A Two-Stage Road Segmentation Approach for Remote Sensing Images

Tianyu Li, Mary Comer, Josiane Zerubia

A Two-Stage Road Segmentation Approach for Remote Sensing Images

Tianyu Li[1][0000−0001−5069−1493], Mary Comer[1][0000−0001−8935−1365], and Josiane Zerubia[2,3][0000−0002−7444−0856]

[1] Purdue University, West Lafayette, IN, USA
cosmosyu.9@gmail.com
[2] Inria, Sophia Antipolis, France
[3] Université Côte d'Azur, France

Abstract. Many road segmentation methods based on CNNs have been proposed for remote sensing images in recent years. Although these techniques show great performance in various applications, there are still problems in road segmentation, due to the existence of complex backgrounds, illumination changes, and occlusions due to trees and cars. In this paper, we propose a two-stage strategy for road segmentation. A probability map is generated in the first stage by a selected network (ResUnet is used as a case study in this paper), then we attach the probability map image to the original RGB images and feed the resulting four images to a U-Net-like network in the second stage to get a refined result. Our experiments on the Massachusetts road dataset show the average IoU can increase up to 3% from stage one to stage two, which achieves state-of-the-art results on this dataset. Moreover, from the qualitative results, obvious improvements from stage one to stage two can be seen: fewer false positives and better connection of road lines.

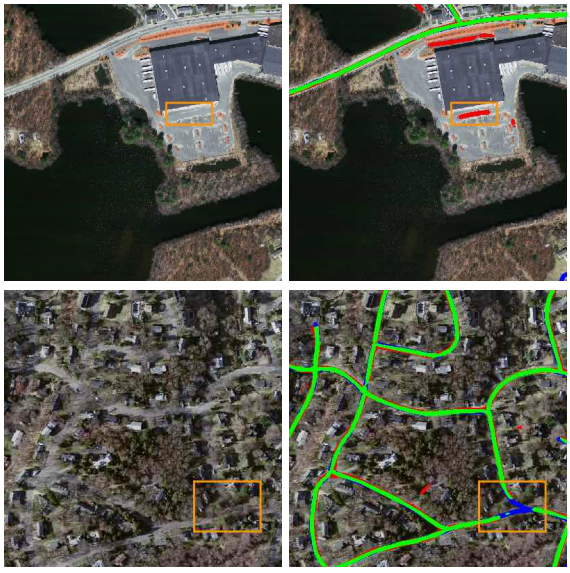
Keywords: Remote sensing · Road segmentation · Convolutional neural network (CNN) · Two stage learning.

1 Introduction

With the development of new technology in satellites, cameras and communications, massive amounts of high-resolution remotely-sensed images of geographical surfaces have become accessible to researchers. To help analyze these massive image datasets, automatic road segmentation has become more and more important. The road segmentation result can be used in urban planning [28], updating geographic information system (GIS) databases [1] and road routing [8].

Traditionally, researchers designed road segmentation algorithms based on some hand-crafted features [12, 14, 16], such as the contrast between a road and its background, the shape and color of road, the edges and so on. The segmentation results are highly dependent on the quality of these features. Compared to manually selected features, the rise of convolutional neural networks (CNN) [10, 21] provides a better solution to road segmentation, with features

generated by learning from labelled training data. Many CNN-based methods have been proposed for road segmentation. For instance, in [22], Ronneberger et al. proposed the U-Net convolutional network for biomedical image segmentation, which has been shown to work well for road segmentation also. Zhang et al. [25], built a deep residual U-Net by combining residual unit [11] with U-Net. Zhou et al. [27] proposed D-LinkNet, consisting of ResNet-34 pre-trained on ImageNet, and a decoder. Feng et al. [9] proposed an attention mechanism-based convolution neural network to enhance feature extraction capability. A Richer U-Net was proposed to learn more details of roads in [24]. Yang et al. [23] proposed a SDUNet which aggregates both the multi-level features and global prior information of road networks by densely connected blocks. Although the performance of these networks has been successfully validated on many public datasets, the segmentation results are far from perfect due to several factors, such as the complexity of the background, occlusions by cars and trees, changes of illumination and the quality of training datasets.



