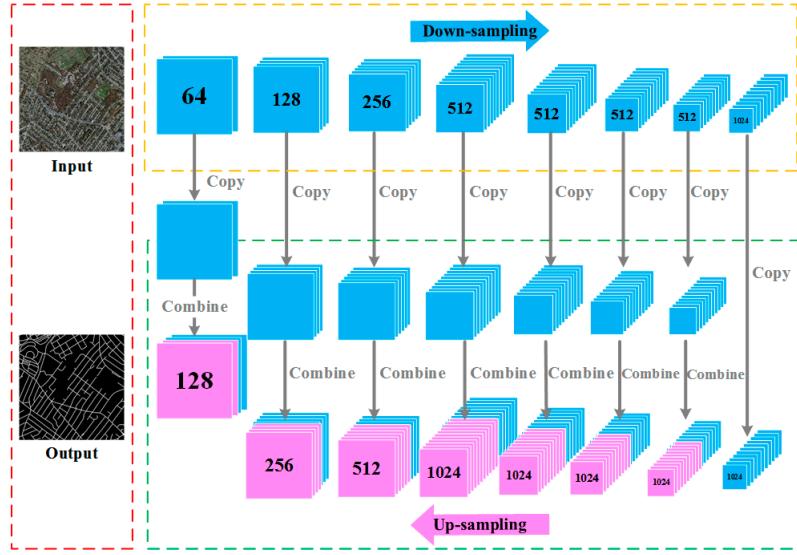


**Figure 2.** Structure of the FCN we used.

In fact, the structure of the FCN part has many choices. We here use Unet [74] due to its deeper architecture and good performance in road extraction task (see Figure 3). In our approach, Unet has a symmetric structure of down-sampling and up-sampling layers, including eight convolutional or deconvolutional layers, and there is not any pooling layer, as shown in Figure 4.



**Figure 3.** Results of the FCN and Unet. (a) Ground Truth; (b) Result of FCN; (c) Result of Unet.



**Figure 4.** Structure of the Unet we used.

### 2.3. Loss Function

Different from CycleGAN, we do not need two GANs or any cycle consistency as our road dataset contains the matches between the aerial and binary images. Moreover, the road extraction task does not need image translation. Our loss function can be defined as follows:

$$L = \operatorname{argmin}_G \max_D \alpha L_{cGAN}(G, D) + \beta L_{content}(G) \quad (5)$$

where  $L_{cGAN}$  denotes the loss of CGAN;  $L_{content}$  denotes the loss of the content, and  $\alpha$  and  $\beta$  are hyper-parameters to balance the two different losses.

First, we use an L1-norm loss function due to its simple form, i.e.,  $L_{content}(G) = L_1(G) = E_{x \sim X} \|G(x) - y\|_1$ , in which  $\|\cdot\|_1$  denotes the L1 distance,  $G(x)$  denotes the binary image generated from the input aerial image,  $y$  denotes the ground truth, and  $X$  denotes the aerial image in the training batch.  $E_{x \sim X} \|G(x) - y\|_1$  also can be written as follows:

$$E_{x \sim X} \|G(x) - y\|_1 = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^{M \times N} |G(x^i)_j - y_j^i| \quad (6)$$

where  $m$  denotes the batch size during the training,  $i$  denotes the index of the samples in the current batch,  $k$  denotes the sample which is different from the  $i$ th sample,  $j$  denotes the index of the pixels in each image, and  $M \times N$  denote the size of the image. The loss function of our model with L1-norm can be written as follows:

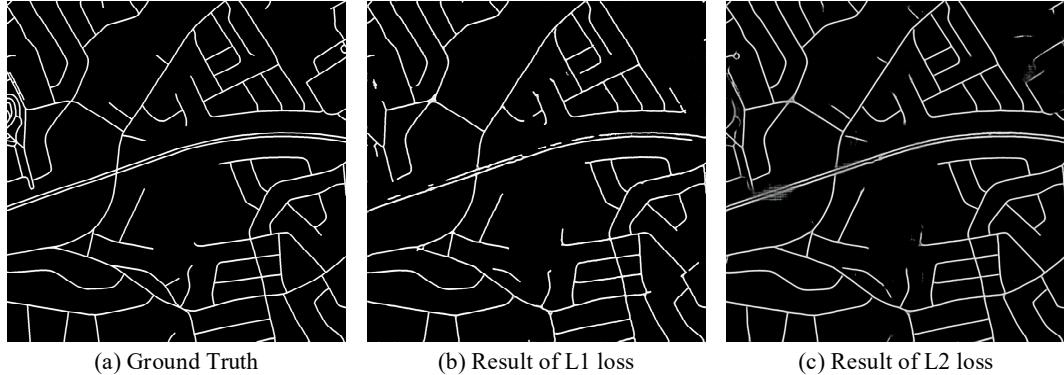
$$L = \arg \min_G \max_D \frac{1}{m} \sum_{i=1, k \neq i}^m \sum_{j=1}^{M \times N} (\alpha (\log D(y^k | x^k) + \log(1 - D(G(x^i)))) + \beta |G(x^i)_j - y_j^i|) \quad (7)$$

The L2-norm loss function is another choice, i.e.,  $L_{content}(G) = L_2(G) = E_{x \sim X} \|G(x) - y\|_2$ , and  $E_{x \sim X} \|G(x) - y\|_2$  can be written as Equation (8). The loss function of our model with L2 loss can be written as Equation (9).

$$E_{x \sim X} \|G(x) - y\|_2 = \sqrt{\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^{M \times N} (G(x^i)_j - y_j^i)^2} \quad (8)$$

$$L = \arg \min_G \max_D \frac{\alpha}{m} \sum_{i=1, k \neq i}^m (\log D(y^k | x^k) + \log(1 - D(G(x^i)))) + \frac{\beta}{m} \sum_{i=1}^m \sqrt{\sum_{j=1}^{M \times N} (G(x^i)_j - y_j^i)^2} \quad (9)$$

The results of our model with L1-norm and L2-norm are shown in Figure 5. Both of them have satisfactory performance, and we choose L2 loss as our element-wise loss function due to its better performance. Detailed comparison will be given in the experimental section.



**Figure 5.** Results using different losses. (a) Ground Truth; (b) Result of L1 loss; (c) Result of L2 loss.

In summary, we use Equation (9) as the loss function of our proposed method to extract roads from aerial images.

#### 2.4. Training Algorithm

In our proposed method, we choose Adaptive Moment Estimation (Adam) [75] to train our network because it is one of the best algorithms in deep learning to optimize the network parameters. In fact, we usually regard Adam as the combination of Stochastic Gradient Descent with Momentum (SGDM) [76] and Root Mean Square prop (RMSprop). When we update the parameters of the generative network, each iteration of the Adam algorithm can be written as Equations (10) to (13).

