



U-Net Based Road Area Guidance for Crosswalks Detection from Remote Sensing Images

Ziyi Chen, Ruixiang Luo, Jonathan Li, Jixiang Du & Cheng Wang

To cite this article: Ziyi Chen, Ruixiang Luo, Jonathan Li, Jixiang Du & Cheng Wang (2021): U-Net Based Road Area Guidance for Crosswalks Detection from Remote Sensing Images, Canadian Journal of Remote Sensing, DOI: [10.1080/07038992.2021.1894915](https://doi.org/10.1080/07038992.2021.1894915)

To link to this article: <https://doi.org/10.1080/07038992.2021.1894915>



Published online: 22 Mar 2021.



Submit your article to this journal [↗](#)



Article views: 14



View related articles [↗](#)




View Crossmark data [↗](#)



U-Net Based Road Area Guidance for Crosswalks Detection from Remote Sensing Images

Extraction des routes potentielles au moyen du modèle U-Net pour la détection des passages piétonniers à partir d'images de télédétection

Ziyi Chen^a, Ruixiang Luo^{a,c}, Jonathan Li^b , Jixiang Du^a, and Cheng Wang^c

^aDepartment of Computer Science and Technology, Fujian Key Laboratory of Big Data Intelligence and Security, Xiamen Key Laboratory of Computer Vision and Pattern Recognition, Huaqiao University, Xiamen, Fujian, China; ^bDepartment of Geography and Environmental Management, University of Waterloo, Waterloo, ON, N2L 3G1, Canada; ^cSchool of Information Science and Engineering, Xiamen University, Xiamen, Fujian, China

ABSTRACT

Due to the wide distribution of crosswalks over the road nets, the finding of impaired crosswalk marks is usually long-time delayed, which may put crosswalk pedestrians into danger. To reduce the repairing cost and improve the finding speed of damaged crosswalks, this paper uses remote sensing images to automatically detect crosswalks. The detection results can be used for further examination of crosswalks. However, the detection of crosswalks from remote sensing images suffers from serious interferences of many other kinds of ground targets. Besides, there are rare openly available datasets for the research of crosswalk detection from remote sensing images. To conquer the above problems, this study provides an openly available dataset for the research of crosswalk detection. To improve the robustness, we propose a crosswalk detection framework which uses a U-Net based road area guidance. First, we use CNN models to detect crosswalks. Then, we use U-Net to extract potential road areas. Third, we propose a mixture classification strategy which combines the detection confidence and potential road area guidance for final crosswalk detection. Experimental results show that the road area guidance for crosswalks' detection is effective and can improve the detection performance.

RÉSUMÉ

En raison de la grande dispersion des passages pour piétons sur les routes, la découverte de marques de passage pour piétons altérées est généralement indument retardée, ce qui peut mettre les piétons en danger. Pour réduire le coût de réparation et accélérer la recherche des passages piétons endommagés, cette étude utilise des images de télédétection pour les détecter automatiquement. Les résultats de cette détection peuvent être utilisés pour un examen plus approfondi des passages. Cependant, la détection des passages pour piétons à partir d'images de télédétection souffre de graves interférences de nombreux autres types de cibles au sol. En outre, il existe de rares ensembles d'images librement disponibles pour la recherche de détection des passages pour piétons. Pour surmonter les problèmes ci-dessus, notre étude fournit un ensemble de données de télédétection librement disponible. Pour améliorer la robustesse, cet étude propose un cadre de détection des passages pour piétons utilisant un guide des zones routières basé sur U-Net. Tout d'abord, nous utilisons des modèles CNN pour détecter les passages. Ensuite, nous utilisons le modèle U-Net pour extraire des zones routières potentielles. Troisièmement, nous proposons une stratégie de classification des mélanges qui combine la confiance de détection et le guide des routes potentielles pour la détection finale des passages pour piétons. Les résultats expérimentaux montrent que l'extraction des routes potentielles pour la détection des passages pour piétons est efficace et peut améliorer les performances de détection.

ARTICLE HISTORY

Received 7 October 2020
Accepted 19 February 2021

Introduction

Crosswalks play an important role in guaranteeing traffic safety and governing traffic order. Owing to erosion and material aging, crosswalks become worn continuously. Discovering and repairing the damage and aging situations of crosswalks in time are important to keep crosswalk pedestrians safe. Traditionally, the discovering of damaged and aging crosswalks need a large amount of examination works by human labors, which is inefficient and costly as crosswalks are widely distributed over all the road nets.

Using remote sensing images to automatically recognize and locate the impaired crosswalks is great meaningful and economical. Before recognizing the impaired crosswalks, it needs to detect the crosswalks firstly. However, due to the complex backgrounds, the detection of crosswalks from remote sensing images is still a challenging task. Especially, the ground has a large amount of similar interferences, which resulting in wrong detections and un-robustness. As shown in Figure 1, when using YoloV3 (Redmon and Farhadi 2018), it occurs a wrong detection which is rather similar to crosswalks beside a building. We find that

the main reason is the lack of road guidance information.

Another crying need for crosswalk marking detection from remote sensing images is a suitable large and well-labeled dataset. As far as we know, there are rare well labeled, and publicly open datasets which focus on crosswalk marking detection from remote sensing images.

