






# Reconstruction Bias U-Net for Road Extraction From Optical Remote Sensing Images

Ziyi Chen , *Member, IEEE*, Cheng Wang , *Senior Member, IEEE*, Jonathan Li , *Senior Member, IEEE*, Nianci Xie , Yan Han , and Jixiang Du, *Member, IEEE*

**Abstract**—Automatic road extraction from remote sensing images plays an important role for navigation, intelligent transportation, and road network update, etc. Convolutional neural network (CNN)-based methods have presented many achievements for road extraction from remote sensing images. CNN-based methods require a large dataset with high quality labels for model training. However, there is still few standard and large dataset, which is specially designed for road extraction from optical remote sensing images. Besides, the existing end-to-end CNN models for road extraction from remote sensing images are usually with symmetric structure, studying on asymmetric structure between encoding and decoding is rare. To address the above problems, this article first provides a publicly available dataset LRSNY for road extraction from optical remote sensing images with manually labelled labels. Second, we propose a reconstruction bias U-Net for road extraction from remote sensing images. In our model, we increase the decoding branches to obtain multiple semantic information from different upsamplings. Experimental results show that our method achieves better performance compared with other six state-of-the-art segmentation models when testing on our LRSNY dataset. We also test on Massachusetts and Shaoshan datasets. The good performances on the two datasets further prove the effectiveness of our method.

**Index Terms**—CNN model, dataset, optical remote sensing image, road extraction.

## I. INTRODUCTION

**B**ENEFITING from the prosperity and development of navigation, automatic driving, smart city and intelligent transportation, etc., road network information plays a more and

more important role in our daily life. Due to the new road construction, road network information update is always necessary. There are many kinds of methods for road network information update, such as manually labeling, tracking the changes of cars' driving traces, and automatic road extraction from remote sensing images, etc. Tracking cars' driving traces may miss the appearance information of roads, such as width, border, and so on. Manually labeling is time consuming and with hard manual burden. Automatic road extraction from optical remote sensing images is a more economic and more time saving way compared with the traditional manual road areas labeling [1]. Because the attracting of high research values about road extraction from remote sensing images, much research has focused on automatic road extraction from remote sensing images [1]–[8].

Most state-of-the-art road extraction methods from remote sensing images are CNN models-based. Especially, due to the fast development of end-to-end semantic segmentation CNN models in natural images, the end-to-end semantic segmentation CNN models also achieve great success in road extraction from remote sensing images. However, there are still two aspects that need to be considered when using CNN segmentation models for road extraction from remote sensing images. First, CNN models need a large dataset with labels for training. For now, there is still few publicly available dataset which is large enough, well labeled, and specially designed for road extraction from remote sensing images. Constructing a large dataset for road extraction from remote sensing images with manually labeled labels is costly. Second, most current end-to-end CNN segmentation models use symmetric or approximate symmetric structures between encoding part and decoding part, such as U-Net [9], PSPNet [10], etc. However, as reconstruction is a more challenging job compared with feature extraction work, using neural network parts with similar complexity to finish jobs with different difficulties seems unreasonable, which may result to the unbalance between reconstruction capacity and feature extraction capacity.

To overcome the above two problems of road extraction from remote sensing images, this article's work major focuses on two aspects. First, we cost a major expenditure of time and effort to label a large dataset for road extraction from remote sensing images. The dataset is not only well labeled, but also large enough to train CNN models and obtain sound test evaluations. To make experiments, which using our dataset for training and testing, be fair among different CNN models, we have divided our dataset into constant training, validation, and

Manuscript received November 12, 2020; revised December 27, 2020; accepted January 19, 2021. Date of publication January 22, 2021; date of current version February 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62001175, in part by the Natural Science Foundation of Fujian Province under Grant 2019J01081, in part by the United National Natural Science Foundation of China under Grant U1605254, in part by the National Natural Science Foundation of China under Grant 6187606, Grant 61972167, and Grant 61673186, and in part by the Special National Key Research and Development Plan under Grant 2019YFC1604705. (Corresponding author: Jixiang Du.)

Ziyi Chen, Nianci Xie, Yan Han, and Jixiang Du are with the Department of Computer Science and Technology, Huaqiao University, Xiamen 361021, China (e-mail: chenzyihq@hqu.edu.cn; 592546548@qq.com; 1010786675@qq.com; jxdu@hqu.edu.cn).

Cheng Wang is with the School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: cwang@xmu.edu.cn).

Jonathan Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

This article has supplementary downloadable material available at <https://doi.org/10.1109/JSTARS.2021.3053603>, provided by the authors.

Digital Object Identifier 10.1109/JSTARS.2021.3053603

testing parts. Second, we propose a reconstruction bias U-Net for road extraction from optical remote sensing images. In our reconstruction bias U-Net, we use multiple operation couples of upsampling and convolution to increase the reconstruction ability for each upsampling layer. Finally, through increasing reconstruction operations, we can strengthen the reconstruction ability of CNN model and achieve better segmentation results in the experiments.

The contributions of this article are as follows.

- 1) We provide a publicly available large dataset with manually labeled labels for road extraction from remote sensing images.
- 2) We propose a reconstruction bias U-Net for road extraction from remote sensing images. Benefiting from increasing reconstruction ability, our new CNN model performs better compared with other state-of-the-art semantic segmentation CNN models.

## II. RELATED WORK

In this section, we give a review about datasets of semantic segmentation at first. In the second part, we give a detail review about road extraction from remote sensing images.

### A. Studies on Datasets for Semantic Segmentation

Since most current state-of-the-art semantic segmentation methods are CNN models-based, a large enough dataset with manually labeled labels is rather important for specific target segmentation research works. Many researchers have paid their large attention on and taken large efforts for semantic segmentation dataset construction [11]–[24]. Brostow *et al.* proposed first semantic segmentation video dataset called Cambridge-driving Labeled Video Database (CamVid) [19]. In this dataset, 32 classes were labeled in the videos with 30 Hz for more than 10 min. Cordts *et al.* proposed Cityscapes dataset, which includes 5000 street scenes and corresponding exquisite labels from 50 different cities [18]. Usually, the Cityscapes dataset can be split into 2975 training images, 500 validation images, and 1525 test images for total 19 classes (including roads in street viewpoint). Besides, 20 000 street scenes with coarse ground truth are also provided. Ros *et al.* proposed SYNTHIA dataset which is a large scale scene of virtual city, providing pixel-level labeling for 11 classes (including road areas) [20]. As the dataset was produced based on a virtual city scene, the images vary with different viewpoints, seasons, and weather. Huang *et al.* proposed ApolloScape for automatic driving research. In ApolloScape dataset, more than 147 000 frames with pixel-level semantic labeling are publicly open, covering three different cities. The above representative datasets are based on ground or street viewpoints, there are also many semantic segmentation dataset for remote sensing images. ISPRS dataset [17] provides 38 large aerial image patches with same size and their corresponding DSM data. The dataset greatly promote the CNN model research on the DSM segmentation of remote sensing images. Demir *et al.* raised a challenge competition about semantic segmentation from remote sensing images, including road or street net extraction [15]. Since the competition was over, the download of

the dataset is not available. Nigam *et al.* provided a semantic segmentation dataset obtained by aerial planes flying at height ranging from 5 to 50 m [14]. The dataset contains 3268 images with 11 labeled classes. Mohajerani *et al.* provided a dataset for cloud segmentation study, which include 38 Landsat images [12]. Each image contains four bands information: red, green, blue, and NIR. Schmitt *et al.* proposed a dataset for multispectral image fusion study based on deep learning [11]. Mnih provided an aerial image dataset for road extraction from remote sensing images [25]. Bastani *et al.* provided a road extraction dataset, which using aerial images covering 24 sq km around 15 cities [26]. However, in their provided link, they told readers their dataset cannot be publicly released due to copyrights. Cheng *et al.* told readers they will publicly open their road centerline extraction dataset [2]. We do not find the dataset download link in the paper. Zang *et al.* used Shaoshan dataset for road extraction, however, their dataset is not publicly available [27].