We first use Equation (10) to calculate the gradient of the generative network like other training algorithms based on mini-batch gradient descent, and then we use Equation (11) to calculate mini-batch gradient descent with momentum to avoid the oscillation of the gradient whilst accelerating convergence by retaining  $\rho_1$  of the gradient in the previous iterations and using only  $(1 - \rho_1)$  of the gradient in current iteration as our gradient to update the parameters in the current iteration.  $1 - \rho_1^t$  in the denominator is mainly used to remove the bias of the gradient in first few iterations. Next, we calculate RMSprop term by Equation (12), different from Equation (11), we use the square of the current gradient to replace the current gradient in Equation (11) and use the result to change the learning rate adaptively. Finally, we use Equation (13) to update the parameters of the generative network in the current iteration.

$$g_G = \frac{1}{m} \nabla_{\theta_G} \sum_{i=1}^m \left( \partial \log(1 - D(G(x^i))) + \beta \sqrt{\sum_{j=1}^{M \times N} (G(x^i)_j - y_j^i)^2} \right) \quad (10)$$

$$s_G = \frac{\rho_1 s_G + (1 - \rho_1) g_G}{1 - \rho_1^t} \quad (11)$$

$$r_G = \frac{\rho_2 r_G + (1 - \rho_2) g_G \odot g_G}{1 - \rho_2^t} \quad (12)$$

$$\theta_G = \theta_G - \varepsilon \frac{s_G}{\sqrt{r_G} + \delta} \quad (13)$$

where  $g_G$  denotes the gradient of the parameters in the generative network,  $s_G$  and  $r_G$  are corresponding to the first moment estimate and the second raw moment estimate with bias-correction respectively. In other words, we can regard  $s_G$  as a moment term, and  $r_G$  as an RMSprop term.  $\rho_1$  and  $\rho_2$  are two hyper-parameters called exponential decay rates which are usually set to be 0.9 and 0.999 in the experiments;  $\varepsilon$  denotes the learning rate and  $t$  denotes the number of the iterations;  $\delta$  is a small positive number to keep Equation (13) stable, which is experimentally set as 10-8; and  $\odot$  denotes the dot products of the matrix.

For the discriminate network, we use Equations (14) to (17) to update the paraments in each iteration.

$$g_D = \frac{\partial}{m} \nabla_{\theta_D} \sum_{i=1, k \neq i}^m (\log D(y^k | x^k) + \log(1 - D(G(x^i)))) \quad (14)$$

$$s_D = \frac{\rho_1 s_D + (1 - \rho_1) g_D}{1 - \rho_1^t} \quad (15)$$

$$r_D = \frac{\rho_2 r_D + (1 - \rho_2) g_D \odot g_D}{1 - \rho_2^t} \quad (16)$$

$$\theta_D = \theta_D + \varepsilon \frac{s_D}{\sqrt{r_D} + \delta} \quad (17)$$

where,  $g_D$  denotes the gradient of the parameters in the discriminate network,  $s_D$  and  $r_D$  are corresponding to the first moment and the second raw moment estimates with bias-correction, respectively.

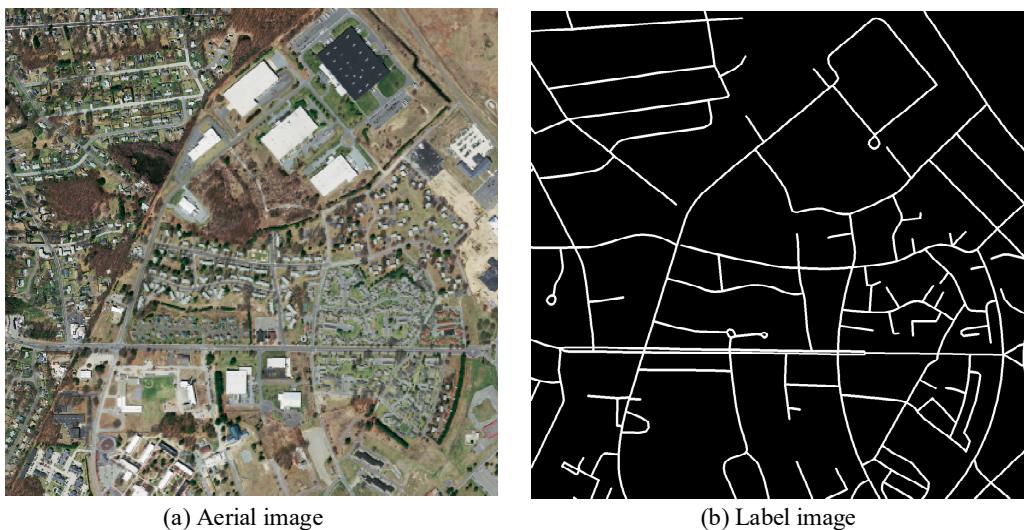
Since our model has two networks, i.e. generative and discriminate networks, whilst one loss function is defined as Equation (9), we train these two parts alternately. We first train the discriminate network using stochastic gradient ascend as shown in Equation (17) for one iteration, and then train the generative network using stochastic gradient descent shown in Equation (13) for another iteration till the training loss converges.

### 3. Experimental Results And Analysis

#### 3.1. Datasets

All the experiments are conducted on the Massachusetts Roads Dataset. This dataset contains aerial images depicting urban, suburban, and rural areas in the state of Massachusetts, USA. The dataset consists of 1171 aerial images, where each image is with the size of  $1500 \times 1500$  pixels. 1108 of these images have been randomly assigned to the training set. The remaining 49 and 14 images are allocated to the test and validation sets respectively. The dataset covers an area of approximately 2600 square kilometers in total, suggesting a Ground Sample Distance (GSD) of 1.0 meter per pixel.

Each aerial image has an accompanying binary label image, indicating whether a pixel in the aerial image belongs to either the road or non-road class. Road centerline vectors retrieved from the OpenStreetMap project were used to generate the label images. The vectors were rasterized as white lines with a line thickness of 7 pixels, which, based on the GSD, is equivalent to 7 meters on the ground. An aerial and label image pair example from this dataset is illustrated in Figure 6.



**Figure 6.** Image and label example taken from the test set in the Massachusetts Roads Dataset. (a) Aerial image; (b) Label image.

### 3.2. Evaluation Criteria

We use accuracy, precision, recall and F1-score to evaluate our results. F1-score is a number between 0 to 1 which considers both precision and recall. The closer F1-score approaches 1, the better results are achieved. Accuracy (A), precision (P), recall (R), and F1-score ( $F_1$ ) can be expressed as follows:

$$A = \frac{TP + TN}{TP + FN + FP + TN} \quad (18)$$

$$P = \frac{TP}{TP + FP} \quad (19)$$

$$R = \frac{TP}{TP + FN} \quad (20)$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (21)$$

where  $TP$ ,  $TN$ ,  $FN$  and  $FP$  denote the true positive, true negative, false negative, and true positive respectively.