To solve the above problems, this paper provides a well labeled and publicly open dataset for crosswalk marking detection from high-resolution optical remote sensing images. Besides, to utilize the road guidance information and improve the detection accuracy, this paper proposes a crosswalks detection method based on Convolutional Neural Network (CNN) models and road guidance information. Firstly, we train CNN models to detect crosswalks. Secondly, we train a U-Net for potential road area extraction. Thirdly, we propose a mixture classification strategy which combines detection confidence and potential road area information guidance to detect crosswalks from remote sensing images. Due to the guidance of potential road areas, our detection framework performs

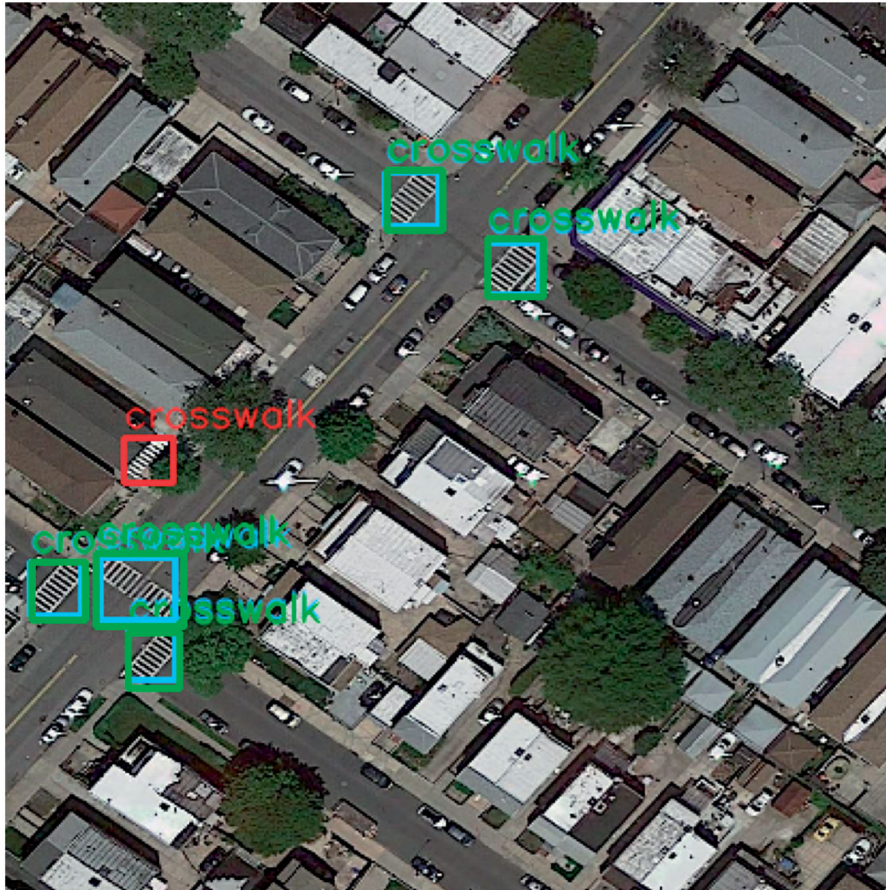


Figure 1. CNN model based detection of crosswalks in a satellite image. An obvious detection error occurs in an area which is similar to crosswalks in front of a building.

better than other compared state-of-the-art CNN detection models, including Faster R-CNN (Ren et al. 2017), YoloV3 (Redmon and Farhadi 2018) and YoloV3 based on DenseNet (Huang et al. 2017). Besides, this paper provides a well labeled and publicly open dataset for crosswalk marking detection from high-resolution optical remote sensing images.

The main contributions of this paper lie on:

1. This paper proposes a framework to detect crosswalks, which combines CNN detection models and road area guidance information. Experimental results prove the effectiveness of our method.
2. This paper supplies a well labeled and publicly open dataset for crosswalk detection from remote sensing images, filling the gap of lacking study dataset for crosswalk marking detection from remote sensing images.

Related work

Object detection from remote sensing images

In 2012, Convolutional Neural Network (CNN) (Krizhevsky et al. 2012) based on deep learning was proposed, which became a mainstream method in computer vision relevant domains owing to its powerful feature learning ability, promoting the rapid development of computer vision. In 2014, Girshick et al. proposed Region Convolutional Neural Network (R-CNN) (Girshick et al. 2014). Since then, object detection has come into the fast lane of development. Various models based on R-CNN began to be proposed to improve object detection model's performance. These models can be divided into two classes: one-stage models and two-stage models. Two-stage object detection models such as Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2017), Mask R-CNN (He et al. 2017) persist in the framework of R-CNN (Girshick et al. 2014), which divides object detection into two steps: classification and regression. one-stage models such as Yolo (Redmon and Farhadi 2018; Redmon et al. 2016; Redmon and Farhadi 2017; Bochkovskiy et al. 2020), SSD (Liu 2016) proposed an unprecedented method to realize classification and regression at the same time.

Remote satellite image is recognized as a significant way to gain geospatial information, which is widely applied in many research areas about computer vision. There are several surveys showing great progress in object detection from remote sensing images (Cheng and Han 2016; Ke Li 2020). However, when

researchers use remote satellite images to realize object detection, they find there appear various problems such as the low resolution of target objects and arbitrary orientations. In order to improve models' performance, researchers have proposed many methods to deal with those problems in object detection from remote sensing images (Zhang et al. 2018; Cheng et al. 2019; Wang et al. 2019; Li et al. 2018; Zhou et al. 2021; Liu et al. 2017; Li et al. 2019; Chen et al. 2020; Zhang et al. 2019; Yao et al. 2019; He et al. 2016; Dong et al. 2020; Chen et al. 2020).