(a) Original image. (b) Segmentation result.

Fig. 1. Illustration of false positives and broken roads (Images taken from Massachusetts dataset [20]). In (b), green represents true positive (TP), red represents false positive (FP), blue represents false negative (FN).

For example, two of the most common problems in road segmentation are: false positives, such as the red pixels in the orange box of the first row of Fig. 1(b), and road connection problems, such as the broken road in the orange box of the second row of Fig. 1(b). To solve the two problems listed above, a post-

processing algorithm is usually applied to the segmented binary images. In [9], Feng et al. used a heuristic method based on connected-domain analysis to reconstruct the broken roads. Inpainting [4,6] is also a popular way to connect broken roads. Since these methods only consider the output from a previous network, they lack information from the original images. Specifically, false connection is unavoidable. In, Fig. 2, the roads in the two red boxes are disconnected. However, they would likely be falsely connected by a post-processing [17] method without checking the original image.

(a) Original image. (b) Ground truth.

Fig. 2. An example of potential false connections.

In [5], Cheng et al. proposed a cascaded convolutional neural network (CasNet) which consists of two end-to-end networks, one aimed at road segmentation and the other one at centerline extraction. Zhou et al. [26], proposed a universal iteration reinforcement (IterR) model for post-processing which considers both the previous segmentation results and the original images. The IterR model improves the IoU score of segmentation results over 1% in their application. Inspired by these approaches, a two-stage road segmentation method aiming to improve the accuracy and connectivity of roads is proposed in this paper. The first stage is a preliminary segmentation of roads with a selected network (in our case study, ResUnet is used). Then a UNet-like network is applied to enhance the segmentation results by learning from the segmentation behavior of the network in stage one along with the original image.

The main contributions of this research are as follows:

1. A two-stage road segmentation training strategy: the network trained in stage one is used to generate the training samples for stage two. To be specific, when an RGB training sample is fed to the trained network in stage one, a probability map and a weight map are generated. The probability map is attached to the RGB training sample as a 4 dimension input to the network in stage two. The weight map is used for calculating the loss function in stage two.

2. Comprehensive experiments on the Massachusetts dataset [20] to show the proposed method can improve the segmentation results from the first stage to the second stage up to 3% in IoU score. A final IoU of 0.653 and F1-score of 0.788 can

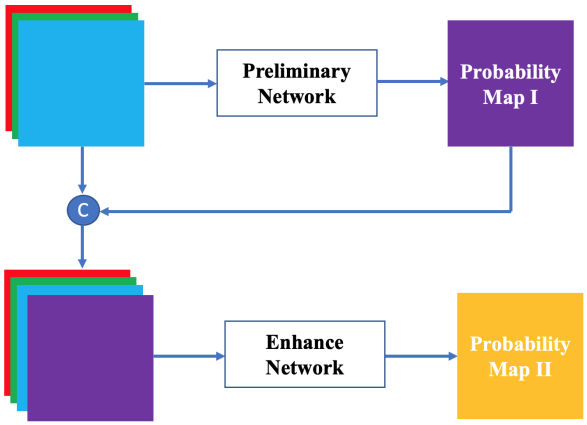


Fig. 3. The diagram of the two-stage segmentation approach.

be reached, which achieves state-of-the-art performance on the Massachusetts dataset.

2 Methodology

The diagram of the two-stage segmentation approach is given in Fig. 3. A probability map is generated by the preliminary network (ResUnet in our case study) in stage one. Then it is attached to the original RGB image before being fed into the enhance network in stage two. Finally we use a threshold (0.5 in our case) to binarize the probability map II as the refined segmentation result.