### 3.3. Parameter Settings

We set  $\alpha = 1$ ,  $\beta = 300$  in the loss function shown in Equation (9) since they can obtain the best results. We choose Unet as our FCN part due to its good performance, and the number of the feature maps of each layer are shown in Figure 3. We use mini-batch Adam to train our network and set the learning rate as 0.0002, the momentum as 0.5, and the max epoch as 300. We perform our network on a GTX 1080ti GPU to accelerate the training process, which consumes about 500 seconds per epoch. We set the batch size as 2 which means that  $m = 2$  in Equation (9).

### 3.4. Comparison Algorithms

To verify the performance, our proposed method is compared with the other methods in three aspects: road extraction on a model of image to image translation (pix2pix), road extraction based on other deep learning methods and road extraction based on GANs.

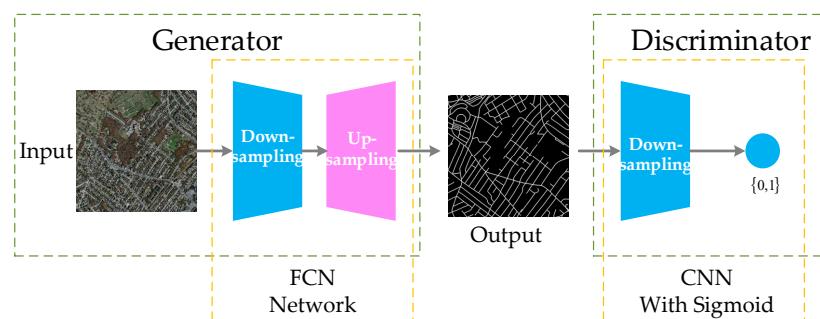
(1) Pix2pix [73]: Pix2pix is a kind of framework that can achieve the state of the art performance in image-to-image translation. The source code can be found at <https://github.com/phillipi/pix2pix>. During the training and testing, the architecture and hyper-parameters are same as [73].

(2) CNN [62]: The architecture and hyper-parameters we use are totally same as [62] which used a CNN with 6 hidden layers including 3 convolutional layers, 1 pooling layer and 2 full connection layers to extract roads in aerial images. The input of the network is a sliding window with the size  $64 \times 64$ , and the output is a label image with the size  $16 \times 16$  corresponding to the central region of the input. During the training, we randomly choose 200 windows of  $64 \times 64$  in each training image, and use Nesterov's Accelerated Gradient (NAG) algorithm [77] with learning rate 0.0025, max epoch 100. During the test, we set the sliding window as  $64 \times 64$  with stride 16 to cover each image of the test set. Furthermore, before the training, we do some data augmentation by mirror and reversal to obtain more training samples, we also throw some selected windows which only contain background to keep the training data of road and background classes balance.

(3) FCN [56]: We use FCN-8s and FCN-4s models respectively, both of which have 13 convolutional layers, 5 pooling layers and 2 deconvolutional layers (FCN-4s has 3 deconvolutional layers). During the down-sampling, we fine-tune partial parameters of VGG16 to accelerate convergence. The algorithm to train is stochastic gradient descent (SGD) with a small learning rate, and we set the max epoch to be 3000.

(4) DCGAN: The framework of DCGAN shown in Figure 7, and the input of the discriminator is only the output of the generator without any condition. Since we only use GAN loss, we set the loss function as:

$$L = \arg \min_G \max_D L_{DCGAN}(G, D) \quad (22)$$



**Figure 7.** The framework of DCGAN.

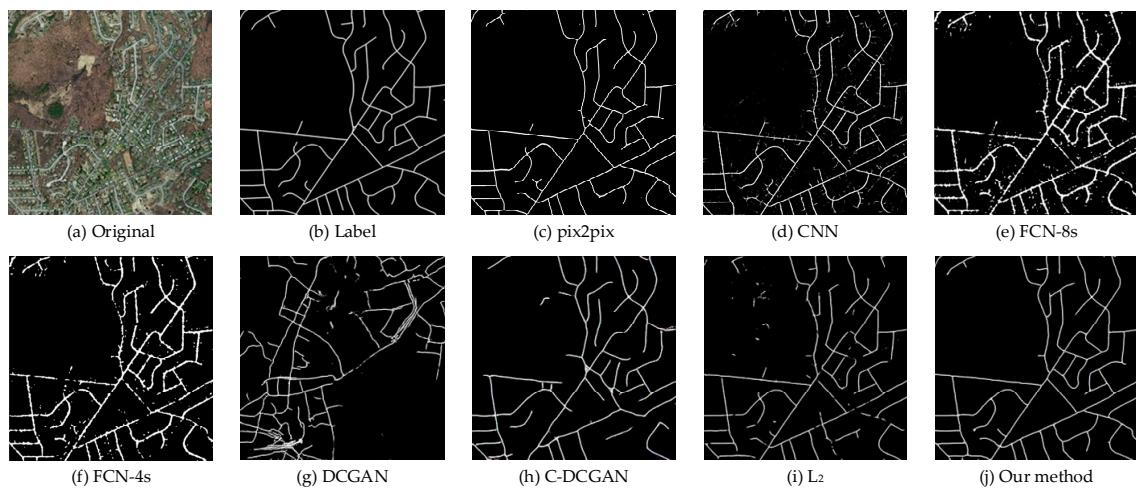
The FCN Network in Figure 7 is the same as Figure 4. The CNN with Sigmoid has same structure as Figure 1. We use mini-batch Adam with learning rate 0.0002, momentum 0.9, batch size 32, and max epoch 500.

(5) C-DCGAN: We add conditions to DCGAN in order to improve performance. The structure is the same as Figure 1. We use the loss function shown in Equation (9) with  $\alpha = 1, \beta = 0$ . Other parameters are the same as those presented in (4).

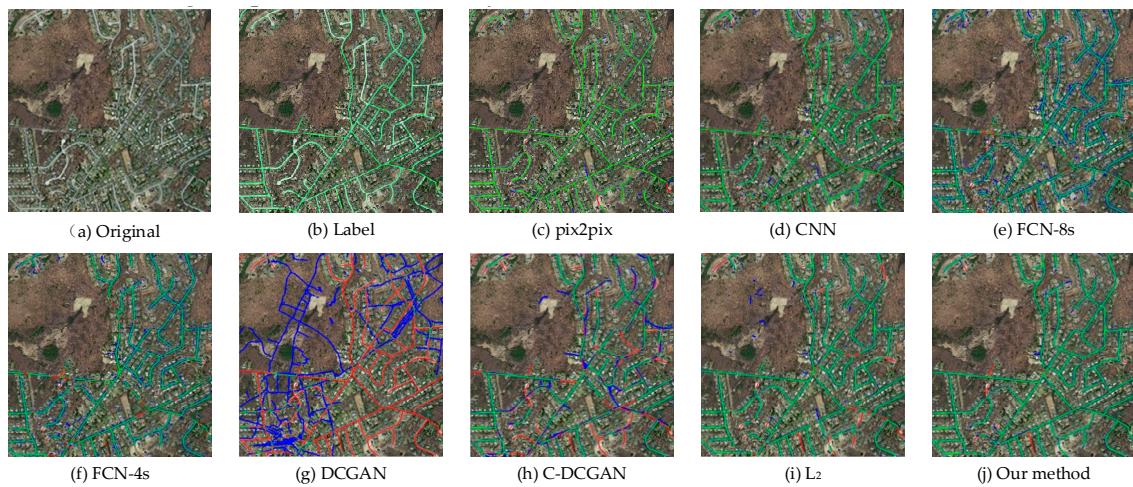
(6) L2 loss only: In order to explore the influence of L2 loss in our model, we set an experiment that only uses L2 loss. It means we set the loss function shown in Equation (9) with  $\alpha = 0, \beta = 300$ . Other parameters are the same as those shown in (5).