From the above introduction, we can know that many researchers have spent large energy on dataset construction for semantic segmentation based on deep learning. However, a dataset specifically for the study of road extraction from remote sensing images based on deep learning models is still scarce and much-needed.

### B. Road Extraction From Remote Sensing Images

Road extraction from remote sensing images usually contains two subtasks: road area extraction and road centerline extraction [28], [29]. Road area extraction methods produce pixel-level labeling of roads [1], [4], [29]–[38], while skeletons of roads are extracted for road centerline extraction [8], [27], [28], [39]–[46].

As roads have outstanding shape feature compared with other ground targets, the morphological features are utilized for road extraction [4], [28], [31], [47]. With the development of machine learning methods, such as support vector machine (SVM), many researchers used machine learning methods combining with artificial designed features for road extraction from remote sensing images and obtained many achievements [34], [36]. Poullis proposed a no threshold framework which called Tensor-Cuts, and applied the framework for preprocessing of road extraction from satellite images since the framework is particularly suitable for linear features extraction [36]. Movaghati *et al.* proposed a road extraction method from satellite images using particle filtering (PF) and extended Kalman filtering (EKF) [48]. The PF is combined with EKF to find best continuation of the road after an obstacle or junction, which achieved satisfactory results. Leninisha *et al.* presented a semiautomatic framework based on geometric active deformable model for road network extraction from high spatial remote sensing images. Different road junctions shape types were extracted using water flow technique, and they achieved good results on test images [49]. Lv *et al.* proposed an adaptive multifeature (which containing color, local entropy, and HSC features) sparsity-based model for road area extraction, and they achieved good results in the experiments [33].

Recently, deep convolutional neural networks (CNN) have led a series of breakthroughs for computer vision tasks [50]–[59].

CNN models also have achieved many success in road extraction from remote sensing images [8], [38], [46], [60]–[64]. Alshehhi *et al.* proposed a single patch-based CNN for extraction of roads and buildings from high-resolution remote sensing data [31]. Experiments were conducted on two challenging datasets to demonstrate the performance of the proposed network architecture. Cheng *et al.* used a cascaded end-to-end CNN for automatic road detection and centerline extraction, which obtained the state-of-the-art results in the experiments [28]. Zhang *et al.* used a semantic segmentation neural network, which combines the strengths of residual learning and U-Net [65] for road area extraction from remote sensing images [29]. They achieved better results compared with other state-of-the-arts approaches. Chen *et al.* proposed a road extraction approach from remote sensing images, which combines Dirichlet mixture models and CNN modes and achieved good results [1]. Ren *et al.* proposed a DA-CapsUNet for road extraction from remote sensing images [37]. In their approach, they used a capsule U-Net architecture to extract and fuse multiscale capsule features. They achieved quite good results in the experimental results.

### III. METHOD

In this section, we show the CNN architecture of our reconstruction bias U-Net first. Then, we give a detail introduction about reconstruction bias part in our CNN model. Finally, we illustrate the loss training of our model.

#### A. Model Architecture of Reconstruction Bias U-Net

Fig. 1 shows the model architecture of our reconstruction bias U-Net. The major backbone of our network is based on the U-Net [9], consisting of encoding and decoding two parts. The initial input of our network is a  $256 \times 256$  remote sensing image. In the encoding part, we use five groups of convolution, ReLU, and maxpooling operations. In the fourth and fifth groups, dropout operation is added to reduce the activated neuron weights, avoiding the over fitting of the network. In the decoding part, we first use four operation couples of upsampling, convolution, and ReLU to obtain four upsampling results. It should be emphasized that the kernel sizes of the four convolution operations are different, learning the reconstruction ability of using different sizes of context information. Then, we concatenate the four upsampling results of first upsampling layer with the output of max-pooling in the fourth operation couple of encoding part, followed by two operation couples of convolution and ReLU. In the second upsampling layer, we use five operation couples of upsampling, convolution, and ReLU. In the next, the five upsampling results and the output of max-pooling in the third operation couple of encoding part are concatenated and followed by two operation couples of convolution and ReLU. The operations of the third upsampling layer are just same with the operations in the first upsampling layer. The outputs of third upsampling layer and the output of max-pooling in the second operation couple of encoding part are concatenated and followed by two operation couples of convolution and ReLU. In the fourth upsampling layer, three operation couples of upsampling, convolution, and ReLU are used. The outputs of fourth upsampling layer and the max-pooling output in the first operation couple of encoding

part are concatenated and followed by five operation couples of convolution and ReLU. In the final two operations, convolution and classification with sigmoid activation function are used. The output of final classification operation is the road area segmentation result with a size of  $256 \times 256$ .

#### B. Details of Decoding Architecture

In the reconstruction bias U-Net, we use multiple operation couples of upsampling and convolution to increase the reconstruction ability. Table I shows the detail information of inputs, outputs, filters, and strides in each layer. From Table I, we can see that through increasing the multiple operation couples of upsampling and convolution, our decoding part occupies much more parameters than encoding part. For each upsampling operation, the upsampling size is fixed at  $2 \times 2$ . There are 4, 5, 4, and 3 upsampling operations in the first, second, third, and fourth upsampling layer, respectively. We design the four upsampling layers from the considering that middle layers need more upsampling operations to learn and joint more reconstruction information. The detail numbers are selected according to the ability of our GPU. If one has a more powerful GPU, he can add the upsampling numbers for each layer. In the first upsampling layer, the followed convolution operations have filter sizes of  $2 \times 2$ ,  $4 \times 4$ ,  $8 \times 8$ , and  $16 \times 16$ , respectively. The convolutional filter numbers are 512, 128, 64, and 32, respectively. The output size of first concatenation operation is  $32 \times 32 \times 1248$ , followed by two convolutions with 512 filters having a  $3 \times 3$  size. In the second upsampling layer, we use five couples of upsampling and convolution. The convolution operations have  $256 \times 2 \times 2$ ,  $64 \times 4 \times 4$ ,  $32 \times 8 \times 8$ ,  $16 \times 16 \times 16$ ,  $832 \times 32$  filters, respectively. The output size of second concatenation operation is  $64 \times 64 \times 632$ . In the third upsampling layer, we use four couples of upsampling and convolution. The convolution operations have  $128 \times 2 \times 2$ ,  $32 \times 4 \times 4$ ,  $16 \times 8 \times 8$ ,  $264 \times 64$  filters, respectively. The output size of third concatenation operation is  $128 \times 128 \times 306$ . In the final upsampling layer, we use three couples of upsampling and convolution. The convolution operations have  $64 \times 2 \times 2$ ,  $16 \times 4 \times 4$ ,  $88 \times 8$  filters, respectively. The output size of fourth concatenation operation is  $256 \times 256 \times 152$ . To further convert the output into  $256 \times 256 \times 1$ , the output of fourth concatenation is followed by five couples of convolution and ReLU. The five convolution operations have  $64 \times 3 \times 3$ ,  $64 \times 3 \times 3$ ,  $33 \times 3 \times 3$ ,  $31 \times 1 \times 1$ , and  $31 \times 1 \times 1$  filters, respectively. In the final layer, we use a sigmoid to obtain the final segmentation result. It should be noted that the numbers of upsampling and convolution operations in decoding part can be increased according to the GPU memory size.