In order to enable models to be rotation-variant, Zhang et al proposed an end-to-end model called Rotated Region Proposal Networks(R²PN) (Zhang et al. 2018) to generate multi-orientated proposals in ship detection applications. Their model could make the inclined ship region proposals more accurate. Chen et al proposed a method to learn Fisher discriminative and rotation-invariant CNN models, which were applied to objects detection from optical remote sensing images, to improve the models' performance(Cheng et al. 2019). They also proposed new object functions to address the problems about class diversity and between-class similarity. Wang et al proposed a model based on CNN network for object detection from remote sensing images (Wang et al. 2019), the model could use CNN's multi-layers to predict different size objects and use the four-point marking method to generate multi-angle Region Proposals. Li et al proposed a novel framework including a local-contextual feature fusion network and region proposal network for object detection from remote sensing images (Li et al. 2018). In their model, they designed a double-channel feature fusion network which can learn contextual and local properties from independent pathways, the final result demonstrated their method is valid. Liu et al (Liu et al. 2017) introduced RR-CNN (rotated region based CNN) for ship detection from remote sensing images. Based on CNN, they proposed three new components to strengthen model's performance in their framework, including a multi-task method for non-maximal suppression (NMS) between different classes, a rotated bounding box regression model, and a rotated region of interest (RRoI) pooling layer.

In order to address the complex background information. Li et al presented a cascade region proposal network with soft-decision non-maximal suppression to improve the network structure, which presented a good performance for airport detection from remote sensing images (Li et al. 2019). The structure used skip-layer feature fusion and hard example mining

methods to improve the model's performance. Chen et al. presented a multi-scale spatial and channel-wise attention (MSCA) mechanism for object detection from remote sensing images (Chen et al. 2020). MSCA not only paid its attention to the foreground but also generated an attention distribution map that combines multi-scale information. The distribution map was applied to the feature map of the deep network. The most significant point of the module is that it can be embedded into any object detection and improve models' efficiency.

In order to realize multi-scale object detection, Zhang et al introduced MS-FF net (Multi-Scale Feature Fusion Network) for object detection from VHR optical remote sensing images (Zhang et al. 2019). There is an additional multi-scale feature fusion layer to fuse the information between detail and semantic features in the MS-FF net. Therefore, it can detect both large and small objects, improving the model's performance. Yan et al introduced a novel Multi-scale Detection Network (MSDN) for object detection from remote sensing images (Yao et al. 2019). The network presented a dilated bottleneck structure to enlarge the receptive field and to improve the regression ability of multi-scale objects with the resolution of deep features maintaining.

Except for dealing with these problems, some researchers devoted themselves to optimizing models' structure. Dong et al proposed a novel high spatial resolution remote sensing images object detection method which uses suitable scale feature (Dong et al. 2020). Scale of feature area is determined by compiling statistics for the scale range of objects in remote sensing images. Cao et al proposed a method to insert deformable layers into the pre-trained networks and fine-tune new networks for object detection from VHR optical images (Cao et al. 2019). Evaluation showed that the one with deformable convolution has better performance. Wei et al introduced a shadow processing algorithm with double threshold random sampling to conduct data preprocessing in remote sensing images (Wei and Zhang 2019). Evaluation showed that the shadow processing algorithm can make object detection models achieve better performance.

Crosswalk marking detection

Crosswalk detection and location is an interesting and important study topic, which is useful for automatic driving (Danilo et al. 2016), keeping pedestrian safe and road guidance (Ahmetovic et al. 2017) etc. Most

existing researches pay their attention to crosswalk detection from street view (Berriel et al. 2017; Christodoulou 2019; Tümen and Ergen 2020) or based on mobile scanning data (Guan et al. 2014; Guan et al. 2014). Berriel et al. used deep learning approach for street view crosswalk classification (Berriel et al. 2017). They used crowdsourcing labeling data to train their model. The experiments proved that the crowdsourcing labeling data is useful. Tümen et al. using VggNet, AlexNet, and LeNet to detect crosswalks from the view of drivers and obtained good performance (Tümen and Ergen 2020). Guan et al. used Mobile Laser Scanning data for road marking detection and achieved good results (Guan et al. 2014). There are a few studies focused on crosswalk detection from the view of remote sensing images (Ahmetovic et al. 2017; Berriel et al. 2017; Prakash et al. 2015), which is most related to our work. Berriel et al. proposed a deep learning based classification method from satellite images (Berriel et al. 2017). They used a crowdsourcing system to enable the automatic acquisition and annotation of a large-scale satellite image database for crosswalks. Then they used the large-scale dataset to train a model to classify whether the image contains crosswalks and achieved good classification results. Prakash et al. proposed a road-following framework for the detection of crosswalk markings from satellite images (Prakash et al. 2015). In their method, they used the road map provided by Open Street Map as crosswalk detection guidance. Based on the road map, they used three steps to detect crosswalks in their crosswalk detection framework. Experiments showed that their method can obtain satisfactory detection results over a large detection area. Crosswalk detection dataset of remote sensing images is also lack of attention although it is an important road marking detection target from remote sensing images. Publicly open and accessible crosswalk detection datasets are rather few. The open-source project "OSM-Crosswalk-Detection: Deep learning based image recognition"¹ and the Swiss OpenStreetMap Association with MapRoulette challenge² are two datasets of crosswalk detection from remote sensing images. However, the above datasets rely on crowdsourcing labeling and other map sources. Another point is that, OSM is essentially a classification dataset as it only can be used for judging whether if an image contains crosswalks. A direct, large enough, and standard crosswalk dataset with good labels is still an urgent requirement.

¹<https://github.com/geometalab/OSM-Crosswalk-Detection>.

²<http://sosm.ch/missing-crosswalks-a-maproulette-challenge/>.