### 3.5. Experimental Results

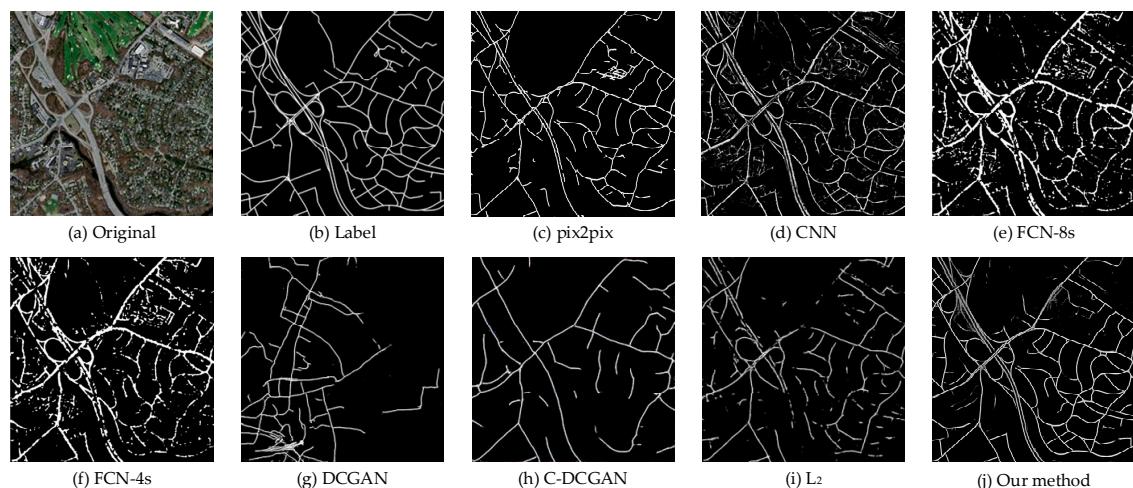
We use two types of Figures to show our results, shown in Figures 8–13. In the first type, we compare the extracted images against the label images using different methods, such as Figures 8, 10 and 12. In the second type, we can extract the hit/miss image by superposing the label images and the extracted images upon the original images to find the hit and miss areas, like Figures 9, 11 and 13. In these Figures, green lines (or points) denote the areas that we extract correctly, red lines (or points) denote the areas that contain roads but the model does not correctly extract, and blue lines (or points) denote the areas that do not contain any road but the model extracts 'roads' incorrectly.



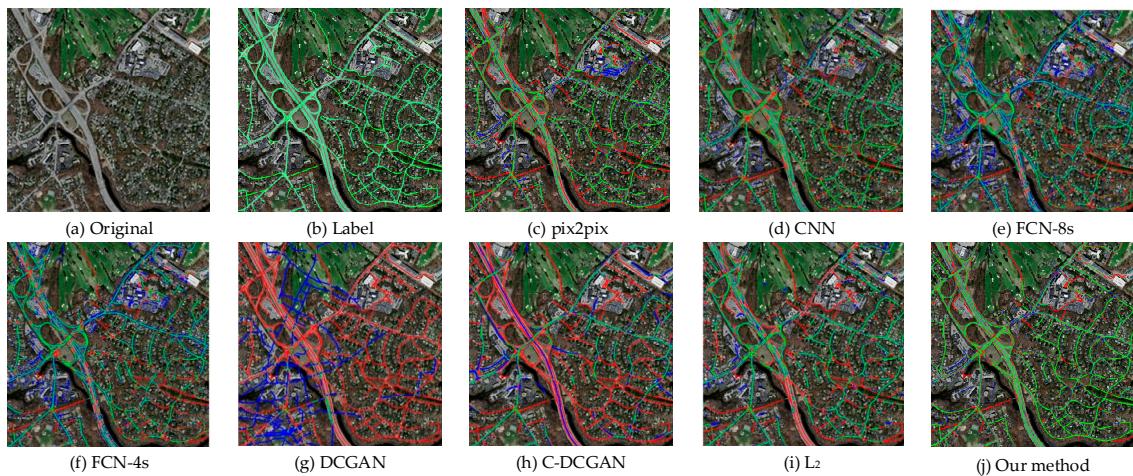
**Figure 8.** Results of road extraction on test image 1 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L<sub>2</sub> loss only; (j) Our proposed method.



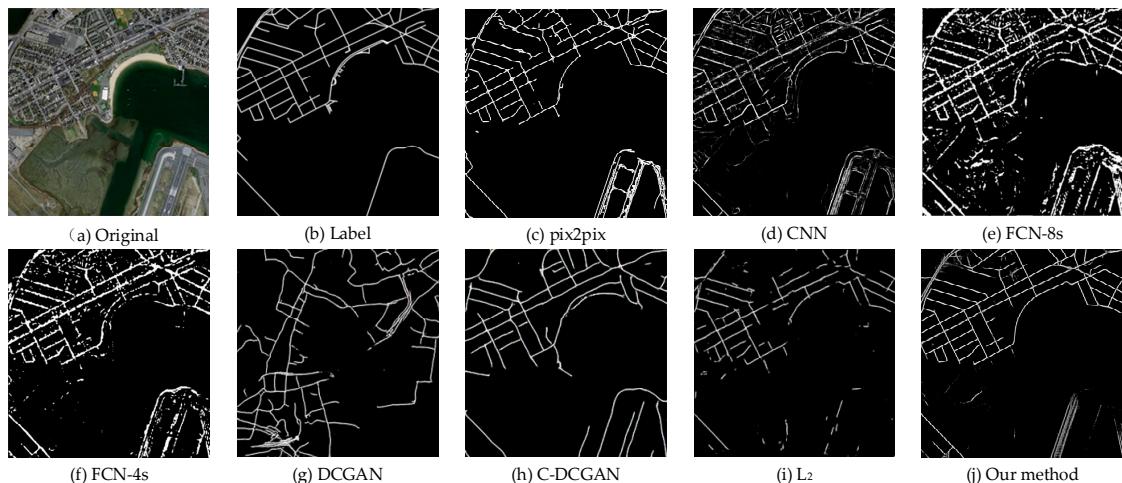
**Figure 9.** Hit/miss on test image 1 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L<sub>2</sub> loss only; (j) Our proposed method.