#### C. Loss Function

Given a set of training images and the corresponding road area segmentation labels ( $I$ ,  $G$ ), the target function of the network can be represented as follows:

$$\text{Min } E(I, G, W) = \sum_{i=1}^N ||I_i * W - G_i||^2 \quad (1)$$

where  $N$  is the number of training images and  $W$  is the parameters of network. In our network training, we use binary cross entropy

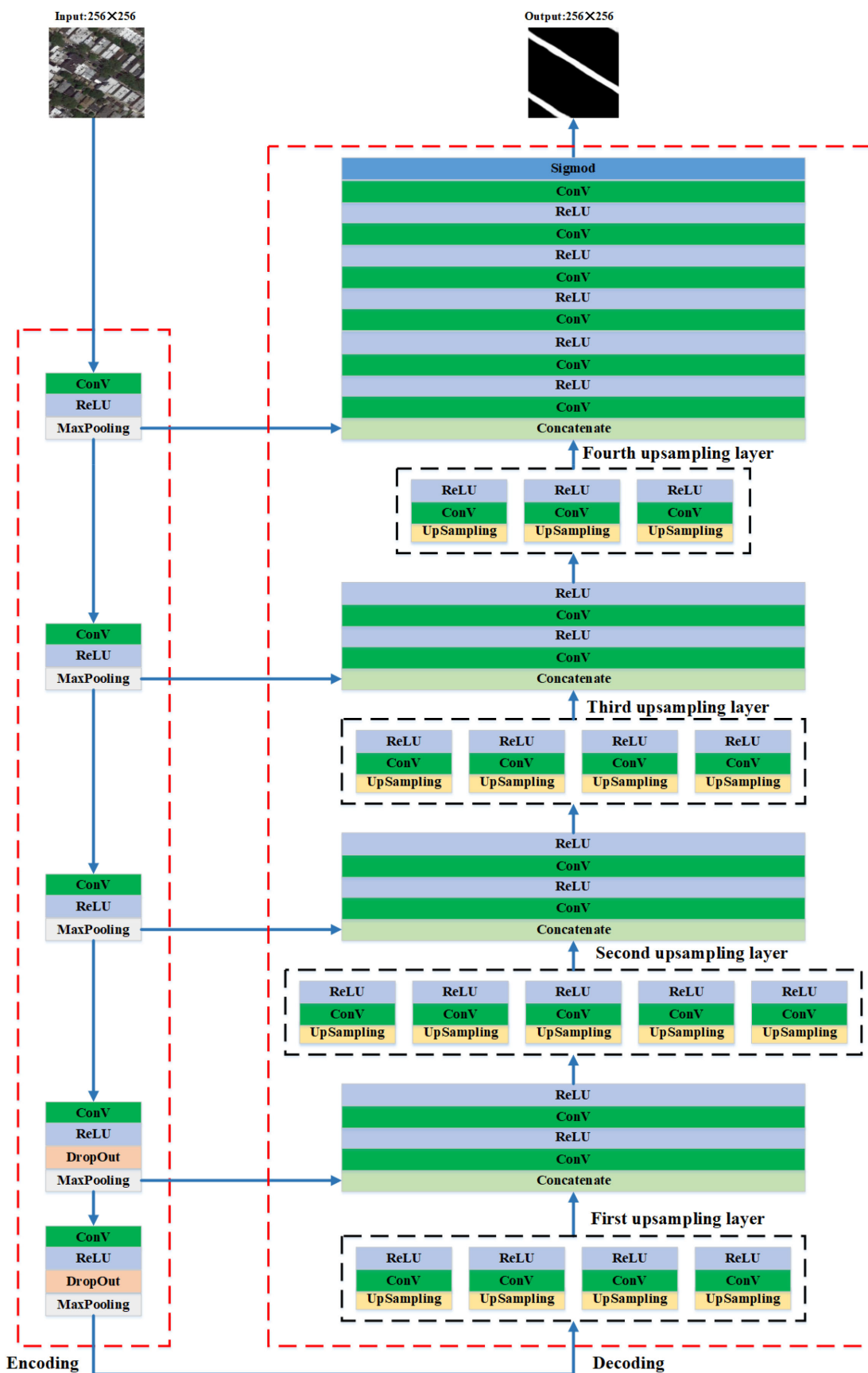


Fig. 1. Architecture of our reconstruction bias U-Net.