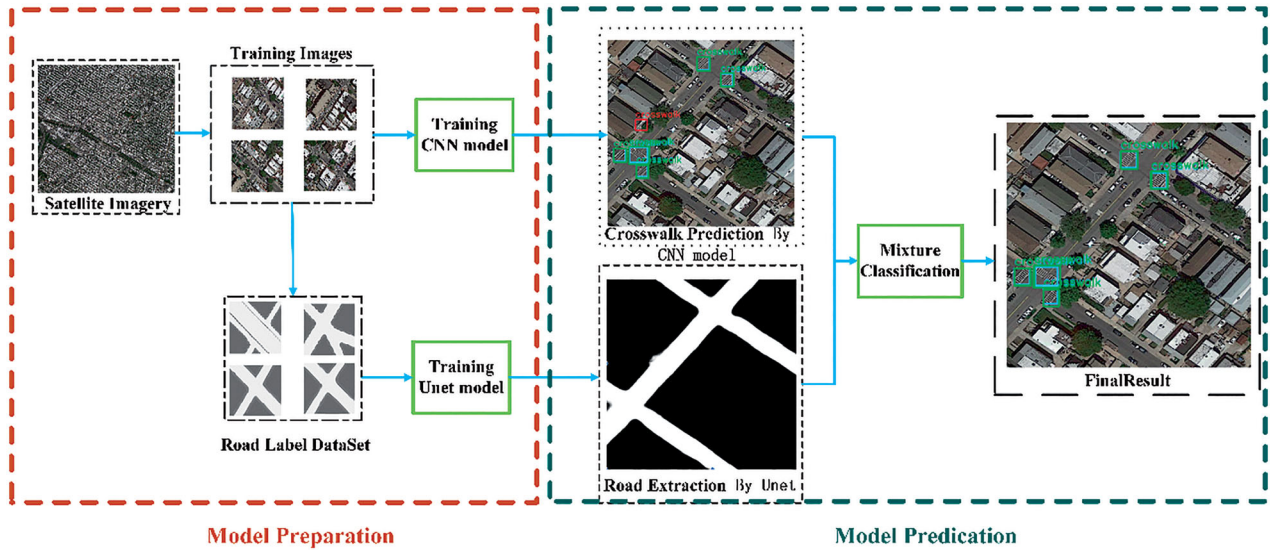


Figure 2. The framework of our method.

Method

In this section, we show the framework of our method firstly. Then, we introduce the object detection CNN models and the road extraction U-Net model used in our approach. Finally, we illustrate the detailed realizations of our method and the analysis about the key parameters.

Framework

As shown in Figure 2, we firstly segment a rather large remote sensing image into small pieces and label crosswalks' locations. Then, with the labeled images, we select a bunch of images as training images to train CNN-based crosswalk detection models. Third, we also label the road areas manually of training images, with which we train a U-Net model to automatically extract potential road areas in test images. Fourth, when testing, we propose a mixture classification strategy which combines the potential road area guidance and the crosswalk detection confidence to detect the crosswalks.

Faster R-CNN, YoloV3, and DenseNet models

In this part, we give a brief introduction to the CNN models we used. Table 1 shows the detailed network structure of YoloV3 (Redmon and Farhadi 2018) used in our method. In YoloV3, the convolution consists of Conv2D Batch Norm, Leaky-ReLu in order, and Residual block proposed in ResNet (He et al. 2016). The output shape is $S^*S^*(3*(5+1))$, where S represents the input image's height and width, 3 presents the number of predicted boxes. In the second bracket,

one position represents the class number, other five positions contain the predictions' coordination and confidence.

In YoloV3, it uses a clustering algorithm to determine prior anchors' size. Then it updates the calculated method on regressive boxes, which makes predicted boxes' centers lock in their cells.

$$\begin{aligned} b_x &= o(t_x) + c_x \\ b_y &= o(t_y) + c_y, \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \quad (1)$$

where b_x , b_y , b_w , and b_h represent the box's center coordinates and its width and height, c_x and c_y represent the offset from the top left corner of the images, p_w and p_h represent the prior boxes' height and width, σ represents sigmoid function, t_x , t_y , t_w , and t_h are learned during training time.

What's more, YoloV3 realizes multi-scale object detection, which yields multi-scale features in convolutional layer and fuses them in predicted phase to make multi-scale objects can be detected.

In our method, we keep loss function of YoloV3 is same with the original, which is composed by coordinate loss, classification loss and confidence loss:

$$\begin{aligned} Loss &= Loss_{coordinate} + Loss_{classification} + Loss_{confidence} \\ Loss_{coordinate} &= \lambda_{coor} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[(b_x - \hat{b}_x)^2 \right. \\ &\quad + (b_y - \hat{b}_y)^2 + (b_w - \hat{b}_w)^2 \\ &\quad \left. + (b_h - \hat{b}_h)^2 \right] \end{aligned}$$

Table 1. The detailed network structure of YoloV3.