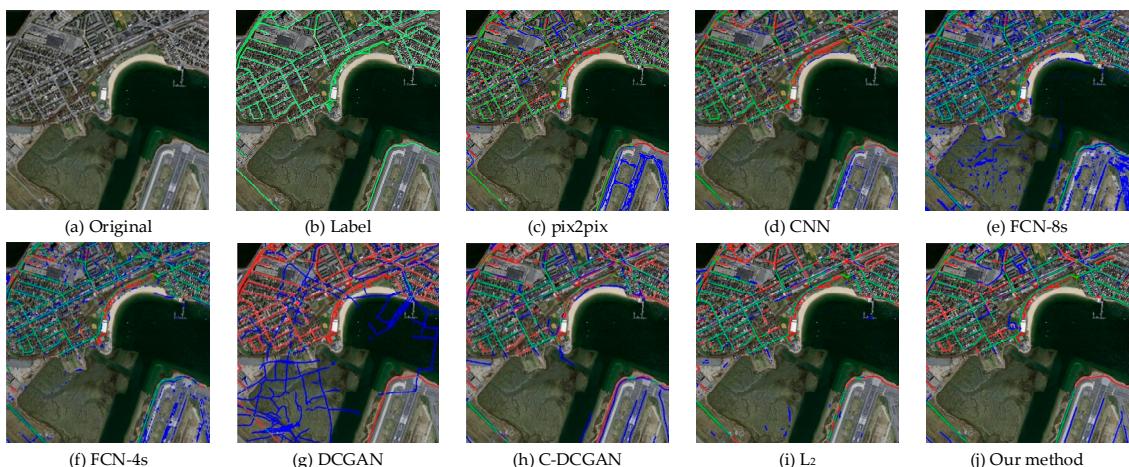
**Figure 10.** Results of road extraction on test image 2 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L<sub>2</sub> loss only; (j) Our proposed method.



**Figure 11.** Hit/miss on test image 2 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L2 loss only; (j) Our proposed method.



**Figure 12.** Results of road extraction on test image 3 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L2 loss only; (j) Our proposed method.



**Figure 13.** Hit/miss on test image 3 by different methods. (a) Original image; (b) Label image; (c) pix2pix; (d) CNN; (e) FCN-8s; (f) FCN-4s; (g) DCGAN; (h) C-DCGAN; (i) L2 loss only; (j) Our proposed method.

We choose three images to demonstrate the system performance, shown in Figures 8–13. The first image contains an area in which roads are narrow, like countryside or edge of the city, shown in Figures 8 and 9. The second image contains an area in which some roads are wide and the road network is relatively complex, like the center of the city, shown in Figures 10 and 11. The third image contains the body of water, and there are roads both inland and waterside, shown in Figures 12 and 13. From the sixth column of the Figure, we can see that although DCGAN uses deep neural networks, it fails to distinguish roads and the background. The main reason is that DCGAN is free to generate random images [61], and the content of the generated images may contain much noise.

Figures 8i, 9i, 10i, 11i, 12i and 13i show the results when we only use L2 loss as our loss function. The results present that L2 loss leads to better performance than C-DCGAN shown in Figures 8h, 9h, 10h, 11h, 12h and 13h.

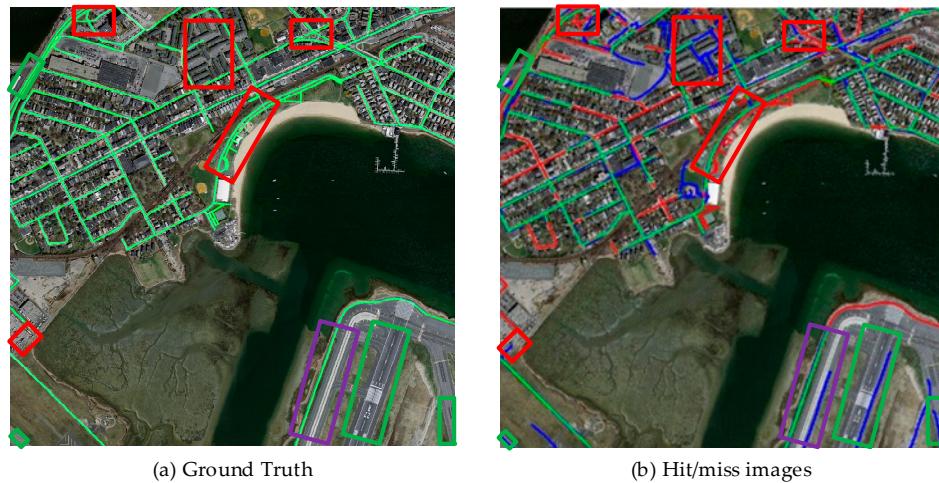
In our proposed model, we assign large hyperparameters to L2 loss ( $\alpha = 1, \beta = 300$ ), and the results of our model are shown in Figures 8j, 9j, 10j, 11j, 12j and 13j. These results are much better than the case where we only use CGAN or L2 loss function. CNN shown in Figures 8d, 9d, 10d, 11d, 12d and 13d and FCN Figure 8e,f, Figure 9e,f, Figure 10e,f, Figure 11e,f, Figure 12e,f and Figure 13e,f also have good performance visually.

Table 1 shows qualitative results of these 8 models, including average accuracy, precision, recall, and F1-score, which are calculated by Equations (18) to (21). The values of accuracy, precision and recall shown in Table 1 come from the average ones on the whole test set (49 images), and F1-score comes from the average precision and recall.

**Table 1.** Performance comparison of different methods on the test set.

Evaluation	Pix2pix	CNN	FCN-8s	FCN-4s	DCGAN	C-DCGAN	L <sub>2</sub>	Our Method
Accuracy	0.97	0.96	0.92	0.94	0.86	0.93	0.98	<b>0.98</b>
Precision	0.81	0.90	0.90	0.86	0.37	0.71	0.85	<b>0.93</b>
Recall	0.72	0.73	0.44	0.49	0.33	0.70	0.72	<b>0.82</b>
F <sub>1</sub> -score	0.76	0.81	0.59	0.62	0.35	0.70	0.78	<b>0.87</b>

From Table 1, we can see our method has the best performance due to the highest F1-score. CNN also has good results, but compared to our method, CNN based methods usually need to take data augmentation and apply sliding windows before the training, these will lead to more computational costs. In some way, CNN based methods are not an end-to-end framework for road extraction. Although our method has achieved the best performance, the results shown in the last row of Table 1 can be improved. Our model can be further improved to extract roads in the areas where road networks are complex, especially when for the thin roads, such as country roads. And in the cases when some objects are similar to roads, such as roofs, our model faces challenges to distinguish these regions, e.g. red blocks shown in Figure 14. In reality, different roads have different widths, but in Massachusetts Roads Dataset, different roads are labeled at the same width (7 pixel). It means that each road in the dataset is labeled with the width of 7 meters. Therefore, some incorrect results that extracted by our model comes from the miss of part of road width in the ground truth. This issue usually occurs when there are wide roads in the images like the purple block shown in Figure 14. For the green blocks shown in Figure 14, some roads are not properly labeled in the ground truth which also leads to the mistakes in road extraction.