TABLE I  
DETAIL NETWORK ARCHITECTURE OF RECONSTRUCTION BIAS U-NET

	Operation couple	Filter	stride	Output size
Input				256×256×3
Encoding	$\begin{bmatrix} Conv \\ ReLu \\ MaxPooling \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3 \\ \dots \\ 2 \times 2 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix} \times 3$	$[128 \times 128 \times 64]$ $[64 \times 64 \times 128]$ $[32 \times 32 \times 256]$
	$\begin{bmatrix} Conv \\ ReLu \\ Dropout \\ MaxPooling \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \\ \dots \\ 0.5 \\ 3 \times 3 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix} \times 2$	$[16 \times 16 \times 512]$ $[16 \times 16 \times 1024]$
	$\begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix}$	$\begin{bmatrix} 2 \times 2 \\ 2 \times 2 / 512 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 4 \times 4 / 128 \\ 2 \times 2 \\ 16 \times 16 / 32 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 8 \times 8 / 64 \end{bmatrix} +$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \times 4$	$[32 \times 32 \times 512] + [32 \times 32 \times 128]$ $+ [32 \times 32 \times 64] + [32 \times 32 \times 32]$
	$\begin{bmatrix} Concat \\ Conv \\ ReLU \\ Conv \\ ReLu \end{bmatrix}$	$\begin{bmatrix} \dots \\ 3 \times 3 / 512 \\ \dots \\ 3 \times 3 / 512 \\ \dots \end{bmatrix}$	$\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$	$[32 \times 32 \times 512]$
	$\begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix}$	$\begin{bmatrix} 2 \times 2 \\ 2 \times 2 / 256 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 4 \times 4 / 64 \\ 2 \times 2 \\ 16 \times 16 / 16 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 8 \times 8 / 32 \\ 2 \times 2 \\ 32 \times 32 / 8 \end{bmatrix} +$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \times 5$	$[64 \times 64 \times 256] + [64 \times 64 \times 64]$ $+ [64 \times 64 \times 32] + [64 \times 64 \times 16]$ $+ [64 \times 64 \times 8]$
	$\begin{bmatrix} Concat \\ Conv \\ ReLU \\ Conv \\ ReLu \end{bmatrix}$	$\begin{bmatrix} \dots \\ 3 \times 3 / 256 \\ \dots \\ 3 \times 3 / 256 \\ \dots \end{bmatrix}$	$\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$	$[64 \times 64 \times 256]$
Decoding	$\begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix}$	$\begin{bmatrix} 2 \times 2 \\ 2 \times 2 / 128 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 4 \times 4 / 32 \\ 2 \times 2 \\ 64 \times 64 / 2 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 8 \times 8 / 16 \end{bmatrix} +$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \times 4$	$[128 \times 128 \times 128] + [128 \times 128 \times 32]$ $+ [128 \times 128 \times 16] + [128 \times 128 \times 2]$
	$\begin{bmatrix} Concat \\ Conv \\ ReLU \\ Conv \\ ReLu \end{bmatrix}$	$\begin{bmatrix} \dots \\ 3 \times 3 / 128 \\ \dots \\ 3 \times 3 / 128 \\ \dots \end{bmatrix}$	$\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$	$[128 \times 128 \times 128]$
	$\begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix} + \begin{bmatrix} Up \\ Conv \end{bmatrix}$	$\begin{bmatrix} 2 \times 2 \\ 2 \times 2 / 64 \end{bmatrix} + \begin{bmatrix} 2 \times 2 \\ 4 \times 4 / 16 \\ 2 \times 2 \\ 8 \times 8 / 8 \end{bmatrix} +$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \times 3$	$[256 \times 256 \times 64] + [256 \times 256 \times 16]$ $+ [256 \times 256 \times 8]$
	$\begin{bmatrix} Concat \\ Conv \\ ReLU \\ Conv \\ ReLu \\ Conv \\ ReLU \\ Conv \\ ReLU \\ Conv \\ ReLU \\ Sigmoid \end{bmatrix}$	$\begin{bmatrix} \dots \\ 3 \times 3 / 64 \\ \dots \\ 3 \times 3 / 64 \\ \dots \\ 3 \times 3 / 3 \\ \dots \\ 1 \times 1 / 3 \\ \dots \\ 1 \times 1 / 3 \\ \dots \\ 1 \times 1 / 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ \dots \\ 1 \\ \dots \\ 1 \\ \dots \\ 1 \\ \dots \\ 1 \\ \dots \\ 1 \end{bmatrix}$	$[256 \times 256 \times 1]$

as the loss function:

$$\mathcal{L}_{Net_w}(I) = - \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^L G_i(j, k) \cdot \log Net_w(I_i(j, k)) + (1 - G_i(j, k)) \cdot \log(1 - Net_w(I_i(j, k))) \quad (2)$$

where  $(M, L)$  represents the shape size of images  $I$  and  $Net_w(I_i(j, k))$  represents the network output of position  $(j, k)$  in image  $I_i$ . We use the Adam (adaptive moment estimation) and accuracy metrics to train our network. Other loss functions (such as mean squared error, pixel-wise cross entropy, etc.) and model training methods (such as stochastic gradient descent) are also can be used for training the network.

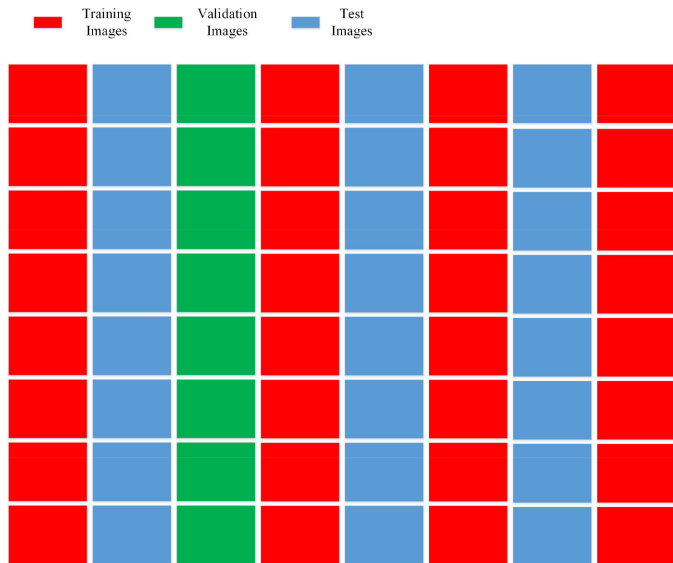


Fig. 2. Division strategy of our original large image. After division, the dataset is divided into training, validation, and test datasets.

#### IV. RESULTS AND DISCUSSION

In this section, we will first give an introduction about the dataset used in experiments. Then, the detail experimental implementations are introduced. Third, we present and analyze the experimental results on the tested datasets.

##### A. Dataset

In this article, we provide a publicly available road extraction dataset from high-resolution remote sensing images. The original image is a  $37949 \times 35341$  high-resolution remote sensing image, covering a center part of New York City and with a resolution of 0.5 m. We cut the original large image into pieces with a size of  $1000 \times 1000$ , generating 1368 images. The 1368 images are further divided into training, validation, and test images. The training, validation, and test images contain 716, 220, and 432 images, respectively.

Compared with Massachusetts Roads Dataset,<sup>1</sup> our dataset has a much higher resolution. The dataset size of our dataset is comparable with Massachusetts Roads Dataset. Massachusetts Roads Dataset contains 1108 training images, and each image has a size of  $1500 \times 1500$ . However, Massachusetts Roads Dataset only has 14 validation images and 49 test images. Compared with Cheng’s road extraction dataset [2], our dataset is much larger. For Deepglobe road extraction dataset,<sup>2</sup> the download was not available since the competition was over.

To maintain the position distribution balance of our dataset, we use a dividing strategy which iteratively assign training, validation, and test images at column level. For example, given an image which are cut into  $8 \times 8$  pieces, as shown in Fig. 2. Then, the pieces lay on first, fourth, sixth, and eighth columns are divided into training set. The pieces lay on second, fifth,

and seventh columns are assigned to test set. The pieces lay on third column are assigned to validation set. Through the above dividing strategy, we can guarantee the position distribution balance of different kinds of image sets. It should be noted that several validation images have small overlap areas with training images. The rest piece images at edge area of the original large image are usually not  $1000 \times 1000$ . To utilize the rest edge areas, we fill the pieces into  $1000 \times 1000$  using small overlaps with nearby column pieces (in our dataset are training pieces). The test images have no overlaps with training and validation images.

We manually label the ground truth of road areas in the original large image and implement the division as the same as the original image. Thus, it is one-to-one correspondence between the original image pieces and the ground truth label pieces. Because it will put a lot of pressure on the GPU memory size if we use training images with a size of  $1000 \times 1000$ , so we resize the images in our dataset into a size of  $256 \times 256$ . Although we only use training, validation, and test images with a size of  $256 \times 256$  in this article, the images with a size of  $1000 \times 1000$  are also reserved and publicly available in our dataset.