YoloV3				
Layer num	Type	Filters	Stride	Output size
1	Convolution	3*3*32	1	416*416*32
2	Convolution	3*3*64	2	208*208*64
3	Residual block	1*1*32	1	208*208*64
		3*3*64		
4	Convolution	3*3*128	2	104*104*128
5	2*Residual block	1*1*64	1	104*104*128
		3*3*128		
6	Convolution	3*3*256	2	52*52*256
7	8*Residual block	1*1*128	1	52*52*256
		3*3*256		
8	Convolution	3*3*512	2	26*26*512
9	8*Residual block	1*1*256	1	26*26*512
		3*3*512		
10	Convolution	3*3*1024	2	13*13*1024
11	4*Residual block	1*1*512	1	13*13*1024
		3*3*1024		
12	Convolutional set	1*1*512	1	13*13*512
		3*3*1024		
		1*1*512		
		3*3*1024		
		1*1*512		
13-1	Convolution	3*3*512	1	13*13*512
13-2	Convolutional	1*1*18	1	13*13*18
14	Upsampling layer 12	2*2		26*26*512
15	Concatenate 14 with layer 9			26*26*1024
16	Convolution set	1*1*256	1	26*26*256
		3*3*512		
		1*1*256		
		3*3*512		
		1*1*256		
17-1	Convolution	3*3*256	1	26*26*256
17-2	Convolutional	1*1*18	1	26*26*18
18	Upsampling layer 16	2*2		52*52*256
19	Concatenate 18 with layer 7			52*52*512
20	Convolution set	1*1*128	1	52*52*128
		3*3*256		
		1*1*128		
		3*3*256		
		1*1*128		
21-1	Convolutional	3*3*128	1	52*52*128
21-2	Conv2d	1*1*18	1	52*52*18

$$Loss_{classification} = \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[-\log(p_c) + \sum_{i=1}^n BCE(\hat{c}_i, c_i) \right]$$

$$Loss_{confidence} = \lambda_{coor} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{i,j}^{noobj} \left[-\log(1 - p_c) \right] \quad (2)$$

$$BCE(\hat{c}_i, c_i) = -\hat{c}_i * \log(c_i) - (1 - \hat{c}_i) * \log(1 - c_i),$$

where S is the grid size, s^2 represents $13*13$, $26*26$ and $52*52$ in size. B is the number of anchor boxes. $1_{i,j}^{obj}$ means the existence of objects, if the anchor contains an object, its value is 1, otherwise its value is 0. $1_{i,j}^{noobj}$ means the absence of objects, if the anchor doesn't contain object its value is 1, or its value is 0. b_x , b_y , b_w and b_h represent the box's center coordinates and its width and height. \hat{b}_x , \hat{b}_y , \hat{b}_w and \hat{b}_h

represent the Ground-Truth's center coordinates and its width and height.

Table 2 shows the detailed network structure of Faster R-CNN (Ren et al. 2017). In Faster R-CNN, the convolution consists of Conv2D and ReLu. The output of the region proposal has 2 vectors: $26*26*18$ is used to evaluate the classification, $26*26*36$ is the coordinates of boxes. Because we add the class "crosswalk" on VGG16 directly, the final output vector's length is 22.

The greatest difference between Faster R-CNN and Fast R-CNN is that Faster R-CNN is not used Select Research to merge super-pixel based on low-level features anymore and replaces it with Region Proposal Network (RPN) which can predict region proposals efficiently. RPN can generate k anchors whose sizes are not exactly same to detect objects in various scales in each sliding position. Through adjusting the anchors in each sliding position make us know the

Table 2. The detailed network structure of Faster R-CNN.

Faster R-CNN				
Layer num	Type	Filters	Stride	Output size
1	Convolution*2	3*3*64	1	416*416*64
2	Max pooling	2*2	2	208*208*64
3	Convolution*2	3*3*128	1	208*208*128
4	Max pooling	2*2	2	104*104*128
5	Convolution*3	3*3*256	1	104*104*256
6	Max pooling	2*2	2	52*52*256
7	Convolution*3	3*3*512	1	52*52*256
8	Max pooling	2*2	2	26*26*256
9	Convolution*3	3*3*512	1	26*26*512
10	Conv2d	3*3*512	1	26*26*512
11-1	Conv2d	1*1*18	1	26*26*18
11-2	Conv2d	1*1*36	1	26*26*36
12-1	Concatenate 11-1 with 11-2			26*26*54
12-2	ROI pooling			49*49
13	Fully connection			4096*1
14	ReLu			4096*1
15	Fully connection for regression			22*4
	Fully connection for classification			22*1

possibility of objects and the coordination of boxes. An advantage of this way is that it can extremely save time in the region proposal step and increase the number of detective objects in an image.

In the training stage, in order to make the model more efficient, authors chose parameter initialization based on the 0-mean standard normal distribution.

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (3)$$

In order to make predicted boxes closer to ground-truth, several parameters are adopted to calculate regressive loss:

$$\begin{aligned} t_x &= \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a} \\ t_w &= \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right) \\ t_x^* &= \frac{x^* - x_a}{w_a}, \quad t_y^* = \frac{y^* - y_a}{h_a} \\ t_w^* &= \log\left(\frac{w^*}{w_a}\right), \quad t_h^* = \log\left(\frac{h^*}{h_a}\right) \end{aligned} \quad (4)$$

where w , h , x , and y represent the box's width and height and its center coordinates, and x , x_a , x^* are variables for predicted box, anchor box and ground-truth box respectively. t_x , t_y , t_w , and t_h are the weight value needed to be calculated.

In our method, we keep loss function of Faster R-CNN is the same with the original, which is composed by classification loss and regression loss:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

Table 3. The detailed network structure of DenseNet-121.