**Figure 14.** Error analysis. (a) Ground Truth; (b) Hit/miss images.

### 3.6. Parameter Analysis

As said in Section 2.2, we choose Unet as our FCN element in the generative network due to its good performance, and in Section 2.3, we choose L2 loss as our element-wise loss.

In this section, we will provide statistics of accuracy, recall, precision and F1-score of our FCN structure, Unet, L1 loss and L2 loss as shown in Tables 2 and 3 to validate the choice of the FCN structure and the element-wise loss. From Table 3, it is observed that L2 loss provides better performance in comparison to L1 loss.

**Table 2.** Performance comparison of our FCN and Unet on the test set.

Methods	Accuracy	Precision	Recall	F <sub>1</sub> -Score
Our FCN	0.97	0.80	0.71	0.75
Our Unet	0.98	0.93	0.82	0.87

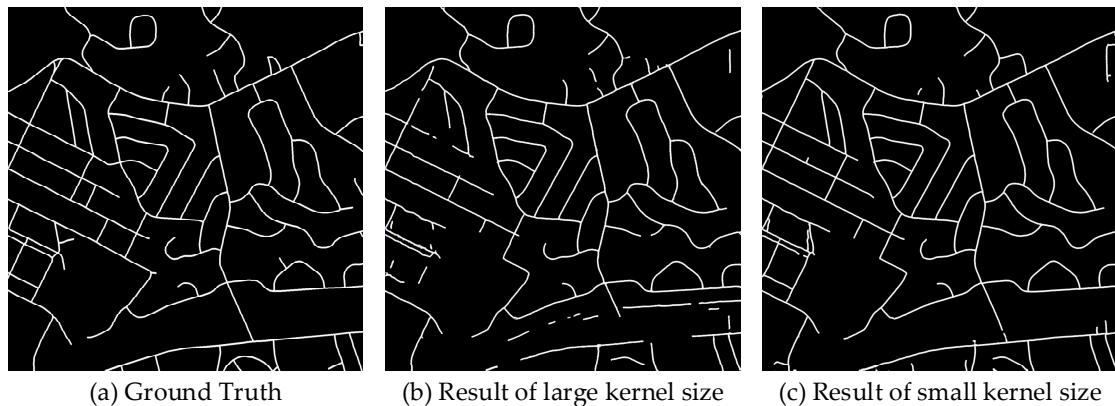
**Table 3.** Performance comparison of L1 loss and L2 loss on the test set.

Methods	Accuracy	Precision	Recall	F <sub>1</sub> -Score
L <sub>1</sub> Loss	0.98	0.87	0.81	0.84
L <sub>2</sub> Loss	0.98	0.93	0.82	0.87

Another important aspect is to determine a proper size of the convolutional kernel for CNN. We choose two groups of the kernel size, one is [4, 4, 4, 4, 3, 3, 3, 3] and the other is [11, 11, 7, 7, 5, 5, 4, 4], and the results are shown in Figure 15 and Table 4.

From Figure 15 and Table 4, we notice that the results of a small kernel size are better than those of a large kernel size. Generally speaking, the size of the kernel size is corresponding to the size of the receptive field. In the task of road extraction, we do not need a large receptive field because the roads in the aerial images are usually tiny so a smaller kernel size can lead to better results.

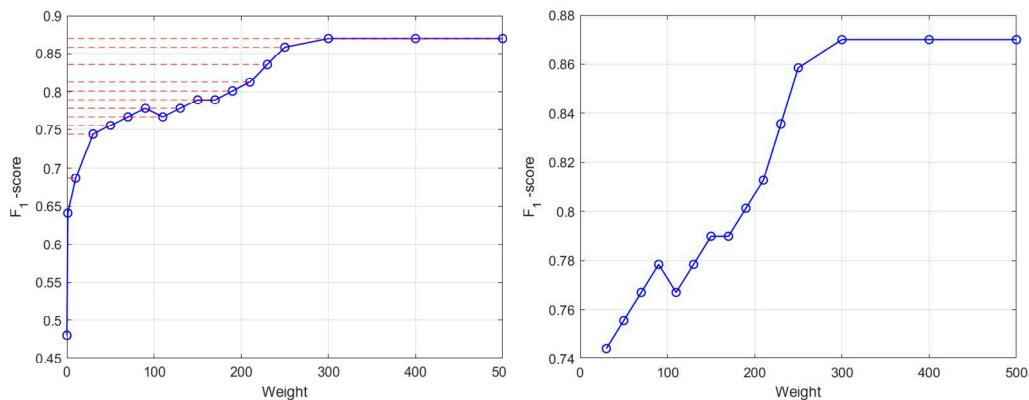
We use Equation (9) as the loss function in our model. How to balance C-DCGAN loss and L2 loss becomes a problem to be solved for road extraction task. From the above experiments, we find that L2 loss plays an important role in the task of road extraction, so we need to choose the best weight of L2 loss for Equation (9). We undertake experiments on the test set by different weights of L2 loss and plot the curves in Figure 16.



**Figure 15.** Results of different kernel size. (a) Ground Truth; (b) Result of large kernel size; (c) Result of small kernel size.

**Table 4.** Performance comparison of different kernel size on the test set.

Kernel Size	Accuracy	Precision	Recall	F <sub>1</sub> -Score
Large	0.97	0.84	0.75	0.79
Small	0.98	0.93	0.82	0.87



**Figure 16.** F1-scores of different weights for L2 loss.

From Figure 16, we can observe that when we increase the weight, F1-score on the whole test set increases as well, especially when the weights fall in the range of 0 to 250, and the performance on the test set significantly improves (F1-score increases from 0.74 to 0.86). When the weights are larger than 250, F1-score increases slowly and reaches 0.87 when the weight equals to 300. After this, F1-score does not change. So we set the weight as 300, and  $\alpha = 1, \beta = 300$  are used in Equation (9).

#### 4. Conclusions

In this paper, a novel end-to-end generative adversarial network has been proposed to perform the road extraction task in aerial images. A conditional GAN with L2 loss achieves better performance than the state of the art methods. Our proposed method, road extraction based on generative adversarial networks, does not need large training datasets, and still has the best performance. Compared to the other methods which also achieve good performance, our method is an end to end framework to extract roads and needs less computational costs.

Although the proposed model has achieved the best performance, extraction results on country roads and complex road network need to be further improved. The performance of remote sensing image processing methods based on deep neural networks relies on the given training dataset. Data

from different sources will have a great impact on model performance. Usually we use fine-tune to transfer trained model to new dataset with a few of training samples inform the new dataset. For zero training sample, we may need to use some prior knowledge of new data to assist the decision making. These are our future research directions.