Fig. 3 shows several images and their corresponding ground truth in our dataset, including training, validation, and test datasets. From Fig. 3, we can see that the images in our dataset are rather challenging, as many road areas are occlude by trees, buildings, and cars, etc. It should be noted that we use white color (255, 255, 255) and black color (0, 0, 0) to respectively represent the road areas and background areas in our labeling images. Besides, because the training, validation, and test images have been divided and are constant, our dataset can be easily used for performance comparison. To make our dataset be easy to remember, we call our dataset as LRSNY (Large Road Segmentation Dataset from Optical Remote Sensing Images of New York). Our dataset can be download from the following website: <ftp://154.85.52.76/LRSNY/>.

To further verify the performance of the proposed reconstruction bias U-Net, we also compared the reconstruction bias U-Net with other methods on other two datasets: Massachusetts road extraction dataset and Shaoshan dataset [1].

For Massachusetts dataset, we divide the original training, validation, and test images into  $256 \times 256$  without overlappings, generating 27 700 training images, 350 validation images, and 1225 test images, respectively.

Shaoshan dataset is a Pleiades optical image covering parts of Shaoshan with an image size of  $11\,125 \times 7918$ . We follow the processing in [1], using 29 and 20 images for training and test, respectively. Each image has a size of  $1589 \times 1131$ . To suit the input size of  $256 \times 256$ , we further divide each image into  $256 \times 256$ . For training images, the division between generated neighbor  $256 \times 256$  images has a overlap. We generate  $256 \times 256$  training images with a skip step of 10 pixels in both row and column directions. Finally, we have 14 580 training images and 456 test images.

##### B. Experimental Implementation Detail

We train our model on a computer with Intel Core i9-9900X 3.5 GHz and 128 GB memories. The computer has two GPUs,

<sup>1</sup>[Online]. Available: <http://www.cs.toronto.edu/~vmnih/data/>

<sup>2</sup>[Online]. Available: <https://competitions.codalab.org/competitions/18467>

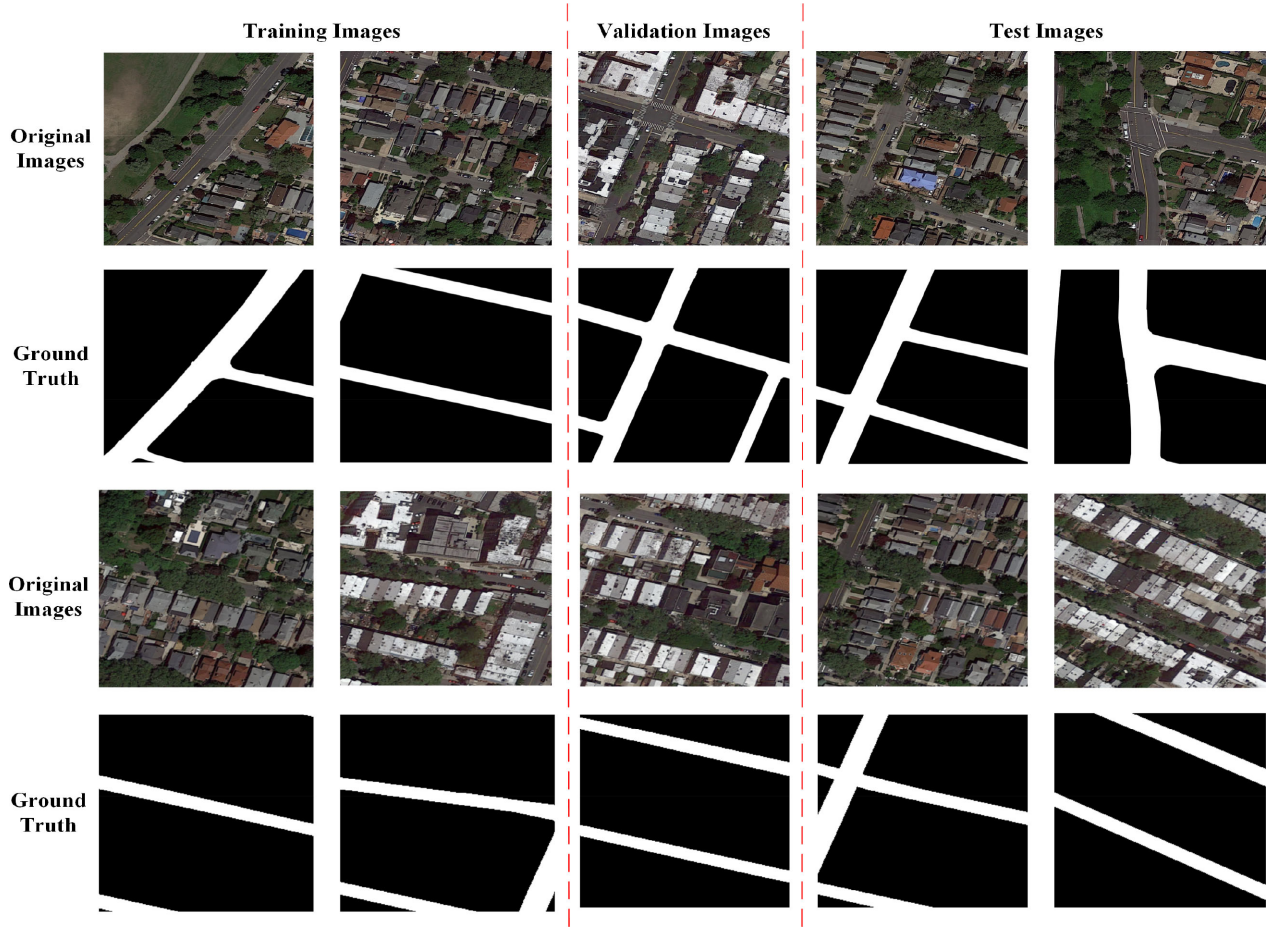


Fig. 3. Training, validation, test images, and the corresponding ground truth exhibition of our dataset.

which type is RTX 2080 Ti with 11 GB GPU memories. During training and test, we use only one GPU. When training our model, we set the training epoch as 200 and the learning rate as 0.0001. Our training batch size is 2. For model saving, we save the model with minimum loss within 200 epochs.

Our implementation is based on Python, Tensorflow,<sup>3</sup> and Keras.<sup>4</sup> To further strengthen the training stage and avoid the over fitting problem of model training, we utilize the data augmentation for training images. The rotation, zoom, shift, shear, and flip operations are all used in our training data augmentation. In our experiment, the rotation range is set at 0.9, the width shift range and height shift range are set at 0.1. And the shear range and zoom range are also set at 0.1.

### C. Evaluation Criteria

In this part, we give a brief introduction about the performance evaluation criteria used in our experiment. To comprehensively evaluate the performance of models, we use four evaluation criteria which are widely used for evaluating road segmentation

performance. The first three criteria are completeness, correctness, and quality, the representations are as follows:

$$\begin{aligned} \text{completeness} &= \frac{TP}{TP + FN} \\ \text{correctness} &= \frac{TP}{TP + FP}, \\ \text{quality} &= \frac{TP}{TP + FN + FP} \end{aligned} \quad (3)$$

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

The fourth evaluation criterion is PRI (Probabilistic Rand Index), which can be computed as follows:

$$\begin{aligned} \text{PRI}(S_{seg}, S_{gt}) &= \frac{1}{C_n^2} \sum_i \sum_{j(i \neq j)} [\psi(l_i = l_j \& l'_i = l'_j) \\ &\quad + \psi(l_i \neq l_j \& l'_i \neq l'_j)] \end{aligned} \quad (4)$$

where  $\psi$  is a discrimination function,  $l_i$  and  $l_j$  are the labels of  $S_{seg}$ ,  $l'_i$  and  $l'_j$  are the labels of  $S_{gt}$ , and  $c_n$  is the total pixel numbers of  $S_{seg}$ .