2.1 Training Sample Generation

The road segmentation task is taken as a supervised learning problem in most deep learning based methods. Usually, the training samples are randomly cropped from high resolution training images (size 1500×1500 in the Massachusetts dataset) and followed by augmentation, such as rotation, flipping and so on. The number of original training images is limited in most applications due to the high cost of manual labeling. However, there are two neural networks in our two-stage segmentation approach, thus it is essential to generate the training samples for each stage properly. Instead of splitting the original training images into two parts, we use the whole training set to generate the training samples (eg. 512×512) for the network in stage one after random cropping and augmentation. After getting the trained network in stage one, another group of training samples are randomly generated in the same way from the same original training images as stage one.

From our experiments, the performance of the ResUnet (or other networks) for stage one is reasonable with the metric IoU of no less than 0.25. Based on this observation, we filter the second group of training samples by removing bad samples for which the trained network in stage one produced an IoU below a threshold T ($T = 0.25$ in our experiments). After the filtering process, we have a probability map for each of the new training samples. By attaching the probability map to each filtered training sample, four-dimensional training samples are generated for stage two. Fig. 4 shows an example of training samples for stage two. The four-dimensional sample is constructed by the RGB channels Fig. 4(a) and the probability map Fig. 4(b).

(a) RGB sample. (b) Probability map. (c) Ground truth.

Fig. 4. An example of training sample for stage two.

2.2 CUnet for Stage Two

The main task of the network in stage two is to remove false positives and connect the broken roads in the preliminary segmentation results. A UNet-like network (CUnet) is applied in stage two. It is just the vanilla UNet with a small change: a skip connection from the fourth dimension (the probability map) to the output is added for learning the residual between ground truth and probability map. The structure of a five layer CUnet is given in Fig. 5, where d_f is the expanded dimension in the first layer. The CUnet tested in our experiments has seven layers with $d_f = 32$.

2.3 Loss Function for CUnet

The binary cross-entropy (BCE) loss function is widely applied in deep learning segmentation [15] tasks. For road segmentation in remote sensing images, considering the imbalance between positive and negative pixels, Feng et al. [9] introduced a categorical balance factor into the BCE, which gives higher weight to negative pixels. In [24], an edge-focused loss function is introduced to guide

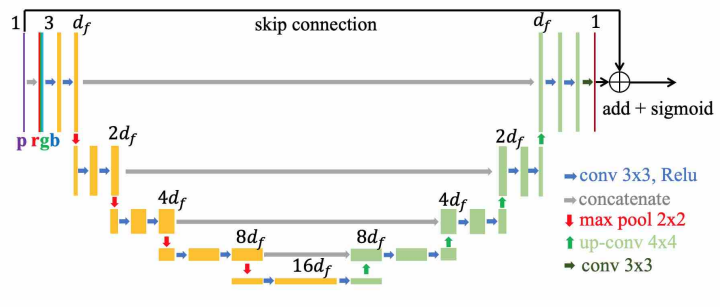


Fig. 5. A five layer CUnet.

the network to pay more attention to the road edge areas by giving the pixels close to edges a higher weight. Inspired by these weighted BCE loss functions, we formulate a weighted BCE loss function by strengthening the attention to the pixels (key pixels) with value larger than a threshold δ in the probability map (the 4th dimension of the input to CUnet). Fig. 6(c - f) shows the weight map generated from the probability map in Fig. 6(b) with $\delta = 0.5, 0.1, 0.05$ and 0.01, respectively. When δ is large (eg. Fig. 6(c)), the disconnection part of a road may not be taken as key pixels. When δ is small (eg. Fig. 6(f)), many false alarms are included as the key pixels. In our experiment, we select $\delta = 0.05$ by trial and error (eg. Fig. 6(e)). As we expect to give more attention to these key pixels, a weight is introduced to the BCE loss function as:

$$L_{wbce} = -\frac{1}{MN}\sum_{i=1}^{MN} d_i[y_i log p_i + (1-y_i)log(1-p_i)] \tag{1}$$

where y_i is the true value of pixel i's category, $p_i \in (0,1)$ is the prediction value for pixel i; M is the number of pixels in one training sample; N is the batch size; d_i is the weight for pixel i:

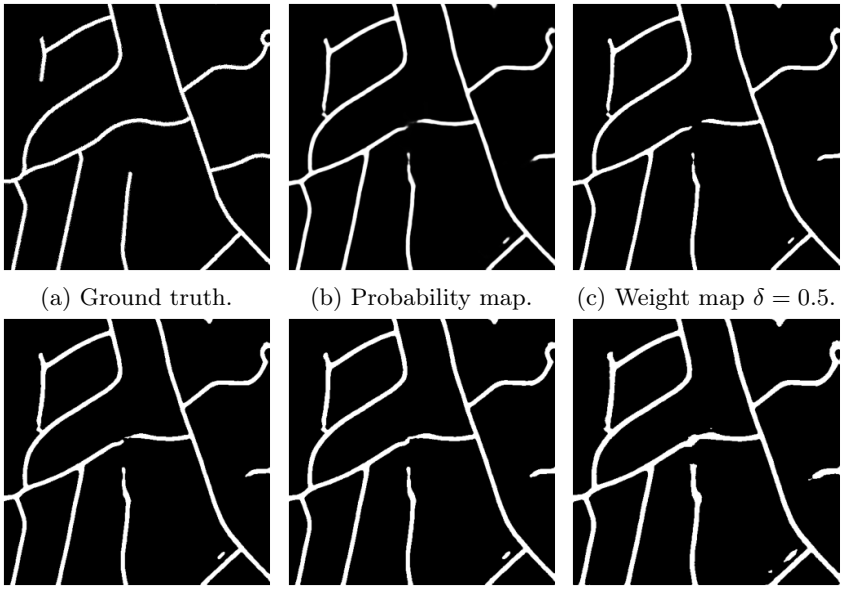
$$d_i = \begin{cases} 1 & \text{if pixel i is not a key pixel} \\ w & \text{if pixel i is a key pixel} \end{cases} \tag{2}$$

$w > 1$ is the weight for the key pixels.

From [27], a joint loss function, which combines BCE loss and the Dice coefficient in eq. (3), has achieved good performance in road segmentation in many datasets.

$$L_{dice} = 1 - \frac{2TP}{2TP + FP + FN} \tag{3}$$

where TP, FP, FN are the number of true positives, false positives and false negatives based on the prediction and ground truth in one batch of samples.



(a) Ground truth. (b) Probability map. (c) Weight map $\delta = 0.5$.

(d) Weight map $\delta = 0.1$. (e) Weight map $\delta = 0.05$. (f) Weight map $\delta = 0.01$.

Fig. 6. An example of key pixels in weight map with different threshold δ.

Consequently, we combine the weighted BCE loss and the Dice coefficient as:

$$L = L_{wbce} + L_{dice} \tag{4}$$

3 Experiments

To verify the effectiveness of our method, comprehensive experiments were conducted on the Massachusetts dataset [20]. A case study based on ResUnet for stage one is performed to select the weight parameter w in eq. (2). Then several popular networks for segmentation such as UNet [22], SegNet [2], ResUnet [25] and D-LinkNet [27], are used for the preliminary segmentation in stage one. We will show the improvements from the CUnet in stage two for each case by quantitative and qualitative analysis. In the end, we also test our method on DeepGlobe dataset [7] to verify the extensiblity of our method.

3.1 Datasets

The Massachusetts dataset [20] is a public dataset created by Mihn and Hinton. The dataset includes 1171 images, each with size 1500×1500 and resolution of 1 m. The 1171 images were split into training set (1108), validation set (14) and test set (49) by the creators. All the networks in our experiments were trained

on the training set. Quantitative evaluation is made on the test set. Qualitative analysis is made on both validation set and test set.