**Author Contributions:** Conceptualization, X.Z. and X.H.; methodology, X.Z.; software, X.H.; validation, X.Z., X.H. and C.L.; formal analysis, X.Z. and C.L.; investigation, X.H. and X.T.; resources, X.Z.; data curation, X.H.; writing original draft preparation, X.H.; writing review and editing, X.Z., X.H., X.T., H.Z., C.L. and L.J.; visualization, X.H.; supervision, X.Z. and C.L.; project administration, X.Z.; funding acquisition, X.Z.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant 61772400, Grant 61801351, Grant 61501353, Grant 61772399, and Grant 61573267. H. Zhou was supported by UK EPSRC under Grant EP/N011074/1, Royal Society Newton Advanced Fellowship under Grant NA160342, and European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie grant agreement no.720325. The APC was funded by the National Natural Science Foundation of China under Grant 61772400, Grant 61501353, Grant 61772399, and Grant 61573267.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Resende, M.; Jorge, S.; Longhitano, G.; Quintanilha, J.A. Use of Hyperspectral and High Spatial Resolution Image Data in an Asphalted Urban Road Extraction. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Boston, MA, USA, 7–11 July 2008.
2. Shen, Z.; Huang, L.; Tao, J. Hyperspectral RS image road feature extraction based on SVM. *J. Chang'an Univ.* **2012**, *5*, 34–38.
3. Tupin, F.; Maitre, H.; Mangin, J.F.; Nicolas, J.M.; Pechersky, E. Detection of linear features in SAR images: Application to road network extraction. *IEEE Trans. Geosci. Remote Sens.* **2002**, *36*, 434–453. [[CrossRef](#)]
4. Corentin, H.; Majid, A.S.; Nina, M. Road Segmentation in SAR Satellite Images with Deep Fully Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *99*, 1–5.
5. Jiang, M.; Miao, Z.; Gamba, P.; Yong, B. Application of Multitemporal InSAR Covariance and Information Fusion to Robust Road Extraction. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3611–3622. [[CrossRef](#)]
6. Soilán, M.; Truong-Hong, L.; Riveiro, B.; Laefer, D. Automatic extraction of road features in urban environments using dense ALS data. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *64*, 226–236. [[CrossRef](#)]
7. Tiwari, P.S.; Pande, H.; Pandey, A.K. Automatic urban road extraction using airborne laser scanning/altimetry and high resolution satellite data. *J. Indian Soc. Remote Sens.* **2009**, *37*, 223–231. [[CrossRef](#)]
8. Matikainen, L.; Karila, K.; Hyppä, J.; Puttonen, E.; Litkey, P.; Ahokas, E. Feasibility Of Multispectral Airborne Laser Scanning for Land Cover Classification, Road Mapping And Map Updating. *ISPRS J. Photogramm. Remote Sens.* **2017**, *3*, 119–122. [[CrossRef](#)]
9. Balado, J.; Díaz-Vilarino, L.; Arias, P.; González-Jorge, H. Automatic classification of urban ground elements from mobile laser scanning data. *Autom. Constr.* **2018**, *86*, 226–239. [[CrossRef](#)]
10. Guan, H.; Li, J.; Yu, Y.; Chapman, M.; Wang, C. Automated Road Information Extraction from Mobile Laser Scanning Data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 194–205. [[CrossRef](#)]
11. Guan, H.; Li, J.; Yu, Y.; Wang, C.; Chapman, M.; Yang, B. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 93–107. [[CrossRef](#)]
12. Zhang, J.; Lin, X.; Liu, Z.; Shen, J. Semi-automatic road tracking by template matching and distance transformation in urban areas. *Int. J. Remote Sens.* **2011**, *32*, 8331–8347. [[CrossRef](#)]
13. Zhang, R.; Zhang, J.X.; Li, H.T. Semiautomatic extraction of ribbon roads from high resolution remotely sensed imagery based on angular texture signature and profile match. *J. Remote Sens.* **2008**, *12*, 224–232.
14. Coulibaly, I.; Spiric, N.; Sghaier, M.O.; Manzo-Vargas, W.; Lepage, R.; St.-Jacques, M. Road extraction from high resolution remote sensing image using multiresolution in case of major disaster. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 2712–2715.
15. Gaetano, R.; Zerubia, J.; Scarpa, G.; Poggi, G. Morphological road segmentation in urban areas from high resolution satellite images. In Proceedings of the 17th International Conference on Digital Signal Processing, Corfu, Greece, 6–8 July 2011; pp. 1–8.

16. Unsalan, C.; Sirmacek, B. Road network detection using probabilistic and graph theoretical methods. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4441–4453. [[CrossRef](#)]
17. Zhou, S.; Xu, Y. To extract roads with no clear and continuous boundaries in RS images. *Acta Geod. Cartogr. Sin.* **2008**, *37*, 301–307.
18. Airouche, M.; Zelmat, M.; Kidouche, M. Statistical edge detectors applied to SAR images. *Int. J. Comput. Commun. Control* **2008**, *3*, 144–149.
19. Zhang, G.; Song, K.; Zhao, P.; Cheng, J. An optimization method for road nets using improved hough transform with width-tolerant. *J. Geomat. Sci. Technol.* **2014**, *31*, 269–273.
20. Leninisha, S.; Vani, K. Water flow based geometric active deformable model for road network. *ISPRS J. Photogramm. Remote Sens.* **2015**, *102*, 140–147. [[CrossRef](#)]
21. Anil, P.N.; Natarajan, S. A novel approach using active contour model for semi-automatic road extraction from high resolution satellite imagery. In Proceedings of the International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; pp. 263–266.
22. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. *Int. J. Comput. Vis.* **1988**, *1*, 321–331. [[CrossRef](#)]
23. Tang, W. Road extraction in quaternion space from high spatial resolution remotely sensed images basing on GVF snake model. *J. Remote Sens.* **2011**, *15*, 1040–1052.
24. Chaudhuri, D.; Kushwaha, N.K.; Samal, A. Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1538–1544. [[CrossRef](#)]
25. Cheng, C.Q.; Ting, M.A. Automatic recognition of landscape linear features from high-resolution satellite images. *J. Remote Sens.* **2003**, *7*, 26–30.
26. Peng, T.; Jermyn, I.H.; Prinet, V.; Zerubia, J. An extended phase field higher-order active contour model for networks and its application to road network extraction from VHR satellite images. *Int. J. Comput. Vis.* **2008**, *88*, 509–520.
27. Laptev, I.; Baumgartner, A.; Steger, C. Automatic road extraction based on multi-scale modeling, context, and snakes. *ISPRS J. Photogramm. Remote Sens.* **1997**, *95*, 93–108.
28. Shen, Z.; Luo, J.; Gao, L. Road extraction from high-resolution remotely sensed panchromatic image in different research scales. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, USA, 25–30 July 2010; pp. 453–456.
29. Huang, X.; Zhang, L. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *Int. J. Remote Sens.* **2009**, *30*, 1977–1987. [[CrossRef](#)]
30. Yi, W.; Chen, Y.; Tang, H.; Deng, L. Experimental research on urban road extraction from high-resolution rs images using probabilistic topic models. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, USA, 25–30 July 2010; pp. 445–448.
31. Hedman, K.; Hirn, S.; Stilla, U. Road extraction from SAR multi-aspect data supported by a statistical context-based fusion. In Proceedings of the Urban Remote Sensing Joint Event, Paris, France, 11–13 April 2007; pp. 1–6.
32. Das, S.; Mirnalinee, T.T.; Varghese, K. Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3906–3931. [[CrossRef](#)]
33. Singh, P.P.; Garg, R.D. A two-stage framework for road extraction from high-resolution satellite images by using prominent features of impervious surfaces. *Int. J. Remote Sens.* **2014**, *35*, 8074–8107. [[CrossRef](#)]
34. Shi, W.; Miao, Z.; Debayle, J. An integrated method for urban main-road centerline extraction from optical remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3359–3372. [[CrossRef](#)]
35. Gamba, P.; Dell’Acqua, F.; Lisini, G. Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping, and simple topological concepts. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 387–391. [[CrossRef](#)]
36. Zhang, Q.; Coulombe, I. Automatic road change detection and GIS updating from high spatial remotely-sensed imagery. *Geo-Spat. Inf. Sci.* **2004**, *7*, 89–95.
37. Chen, H.; Yin, L.; Ma, L. Research on road information extraction from high resolution imagery based on global precedence. In Proceedings of the 2014 3rd International Workshop on Earth Observation and Remote Sensing Applications, Changsha, China, 11–14 June 2014; pp. 151–155.