<sup>3</sup>[Online]. Available: <https://www.tensorflow.org/>

<sup>4</sup>[Online]. Available: <https://keras.io/>

TABLE II  
THE COMPARISON RESULTS AMONG OUR METHOD AND OTHER SIX  
STATE-OF-THE-ART SEGMENTATION METHODS TESTED ON OUR  
LRSNY DATASET

Method	Completeness	Correctness	Quality
U-Net[9]	<b>0.9398</b>	0.91599	0.86523
SegNet[66]	0.91233	0.93219	0.85555
PSPNet-50[10]	0.91221	0.94351	0.86497
Residual U-Net[67]	0.90218	0.90899	0.82744
DeepLabV3[68]	0.90588	0.9323	0.84996
DANet[69]	0.90504	<b>0.94521</b>	0.85993
Ours	0.92143	0.93864	<b>0.86908</b>

TABLE III  
COMPARISON RESULTS AMONG OUR METHOD AND OTHER SIX  
STATE-OF-THE-ART SEGMENTATION METHODS TESTED ON  
MASSACHUSETTS DATASET

Method	Completeness	Correctness	Quality
U-Net[9]	0.76628	0.80876	0.64873
SegNet[66]	0.72053	<b>0.82459</b>	0.62477
PSPNet-50[10]	0.76261	0.77921	0.6271
Residual U-Net[67]	<b>0.79688</b>	0.76862	0.64271
DeepLabV3[68]	0.73984	0.78322	0.6141
DANet[69]	0.74218	0.81209	0.6334
Ours	0.78525	0.79141	<b>0.65059</b>

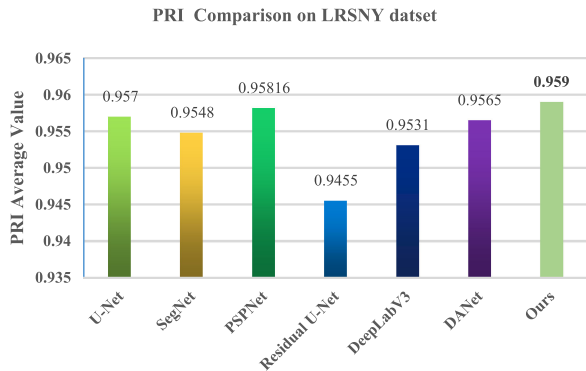


Fig. 4. PRI comparison among our method and other six state-of-the-art methods tested on LRSNY dataset.

#### D. Experimental Results

In this section, we first present the comparison results among our model and other state-of-the-art segmentation methods on LRSNY dataset. The compared methods include the original U-Net [9], SegNet [66], PSPNet [10], Residual U-Net [67], DeepLabV3 [68], and DANet [69]. For each method, we train the model 200 epochs, which is the same as our model training setting. Besides, to make the comparison fair, the training images augmentation operation and augmentation parameter settings are also just the same with our method for all the other comparing methods.

Table II shows the comparison results among our method and other six state-of-the-art segmentation methods tested on our LRSNY dataset. From Table II, we can see that our method obtains best performance according to the quality scores, which proves the effectiveness of our reconstruction bias U-Net. It can be seen that our method obtains about 0.4%, 1.4%, 0.4%, 4.2%, 2%, 1% higher quality scores than the original U-Net, SegNet, PSPNet-50, Residual U-Net, DeepLabV3, DANet, respectively.

Fig. 4 shows the PRI comparison among our method and other six state-of-the-art methods tested on LRSNY dataset. From Fig. 4, we can clearly see that our method obtains highest PRI score among all the methods, which further proves the good performance of our method.

Fig. 5 shows the visual segmentation results of different methods tested on our LRSNY dataset. The first to eighth columns are the original images, ground truth, results of original U-Net, SegNet, Residual U-Net, DeepLabV3, DANet and ours,

TABLE IV  
COMPARISON RESULTS AMONG OUR METHOD AND OTHER FIVE METHODS  
TESTED ON SHAOSHAN DATASET

Method	Completeness	Correctness	Quality
Zang <i>et al.</i> [27]	0.7786	0.7135	0.5963
ResidualUnet[29]	0.7454	0.9149	0.6970
PSPNet[10]	0.6888	0.9434	0.6615
ESPNet[70]	0.7431	0.8882	0.6795
Chen <i>et al.</i> [1]	<b>0.8247</b>	0.8443	0.7159
Ours	0.7605	<b>0.9526</b>	<b>0.7328</b>

respectively. From the figure, we can see that the segmentation results of our method obtain a better balance and stable performance when dealing with different images. The reason is due to our reconstruction bias part in our model, resulting to better performance.

In the next part of this section, we present the comparison results among our reconstruction bias U-Net with other six methods on Massachusetts road extraction dataset. In the experiment, the compared methods are the same as the six methods compared in the prior part. The training implementation parameters are also the same as the parameters in the prior experiment, except that we set training epoch as 20, steps per epoch as 27 700, and validation steps as 350.

Table III shows the comparison results among our method and other six state-of-the-art methods on Massachusetts dataset. In the table, we can see that our method achieved a quality score of 0.65059, which is the highest in all the compared methods. The original U-Net, SegNet, PSPNet-50, Residual U-Net, DeepLabV3, and DANet obtain quality scores of 0.64873, 0.62477, 0.6271, 0.64271, 0.6141, and 0.6334, respectively. We think the superior performance of our method is due to the more powerful reconstruction ability.

In the third part of this section, we compared our method with other five methods on Shaoshan dataset. The five methods include Zang *et al.* [27], Residual U-Net, PSPNet-50, ESPNet [70], and Chen *et al.* [1]. For these five methods, we just follow the results in [1].

Table IV shows the comparison results among our method and other five methods on Shaoshan dataset. From the table, we can see that our method also achieves the highest quality score in all the methods, obtaining a quality score as high as 0.7328. Compared with other five methods, we improve the performance about 14%, 4%, 7%, 6%, and 2%, respectively.