The DeepGlobe road dataset [7] contains 6226 images with labelled road maps, each with size 1024×1024 and resolution of 0.5 m. Following [3, 19], we split the annotated images into 4696 images for training and 1530 images for testing.

Since the training set and test set in the Massachusetts dataset are split by the creators, our main experiments in section 3.3, 3.4 are based on the Massachusetts dataset. The DeepGlobe dataset is randomly split by us, thus we test on it to show the extensibility of our method.

3.2 Experiment Settings

Pytorch was used to implement all the networks in our experiments, running on a workstation with two 24Gb Titan RTX GPUs. The 512×512 training samples were generated by randomly cropping from the original training images, followed by flipping, randomly rotating and changing the brightness. For fair comparison, we created two groups of training samples, for stage one and stage two, separately. All the networks for stage one were trained on the first group of samples. Training samples for CUnet in stage two were generated by the method described in Section 2.1 from the second group of samples.

We set the threshold $\delta = 0.05$ for generating the probability map I in Fig. 3 to include more potential road pixels. The selection of parameter w is discussed in section 3.3.

For the training process, the learning rate is set to 0.0001, batch size is 8. To prevent the networks from overfitting, the training samples were divided into training subset and validation subset with ratio 0.95 and 0.05, respectively. Early stopping was applied once the validation loss stopped decreasing for 10 epochs continuously. The training process was the same for all the networks trained in our experiments.

Evaluation metrics include precision, recall, F1-score, and IoU, which are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{7}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{8}$$

where TP represents true positive, FP represents false positive, FN represents false negative.

3.3 Selection of Parameter w

In the joint loss function (eq. (4)), which combines the weighted BCE loss and the Dice coefficient, w controls the degree of attention we put on the key pixels. Intuitively, if we put too much attention on the key pixels, this may result in false classification for other pixels. Thus we made a sensitivity test for w. In this test, ResUnet is taken as the network for stage one, we set $w = 1, 1.03, 1.05, 1.11$, respectively for training the CUnet in stage two. When $w = 1.0$, every pixel in the training sample gets the same weight, eq. (4) returns to the Dice + BCE loss rather than Dice + weighted BCE. The ResUnet in stage one is trained with the Dice + BCE loss. Table 1 presents the evaluation results on the 49 test images. As we can see, the results from all four CUnets in stage two are much better than the result from the ResUnet in stage one. The best IoU and F1-score are reached when $w = 1.03$. Thus we set $w = 1.03$ for the experiments in the following sections.

Table 1. Test results for different w: S1 means stage one (ResUnet), S2 means stage two (CUnet).

Stage	w	IoU	F1	Prec	Recall
S1	1.0	0.6328	0.7723	0.7783	0.7708
S2	1.0	0.6455	0.7820	**0.7973**	0.7715
	1.03	**0.6530**	**0.7877**	0.7927	**0.7864**
	1.05	0.6529	0.7857	0.7941	0.7847
	1.11	0.6486	0.7834	0.7946	0.7779

Table 2. Comparison experiments on the Massachusetts dataset.

Stage-Network	IoU	F1	Prec	Recall
S1 UNet8	0.6120	0.7565	0.7344	0.7837
S2 CUnet	0.6433	0.7804	0.7913	0.7729
S1 UNet32	0.6407	0.7781	0.7922	0.7686
S2 CUnet	0.6474	0.7832	0.7999	0.7716
S1 SegNet	0.6377	0.7758	0.7834	0.7728
S2 CUnet	0.6476	0.7832	**0.8031**	0.7685
S1 ResUnet	0.6328	0.7723	0.7783	0.7708
S2 CUnet	0.6530	0.7877	0.7927	0.7864
S1 D-LinkNet	0.6263	0.7677	0.7761	0.7621
S2 CUnet	**0.6534**	**0.7880**	0.7901	**0.7885**