38. Movaghati, S.; Moghaddamjoo, A.; Tavakoli, A. Road extraction from satellite images using particle filtering and extended Kalman filtering. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2807–2817. [CrossRef]
39. Poullis, C. Tensor-cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images. *ISPRS J. Photogramm. Remote Sens.* **2014**, *95*, 93–108. [CrossRef]
40. Miao, Z.; Shi, W.; Gamba, P.; Li, Z. An object-based method for road network extraction in VHR satellite images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4853–4862. [CrossRef]
41. Sghaier, M.O.; Lepage, R. Road extraction from very high resolution remote sensing optical images based on texture analysis and beamlet transform. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *9*, 1946–1958. [CrossRef]
42. Sun, Z.; Fang, H.; Deng, M.; Chen, A.; Yue, P.; Di, L. Regular shape similarity index: A novel index for accurate extraction of regular objects from remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3737–3748. [CrossRef]
43. Yin, D.; Du, S.; Wang, S.; Guo, Z. A direction-guided ant colony optimization method for extraction of urban road information from very-high-resolution images. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4785–4794. [CrossRef]
44. Li, M.; Stein, A.; Bijker, W.; Zhan, Q. Region-based urban road extraction from VHR satellite images using binary partition tree. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *44*, 217–225. [CrossRef]
45. Liu, R.; Song, J.; Miao, Q.; Xu, P.; Xue, Q. Road centerlines extraction from high resolution images based on an improved directional segmentation and road probability. *Neurocomputing* **2016**, *212*, 88–95. [CrossRef]
46. Courtrai, L.; Lefvre, S. Morphological path filtering at the region scale for efficient and robust road network extraction from satellite imagery. *Pattern Recognit. Lett.* **2016**, *83*, 195–204. [CrossRef]
47. Liu, R.; Miao, Q.; Huang, B.; Song, J.; Debayle, J. Improved road centerlines extraction in high-resolution remote sensing images using shear transform, directional morphological filtering and enhanced broken lines connection. *J. Vis. Commun. Image Represent.* **2016**, *40*, 300–311. [CrossRef]
48. Cheng, G.; Zhu, F.; Xiang, S.; Wang, Y.; Pan, C. Accurate urban road centerline extraction from VHR imagery via multiscale segmentation and tensor voting. *Neurocomputing* **2016**, *205*, 407–420. [CrossRef]
49. Hui, Z.; Hu, Y.; Jin, S.; Yao, Z.Y. Road centerline extraction from airborne lidar point cloud based on hierarchical fusion and optimization. *ISPRS J. Photogramm. Remote Sens.* **2016**, *118*, 22–36. [CrossRef]
50. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. In Proceedings of the Neural Information Processing Systems Conference, Lake Tahoe, NV, USA, 3–8 December 2012.
51. Srivastava, N.; Mansimov, E.; Salakhutdinov, R. Unsupervised learning of video representations using lstms. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 843–852.
52. Mnih, V.; Hinton, G.E. Learning to detect roads in high-resolution aerial images. In Proceedings of the 11th European Conference on Computer Vision. Springer: Berlin, Heidelberg, Germany, 5–11 September 2010; pp. 210–223.
53. Mnih, V.; Hinton, G. Learning to label aerial images from noisy data. In Proceedings of the International Conference on Machine Learning, Edinburgh, Scotland, 26 June–1 July 2012; pp. 567–574.
54. Cheng, G.; Wang, Y.; Xu, S.; Wang, H.; Xiang, S.; Pan, C. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3322–3337. [CrossRef]
55. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene Classification with Recurrent Attention of VHR Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1155–1167. [CrossRef]
56. Maturana, D.; Scherer, S. 3D Convolutional Neural Networks for landing zone detection from LiDAR. In Proceedings of the IEEE International Conference on Robotics & Automation, Seattle, WA, USA, 26–30 May 2015; pp. 3471–3478.
57. Ying, L.; Haokui, Z.; Qiang, S. Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67.
58. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]

59. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.
60. Zhong, Z.; Li, J.; Cui, W.; Jiang, H. Fully convolutional networks for building and road extraction: Preliminary results. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, 10–15 July 2016; pp. 1591–1594.
61. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [[CrossRef](#)]
62. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 645–657. [[CrossRef](#)]
63. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Neural Information Processing Systems Conference, Montréal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
64. Shi, Q.; Liu, X.; Li, X. Road detection from remote sensing images by generative adversarial networks. *IEEE Access* **2018**, *6*, 25486–25494. [[CrossRef](#)]
65. Costea, D.; Marcu, A.; Leordeanu, M.; Slusanschi, E. Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2100–2109.
66. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
67. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv*, 2014; arXiv:1411.1784.
68. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv*, 2017; arXiv:1701.07875.
69. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv*, 2017; arXiv:1701.04862.
70. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2242–2251.
71. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 1857–1865.
72. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2868–2876.
73. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 5967–5976.
74. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Germany, 2015; pp. 234–241.
75. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
76. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Netw.* **1999**, *12*, 145–151. [[CrossRef](#)]
77. Bengio, Y.; Boulanger-Lewandowski, N.; Pascanu, R. Advances in optimizing recurrent networks. In Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing, Vancouver, BC, Canada, 26–30 May 2013; pp. 8624–8628.