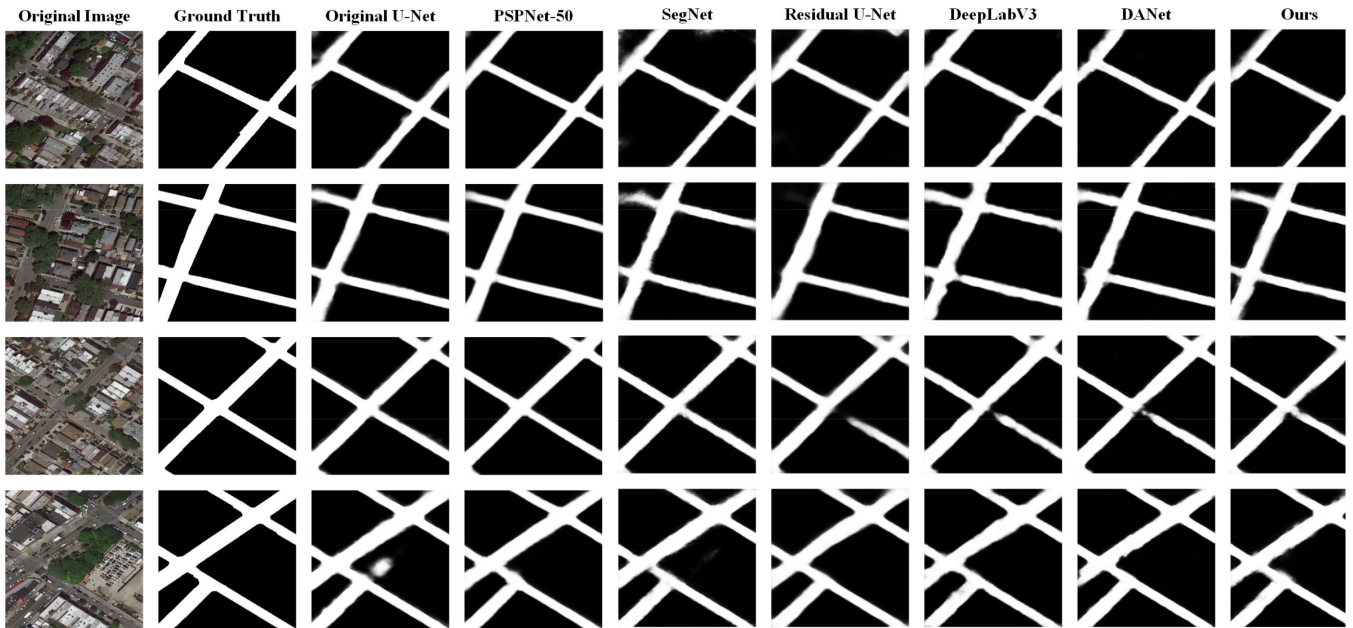


Fig. 5. Visual segmentation results' exhibition of different methods tested on our LRSNY dataset.

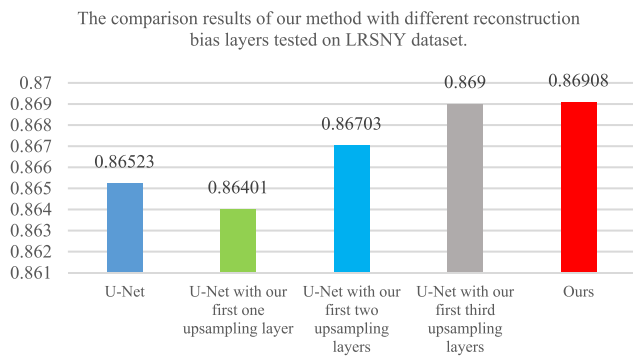


Fig. 6. Comparison results of our method with different reconstruction bias increased layers tested on LRSNY dataset.

This experimental results further prove the good performance of our method.

In the final part of this section, we analyze the effectiveness of reconstruction bias layers. Fig. 6 shows the comparison results of our method with different reconstruction bias layers tested on LRSNY dataset. U-Net with our first one upsampling layer means that the model only use the first enforced upsampling layer to replace the original first upsampling layer in original U-Net. From Fig. 6, we can see that for better performance the more enforced upsampling layers are used. This experiment proves the effectiveness of our reconstruction bias strategy.

## V. CONCLUSION

In this article, we proposed a reconstruction bias U-Net for road extraction from high-resolution optical remote sensing images. Our reconstruction bias U-Net consisted of two parts: Encoding and Decoding. In the Encoding part, we used five

operation couples of convolution, ReLU and max pooling. In the fourth and fifth operation couples, the drop out was also applied. In the Decoding part, we used four layers for upsampling. In each upsampling layer, multiple upsampling operation couples, which contain upsampling, convolution, and ReLU, were set up. Every upsampling filter size was fixed at  $2 \times 2$ , but the convolutional filter sizes are different among one upsampling layer, obtaining multiple reconstruction information. Each upsampling layer was followed by an operation couple of concatenation, convolution, ReLU, convolution, and ReLU. The final of our network was a sigmoid layer. The input and output sizes were  $256 \times 256 \times 3$  and  $256 \times 256 \times 1$ , respectively. Besides, we proposed a publicly available dataset about road extraction from remote sensing images, called LRSNY. In the experimental part, we compared our method with other six state-of-the-art image segmentation method on LRSNY. Experimental results showed the good performance of our method and proved the effectiveness of our reconstruction bias part in our model. To further verify the performance of our model, we also compared our method with other methods on another two datasets: Massachusetts and Shaoshan Datasets. The experimental results on the two datasets both prove the effectiveness of our reconstruction bias model, as our model achieved the best performance among the compared methods on the both two datasets.

## REFERENCES

- [1] Z. Chen, W. Fan, B. Zhong, J. Li, J. Du, and C. Wang, "Coarse-to-fine road extraction based on local Dirichlet mixture models and multiscale-high-order deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4283–4293, Oct. 2020.
- [2] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.