DenseNet-121				
Layer num	Type	Filters	Stride	Output size
1	Convolution	7*7*64	2	208*208
2	Max pooling	3*3	2	104*104
3	Dense block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	1	104*104
4	Transitional layer	1*1*64 conv	1	104*104
		2*2 average pool	2	52*52
5	Dense block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	1	52*52
6	Transitional layer	1*1*64 conv	1	52*52
		2*2 average pool	2	26*26
7	Dense block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	1	52*52
8	Transitional layer	1*1*64 conv	1	26*26
		2*2 average pool	2	13*13
9	Dense block	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	1	13*13

$$L_{cls}(p_i, p_i^*) = -\log[p_i p_i^* + (1 - p_i)(1 - p_i^*)] \quad (5)$$

$$L_{reg}(t_i, t_i^*) = \text{Smooth}_{L1}(t_i - t_i^*) = \begin{cases} 0.5 * x^2 & |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

where p_i is the possibility of predicted classification in i-th anchor, p_i^* is the label of i-th anchor, if the i-th anchor is a positive sample, $p_i^* = 1$, or $p_i^* = 0$. t_i is the parameterized coordinates of predicted Bounding box of i-th anchor, t_i^* is the parameterized coordinates of Ground Truth of i-th anchor. N_{cls} is the size of mini-batch. N_{reg} is the number of anchors.

With the epochs going by, the boxes predict by Faster R-CNN fit ground-truth slightly and output the boxes whose size is similar to the ground-truth boxes.

Table 3 shows the detailed network structure of DenseNet-121 used in our method. In DenseNet, Dense Block is a fully-connected construction, the Convolution layer is composed by BatchNorm-ReLu-Conv.

In DenseNet, it adds residual blocks in the network. Adding residual blocks in the network accelerates the training speed and improves the model's performance. The mathematic expression on residual blocks is:

$$F(x) = H(x) - x \quad (6)$$

where $H(x)$ represents the output of a residual block and x represents the input of a residual block. $F(x)$ represents the residual.

Besides, DenseNet adopts a fully connected structure to recognize all previous output as present input in a dense block. The dense residual blocks not only try their best to preserve feature information but also

Table 4. The detailed network structure of U-Net.

U-Net					
Layer num	Type	Filters	Stride	Output	
1	Conv2d	3*3*64	1	256*256*64	
2	Conv2d	3*3*64	1	256*256*64	
3	Max pooling	2*2		128*128*64	
4	Conv2d	3*3*128	1	128*128*128	
5	Conv2d	3*3*128	1	128*128*128	
6	Max pooling	2*2		64*64*128	
7	Conv2d	3*3*256	1	64*64*256	
8	Conv2d	3*3*256	1	64*64*256	
9	Max pooling	2*2		32*32*256	
10	Conv2d	3*3*512	1	32*32*512	
11	Conv2d	3*3*512	1	32*32*512	
12	Max pooling	2*2		16*16*512	
13	Conv2d	3*3*1024	1	16*16*1024	
14	Conv2d	3*3*1024	1	16*16*1024	
15	Upsampling layer 14	2*2		32*32*1024	
16	Conv2d	2*2*512	1	32*32*512	
17	Concatenate layer 16 with layer 11			32*32*1024	
18	Conv2d	3*3*512	1	32*32*512	
19	Conv2d	3*3*512	1	32*32*512	
20	Upsampling layer 19	2*2		64*64*512	
21	Conv2d	2*2*256	1	64*64*256	
22	Concatenate layer 21 with layer 8			64*64*512	
23	Conv2d	3*3*256	1	64*64*256	
24	Conv2d	3*3*256	1	64*64*256	
25	Upsampling layer 24	2*2		128*128*256	
26	Conv2d	2*2*128	1	128*128*128	
27	Concatenate layer 26 with layer 5			128*128*256	
28	Conv2d	3*3*128	1	128*128*128	
29	Conv2d	3*3*128	1	128*128*128	
30	Upsampling	2*2		256*256*128	
31	Conv2d	2*2*64	1	256*256*64	
32	Concatenate layer 31 with layer 2			256*256*128	
33	Conv2d	3*3*64	1	256*256*64	
34	Conv2d	3*3*64	1	256*256*64	
35	Conv2d	3*3*2	1	256*256*2	
36	Conv2d with Sigmoid	1*1*1	1	256*256*1	

decreases the number of feature parameters, which strengthens the models' performance on classification. The mathematic expression on dense blocks is:

$$x_l = H_l([x_0 \dots x_{l-1}]) \quad (7)$$

where $H_l()$ represents the function to fuse previous layers' output, and x_i represents the i -th layer.

When training YoloV3 based on DenseNet, we used the same loss function with YoloV3. Therefore, we don't mention loss function in this section anymore.

Road area extraction based on U-Net

In this part, we give a brief introduction about U-Net (Ronneberger et al. 2015) used in our approach for road area extraction.

In U-Net, the U-shape structure and skip-connection are the impressive points of the model. During down-sampling, U-Net continuously encodes four times and makes final feature images smaller 16 times than original images. Symmetrically, it will decode feature images four times by fusing previous feature

images, the final feature images will have the same size as the original images.

U-Net adopts the skip-connection structure to concatenate deep and shallow layer information. Skip-connection structure not only improves U-Net's performance but also provides many detailed features for later steps to image segmentation.

When training our U-Net model, we chose Binary Cross-Entropy (BCE) as loss function:

$$\begin{aligned} BCE(x) &= \frac{\sum_{i=1}^C BCE(x)_i}{C} \\ &= \frac{-\sum_{i=1}^C [y_i \log f_i(x) + (1 - y_i) \log(1 - f_i(x))]}{C}, \end{aligned} \quad (8)$$

where C is the number of categories. y_i is the i -th truth label, and $f_i(x)$ is the predicted result on the input x .