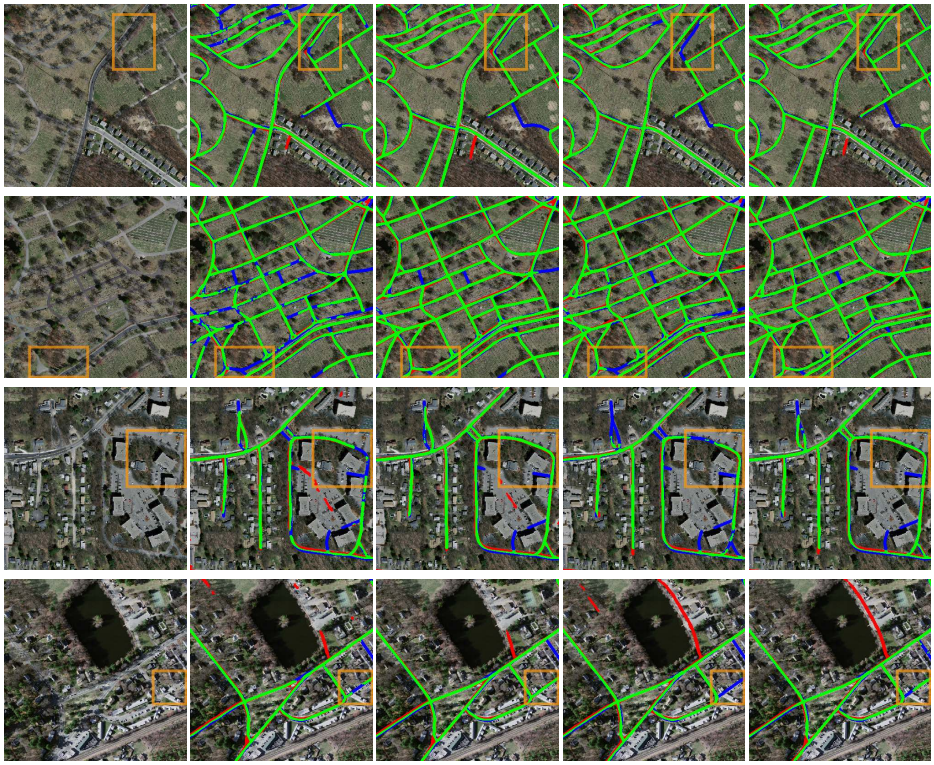


Fig. 7. Segmentation results from the Massachusetts dataset: green represents true positive (TP), red represents false positive (FP), blue represents false negative (FN). (a): Original image; (b): ResUnet; (c): ResUnet+CUnet; (d) D-LinkNet; (e) D-LinkNet+CUnet.

3.4 Test on the Massachusetts Dataset

To further validate the two-stage segmentation method, we used five networks for the preliminary segmentation in stage one, UNet8 (a 7-layer UNet with $d_f = 8$), UNet32, SegNet, ResUnet and D-LinkNet, all trained with Dice + BCE loss. For each case, a CUnet was trained for stage two with $w = 1.03$ in Dice + weighted BCE loss. Table 2 presents the quantitative comparison of the results from stage one and stage two for each case. As shown in Table 2, the IoU and F1-score improved a lot from stage one to stage two for all cases. The highest IoU 0.6534 and F1-score 0.7880 are reached in the case based on D-LinkNet. Compared with the D-LinkNet case, ResUnet case achieved very close IoU 0.6530 and F1-score 0.7877, but lower recall and higher precision.