- [3] M. Maboudi, J. Amini, S. Malihi, and M. Hahn, "Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 138, pp. 151–163, 2018.
- [4] M. O. Sghaier and R. Lepage, "Road extraction from very high resolution remote sensing optical images based on texture analysis and beamlet transform," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 5, pp. 1946–1958, May 2017.
- [5] R. Alshehhi, P. R. Marpu, L. W. Woon, and M. Dalla Mura, "Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 139–149, 2017.
- [6] I. Coulibaly, N. Spiric, R. Lepage, and M. St-Jacques, "Semiautomatic road extraction from VHR images based on multiscale and spectral angle in case of earthquake," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 238–248, Jan. 2017.
- [7] Y. Zang, C. Wang, Y. Yu, L. Luo, K. Yang, and J. Li, "Joint enhancing filtering for road network extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1511–1525, Mar. 2016.
- [8] Q. Guo and Z. Wang, "A self-supervised learning framework for road centerline extraction from high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4451–4461, 2020.
- [9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
- [10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.
- [11] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "SEN12MS—A curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion," in *Proc. PIA19: Photogrammetric Image Anal.*, 2019, pp. 153–160.
- [12] S. Mohajerani and P. Saedi, "Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1029–1032.
- [13] X.-Y. Tong *et al.*, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, 2020, Art. no. 111322.
- [14] I. Nigam, C. Huang, and D. Ramanan, "Ensemble knowledge transfer for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1499–1508.
- [15] I. Demir *et al.*, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. Workshop (CVPRW)*, 2018, pp. 172–17209.
- [16] M. Zhang, X. Hu, L. Zhao, Y. Lv, M. Luo, and S. Pang, "Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images," *Remote Sens.*, vol. 9, no. 5, 2017, Art. no. 500.
- [17] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6054–6068, Nov. 2017.
- [18] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 3213–3223.
- [19] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.
- [20] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 3234–3243.
- [21] F. Yu *et al.*, "BDD100k: A diverse driving video database with scalable annotation tooling," 2018, *arXiv:1805.04687*.
- [22] G. Neuhold, T. Ollmann, S. R. Bulow, and P. Kotschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput.*, 2017, pp. 5000–5009.
- [23] X. Huang *et al.*, "The apollo-scapes dataset for autonomous driving," 2018, *arXiv:1803.06184*.
- [24] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The INRIA aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 3226–3229.
- [25] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation. Dept. Comput. Sci., Univ. Toronto, Toronto, Canada, 2013.
- [26] F. Bastani, S. He, S. Abbar, M. Alizadeh, and H. Balakrishnan, "Road-Tracer: Automatic extraction of road networks from aerial images," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4720–4728.
- [27] Y. Zang, C. Wang, Y. Yu, L. Luo, K. Yang, and J. Li, "Joint enhancing filtering for road network extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1511–1525, Mar. 2017.
- [28] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.
- [29] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.
- [30] M. Maboudi, J. Amini, S. Malihi, and M. Hahn, "Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 138, pp. 151–163, 2018.
- [31] R. Alshehhi, P. R. Marpu, L. W. Wei, and M. D. Mura, "Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 139–149, 2017.
- [32] I. Coulibaly, N. Spiric, R. Lepage, and M. St-Jacques, "Semiautomatic road extraction from VHR images based on multiscale and spectral angle in case of earthquake," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 238–248, Jan. 2018.
- [33] Z. Lv, Y. Jia, Q. Zhang, and Y. Chen, "An adaptive multifeature sparsity-based model for semiautomatic road extraction from high-resolution satellite images in urban areas," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1238–1242, Aug. 2017.
- [34] M. Li, A. Stein, W. Bijker, and Q. Zhan, "Region-based urban road extraction from VHR satellite images using binary partition tree," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 44, pp. 217–225, 2016.
- [35] D. Yin, S. Du, S. Wang, and Z. Guo, "A direction-guided ant colony optimization method for extraction of urban road information from very-high-resolution images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4785–4794, Oct. 2016.
- [36] C. Poullis, "Tensor-cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 95, no. 95, pp. 93–108, 2014.
- [37] Y. Ren, Y. Yu, and H. Guan, "DA-CapsUNet: A dual-attention capsule U-Net for road extraction from remote sensing imagery," *Remote Sens.*, vol. 12, no. 18, 2020, Art. no. 2866.
- [38] C. Tao, J. Qi, Y. Li, H. Wang, and H. Li, "Spatial information inference net: Road extraction using road-specific contextual information," *ISPRS J. Photogrammetry Remote Sens.*, vol. 158, pp. 155–166, 2019.
- [39] G. Cheng, F. Zhu, S. Xiang, and C. Pan, "Road centerline extraction via semisupervised segmentation and multidirectional nonmaximum suppression," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 545–549, Apr. 2017.
- [40] G. Cheng, F. Zhu, S. Xiang, Y. Wang, and C. Pan, "Accurate urban road centerline extraction from VHR imagery via multiscale segmentation and tensor voting," *Neurocomputing*, vol. 205, pp. 407–420, 2016.
- [41] Y. Zang, C. Wang, L. Cao, Y. Yu, and J. Li, "Road network extraction via aperiodic directional structure measurement," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3322–3335, Jun. 2016.
- [42] Z. Hui, Y. Hu, S. Jin, and Z. Y. Yao, "Road centerline extraction from airborne LiDAR point cloud based on hierarchical fusion and optimization," *ISPRS J. Photogrammetry Remote Sens.*, vol. 118, pp. 22–36, 2016.
- [43] L. Courtrais and S. Lefèvre, "Morphological path filtering at the region scale for efficient and robust road network extraction from satellite imagery," *Pattern Recognit. Lett.*, vol. 83, pp. 195–204, 2016.
- [44] R. Liu, J. Song, Q. Miao, P. Xu, and Q. Xue, "Road centerlines extraction from high resolution images based on an improved directional segmentation and road probability," *Neurocomputing*, vol. 212, pp. 88–95, 2016.
- [45] W. Shi, Z. Miao, and J. Debayle, "An integrated method for urban main-road centerline extraction from optical remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3359–3372, Jun. 2014.
- [46] R. Liu *et al.*, "Multiscale road centerlines extraction from high-resolution aerial imagery," *Neurocomputing*, vol. 329, pp. 384–396, 2019.
- [47] W. Shi, Z. Miao, Q. Wang, and H. Zhang, "Spectral–Spatial classification and shape features for urban road centerline extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 4, pp. 788–792, Apr. 2014.
- [48] S. Movaghghi, A. Moghaddamjoo, and A. Tavakoli, "Road extraction from satellite images using particle filtering and extended Kalman filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2807–2817, Jul. 2010.

- [49] S. Leninisha and K. Vani, "Water flow based geometric active deformable model for road network," *ISPRS J. Photogrammetry Remote Sens.*, vol. 102, pp. 140–147, 2015.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [52] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525.
- [53] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
- [54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [55] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
- [57] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020.
- [59] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
- [60] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, and Y. Liu, "RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2043–2056, Apr. 2018.
- [61] X. Lu *et al.*, "Multi-scale and multi-task deep learning framework for automatic road extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9362–9377, Nov. 2019.
- [62] L. Gao, W. Song, J. Dai, and Y. Chen, "Road extraction from high-resolution remote sensing imagery using refined deep residual convolutional neural network," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 552.
- [63] A. Abdollahi, B. Pradhan, and A. Alamri, "VNet: An end-to-end fully convolutional neural network for road extraction from high-resolution remote sensing data," *IEEE Access*, vol. 8, pp. 179424–179436, 2020.
- [64] X. Yang, X. Li, Y. Ye, R. Y. Lau, X. Zhang, and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7209–7220, Sep. 2019.
- [65] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.
- [66] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [67] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.
- [68] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [69] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.
- [70] M. Sachin, R. Mohammad, C. Anat, S. Linda, and H. Hannaneh, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. IEEE Conf. Eur. Conf. Comput. Vis.*, 2018.



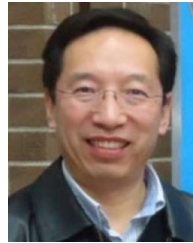
**Ziyi Chen** (Member, IEEE) received the Ph.D. degree in signal and information processing from Xiamen University, Xiamen, China, in 2016.

He is currently a Lecturer with the Department of Computer Science and Technology, Huaqiao University, China. His current research interests include computer vision, machine learning, and remote sensing image processing.



**Cheng Wang** (Senior Member, IEEE) received the Ph.D. degree in information communication engineering from the National University of Defense Technology, Hefei, China, in 2002.

He is currently a Professor with the School of Information Science and Engineering, Xiamen University, China. His current research interests include remote sensing image processing, mobile laser scanning data analysis, and multi-sensor fusion.



**Jonathan Li** (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000.

He is currently a Professor with the Department of Geography and Environmental Management, University of Waterloo, Canada. His current research interests include information extraction from earth observation images and 3-D surface reconstruction from mobile laser scanning point clouds.



**Nianci Xie** is currently a junior undergraduate student of Huaqiao University Quanzhou, China. His major is computer science.

His research interest includes computer vision.



**Yan Han** is currently a junior undergraduate student with Huaqiao University, Quanzhou, China. His major is computer science.

His research interest includes computer vision.



**Jixiang Du** (Member, IEEE) received the B.Sc. and M.Sc. degrees in vehicle engineering from the Hefei University of Technology, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005.

He is currently a Professor with the College of Computer Science and Technology, Huaqiao University, Xiamen, China.