Table 4 shows the detailed network structure of U-Net used in our experiments. In the training stage, we need to adjust image size from 416*416 to 256*256 to train. In order to ensure U-Net will not be sensitive

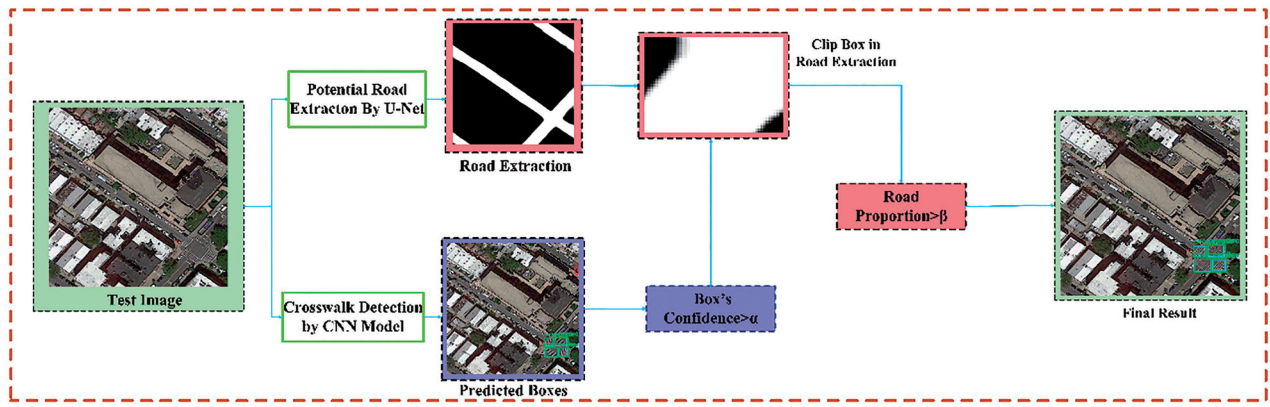


Figure 3. Mixture classification strategy of our approach.

to images' size adjustment. Therefore, we utilize the same method in training and evaluating models.

Mixture classification strategy of our method

Before introducing our method, we illustrate two thresholds adopted in our method.

α : A threshold to filter out the low confidence crosswalk detection boxes. If a box's confidence is greater or equal than the threshold α , we'll combine the road information guidance to judge further. Otherwise, we will filter it out. Usually, the higher α , the lower recall rate.

β : A threshold used in judging whether a predicted box is a crosswalk when combining road guidance produced by U-Net. We calculate the overlap proportion between potential road areas and the predicted crosswalk box. If the overlap ratio is greater or equal than β , we recognize it as a true positive sample. Otherwise, we will filter it out.

In order to use both detection confidence and road information guidance, we propose a mixture classification strategy. As shown in Figure 3, we first detect crosswalks using the trained CNN model and extract potential road areas using the trained U-Net. Then, we evaluate the confidence of detected crosswalk boxes. If the confidence is smaller than the threshold α , the detected box will be dropped. Otherwise, we compute the overlap proportion of detected boxes and with extracted road areas. If the overlap proportion is larger than threshold β , the detected box will be recognized as a crosswalk. Otherwise, the detected box is judged as not a crosswalk. Note that, the U-Net might fail on several images. When the U-Net extracts nothing, we will use α to filter out the samples by boxes' confidence directly.

Results

Dataset

In our experiment, we train and evaluate our models on London suburb dataset with a resolution of 0.15 m. The original image is a large image with a size of 10246×9542 . We use MATLAB to divide the original image into pieces whose size is 416×416 . After clipping, we label the crosswalks for each image. During labeling, we use LabelIMG³ to label crosswalks in all the images. In order to train U-Net, we also label the road as white area and label background as black area for training images.

When we get the original dataset, we find it only has about 2000 samples. we split the original dataset into train dataset, validation dataset and test dataset, the training dataset accounts for 60% of the total images, the validation dataset accounts for 10% of the total images, and the test dataset accounts for the remaining 30%. In order to enlarge our training dataset, we rotate each training image to generate new training images, which is a common method in data augmentation. For each training image, every five-degree rotation generates one more training image. After rotation, crosswalks in the new rotated images need to calculate new coordinates, the calculation equations are as follows:

$$X' = X * \cos(\theta) + Y * \sin(\theta), \quad (9)$$

$$Y' = Y * \cos(\theta) - X * \sin(\theta), \quad (10)$$

where θ presents the rotated angle, X and Y represent the corresponding coordination based on the coordinate of an image center.

³<https://pypi.org/project/labelimg/>



Figure 4. Our crosswalk dataset exhibition. The first row is the original images, and the second row is the corresponding crosswalk labels which are boxes.

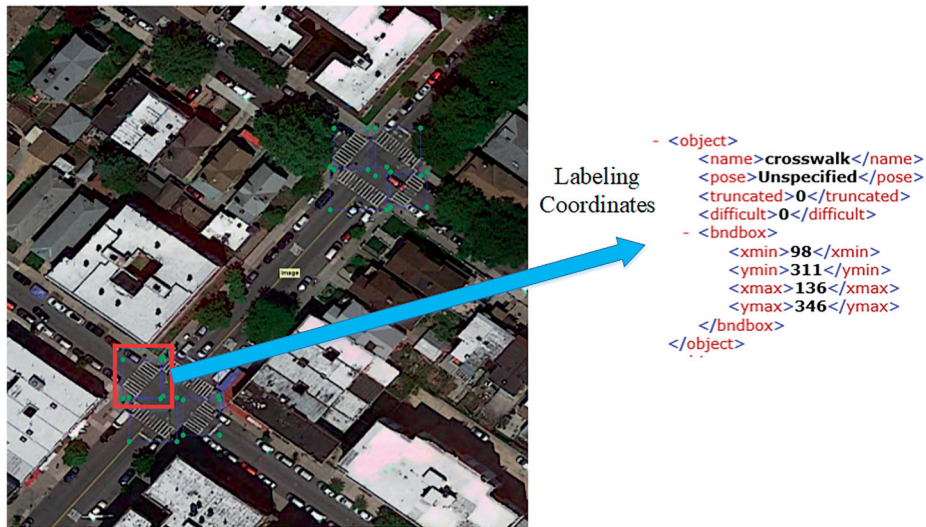


Figure 5. The representation format exhibition of our crosswalks' labeling coordinates saved in the XML file. The crosswalk in the red rectangle of the left image is the target crosswalk, and the right image is the corresponding coordinates saved in the XML file.