Since the two-stage approach obtained the highest IoUs in Table 2 with ResUnet and D-LinkNet as the stage-one network, we only compare with their

qualitative results in this section. Fig. 7 gives four examples to show the performance of our method on the broken roads from preliminary segmentations. In the segmentation results, green represents true positive (TP), red represents false positive (FP), blue represents false negative (FN). In the orange box of the first row and the second row of Fig. 7, there is a small lane which is close to the wider avenue and partially occluded by trees. ResUnet and D-LinkNet failed to fully connect the road in the segmentation results in stage one. These broken roads were successfully reconnected in stage two by CUnet. In the third row of Fig. 7, the road is adjacent to a parking lot which has the same color as the road. The performance of ResUnet and D-LinkNet are not satisfactory for this case. Again, the CUnet improved the results a lot for both ResUnet and D-LinkNet. In the last row of Fig. 7, CUnet extracted the whole road in the orange box for the ResUnet case. However, it failed to extract the complete road for the D-LinkNet case. For the ResUnet case, the task is to connect broken lines, but for the D-LinkNet case, the CUnet needs to rediscover the missing road. This demonstrates that the performance of CUnet in stage two is dependent on the output from stage one.

In conclusion, for both ResUnet and D-LinkNet cases, the CUnet can help to enhance the road connections significantly.

3.5 Test on the DeepGlobe Dataset

To verify the extensibility of our method, we test ResUnet + CUnet and D-LinkNet + CUnet methods on the DeepGlobe dataset [7]. Table 3 shows the comparison results between ResUnet + CUnet and D-LinkNet + CUnet. The IoU increases from 0.6364 to 0.6514 for the D-LinkNet + CUnet case. For the case of ResUnet, the IoU is relatively low in stage one, however, the IoU reaches 0.6456 in stage two, which can be attributed to the re-discoverability of the CUnet. Fig. 8 shows four examples from the DeepGlobe dataset. Although the image resolution and road type in the DeepGlobe dataset are different from the Massachusetts dataset, our method shows similar improvement from stage one to stage two.

Table 3. Quantitative results on the DeepGlobe dataset.

Stage-Network	IoU	F1	Prec	Recall
S1 ResUnet	0.5693	0.7117	0.7343	0.7233
S2 CUnet	0.6456	0.7733	0.7635	**0.8085**
S1 D-LinkNet	0.6364	0.7670	0.7644	0.7941
S2 CUnet	**0.6514**	**0.7780**	**0.7758**	0.8040

Fig. 8. Segmentation results from the DeepGlobe dataset: green represents true positive (TP), red represents false positive (FP), blue represents false negative (FN). (a): Original image; (b): ResUnet; (c): ResUnet+CUnet; (d) D-LinkNet; (e) D-LinkNet+CUnet.

4 Conclusions and Perspectives

In this paper, a two-stage segmentation strategy is proposed for road segmentation in remote sensing images. The network in stage one gives preliminary segmentation results. In stage two, a proposed CUnet is applied to enhance the result from stage one. The experimental results on the Massachusetts dataset show that this strategy works for many different CNNs selected in stage one, with the enhanced segmentation results being better than the preliminary results not only in precision, but also in recall. Moreover, the qualitative results show that this strategy can alleviate the broken road problem to some extent. In future work, we plan to apply this two-stage segmentation strategy to other segmentation applications, such as roof segmentation in remote sensing images and blood vessel segmentation [13, 18] in retina fundus images.

Acknowledgements Josiane Zerubia thanks the IEEE SPS Distinguished Lecturer program which enabled her to give several talks in the USA in 2016-2017 and to start this collaboration with Prof. Comer and her PhD student (Tianyu Li) in Purdue.

References

1. Bachagha, N., Wang, X., Luo, L., Li, L., Khatteli, H., Lasaponara, R.: Remote sensing and GIS techniques for reconstructing the military fort system on the Roman boundary (Tunisian section) and identifying archaeological sites. Remote Sensing of Environment **236**, 111418 (2020)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(12), 2481–2495 (2017)
3. Batra, A., Singh, S., Pang, G., Basu, S., Jawahar, C.V., Paluri, M.: Improved road connectivity by joint learning of orientation and segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10377–10385 (2019)
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. p. 417–424. SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., USA (2000)
5. 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 Transactions on Geoscience and Remote Sensing **55**(6), 3322–3337 (2017)
6. Cira, C., Kada, M., Manso-Callejo, M., Alcarria, R., Sanchez, B.B.: Improving road surface area extraction via semantic segmentation with conditional generative learning for deep inpainting operations. ISPRS International Journal of Geo-Information **11**(1) (2022)
7. Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R.: Deepglobe 2018: A challenge to parse the earth through satellite images. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018)
8. Etten, A.V.: City-scale road extraction from satellite imagery v2: Road speeds and travel times. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
9. Feng, D., Shen, X., Xie, Y., Hu, J., Liu, Y.: Efficient occluded road extraction from high-resolution remote sensing imagery. Remote Sensing **13**, 4974 (2021)
10. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition **77**, 354–377 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . pp. 770–778 (2016)
12. Hinz, S., Baumgartner, A.: Automatic extraction of urban road networks from multi-view aerial imagery. ISPRS Journal of Photogrammetry and Remote Sensing **58(1-2)**, 83 – 98 (2003)

13. Hoover, A.D., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Transactions on Medical Imaging **19**(3), 203–210 (March 2000)
14. Hu, J., Razdan, A., Femiani, J., Cui, M., Wonka, P.: Road network extraction and intersection detection from aerial images by tracking road footprints. IEEE Transactions on Geoscience and Remote Sensing **45**(12), 4144–4157 (2007)
15. Jadon, S.: A survey of loss functions for semantic segmentation. In: IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). pp. 1–7 (2020)
16. Li, T., Comer, M., Zerubia, J.: A Connected-Tube MPP Model for Object Detection with Application to Materials and Remotely-Sensed Images. In: IEEE International Conference on Image Processing (ICIP). pp. 1323–1327 (2018)
17. Li, T., Comer, M., Zerubia, J.: Feature extraction and tracking of CNN segmentations for improved road detection from satellite imagery. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 2641–2645 (2019)
18. Li, T., Comer, M., Zerubia, J.: An unsupervised retinal vessel extraction and segmentation method based on a tube marked point process model. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1394–1398 (2020)
19. Lu, X., Zhong, Y., Zheng, Z., Chen, D., Su, Y., Ma, A., Zhang, L.: Cascaded multitask road extraction network for road surface, centerline, and edge extraction. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–14 (2022)
20. Mnih, V.: Machine Learning for Aerial Image Labeling. Ph.D. thesis, University of Toronto (2013)
21. Mnih, V., Hinton, G.E.: Learning to detect roads in high-resolution aerial images. In: European Conference on Computer Vision (ECCV). pp. 210–223. Springer, Berlin, Heidelberg (2010)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241. Springer, Cham (2015)
23. Yang, M., Yuan, Y., Liu, G.: SDUNet: Road extraction via spatial enhanced and densely connected UNet. Pattern Recognition **126**, 108549 (2022)
24. Zao, Y., Shi, Z.: Richer U-Net: Learning more details for road detection in remote sensing images. IEEE Geoscience and Remote Sensing Letters **19**, 1–5 (2022)
25. Zhang, Z., Liu, Q., Wang, Y.: Road extraction by deep residual U-Net. IEEE Geoscience and Remote Sensing Letters **15**(5), 749–753 (May 2018)
26. Zhou, K., Xie, Y., Gao, Z., Miao, F., Zhang, L.: FuNet: A novel road extraction network with fusion of location data and remote sensing imagery. ISPRS International Journal of Geo-Information **10**(1) (2021)
27. Zhou, L., Zhang, C., Wu, M.: D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 192–1924 (2018)
28. Zhou, M., Sui, H., Chen, S., J. Wang, X.C.: BT-RoadNet: A boundary and topologically-aware neural network for road extraction from high-resolution remote sensing imagery. ISPRS Journal of Photogrammetry and Remote Sensing **168**, 288–306 (2020)