After we enlarge the original training dataset, the new training dataset has about 100,000 samples in 29,664 training images.

Our dataset is publicly open and available at the website: ftp://154.85.52.76/Crosswalk_dataset_sub/.

Figure 4 shows three couples of the original images and their corresponding crosswalk labels. The label of each crosswalk includes the coordinates of a box's top left corner and bottom right corner. In our dataset, the labeling information of each crosswalk is saved in an XML file. Figure 5 shows an example of the representation format of the labeled crosswalks'

coordinates. The crosswalk in the red rectangle of the left image is the target crosswalk, and the right image is the corresponding coordinates saved in the XML file.

Experimental set up

In order to evaluate our method, we first train four models: YoloV3, Faster R-CNN, YoloV3 based DenseNet-121, and U-Net. Our experiment runs on two RTX 2080 Ti GPUs. The training images and

labels are same for YoloV3, Faster R-CNN, and YoloV3 based DenseNet-121.

Before we train models, we must choose appropriate parameters for each model. In YoloV3, we set batch size as 30, learning rate as 0.0001, epoch as 50. In YoloV3 based on DenseNet, we set batch size as 18, learning rate as 0.0001, epoch as 50. In Faster R-CNN, we set batch size as 20, learning rate as 0.001, iteration as 120000. In U-Net, we set batch size as 10, learning rate as 0.0001, epoch as 10.

During training phase, Faster R-CNN needs a VGG-16 (Simonyan and Zisserman 2015) pre-trained on ImageNet⁴ for initialization. Other models do not need pre-trained models for initialization. Therefore, the training speed on Faster R-CNN is faster than other object detection models. In order to make full use of our computing resource, we make each batch as large as possible, which is set at 20 in our experiments. When training U-Net, we resize the output size at 416*416, which is same as the output size of CNN crosswalk detection models. Since our training dataset is large enough, all of our models are trained properly.

What's more, it's possible for models to overfit in training. In order to avoid overfitting, we took some measures in training our models. Firstly, we use data augmentation to enlarge our training dataset. Then, we add few other objects in our training dataset. Finally, we make learning rate decay and use early-stopping when validation loss decreases obviously during training.

After training models, we get loss charts of each model. We find that all of models' validation loss are converged. In YoloV3, YoloV3 based on DenseNet, we used TensorBoard to record validation loss, each x-tick represents an epoch. In Faster R-CNN and U-Net, we output the loss into a text file. We average every 100 iterations as one x-tick.

Because of early-stopping, YoloV3 stopped after 46 epochs. YoloV3 based on DenseNet stopped after 49 epochs (Figures 6–9).

When giving a test image, we first obtain the predicted boxes of crosswalks. Then, we extract the potential road areas in the test image. Based on the confidence of predicted crosswalks, we drop the predictions which have lower confidence than the threshold α . Confidence claimed the possibility of objects' appearance, the formula of confidence is as follow:

$$IOU_{predict}^{truth} = \frac{\{Ground - Truth\} \cap \{Predicted\}}{\{Ground - Truth\} \cup \{Predicted\}} \quad (11)$$

⁴<http://www.image-net.org/>.

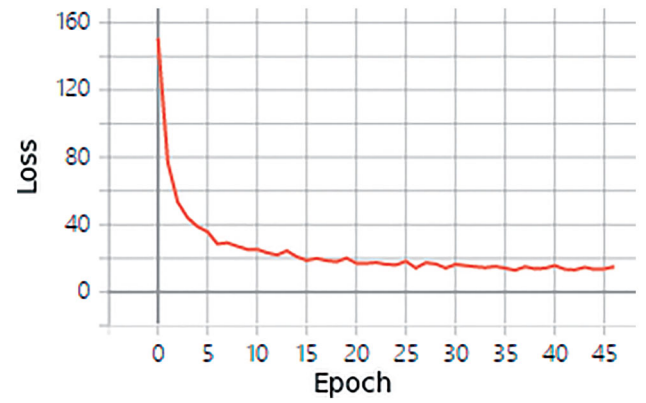


Figure 6. The loss charts of YoloV3 model in our method.

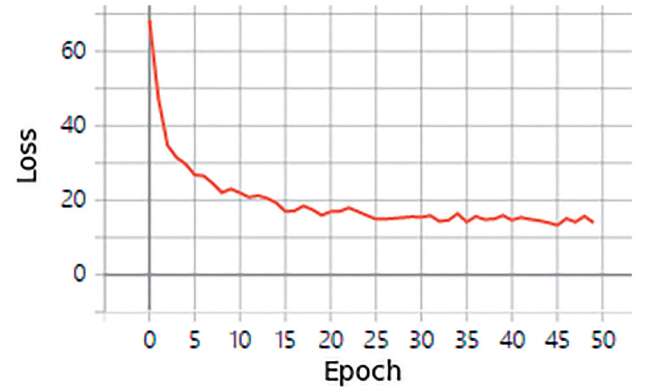


Figure 7. The loss charts of YoloV3 base on DenseNet model in our method.

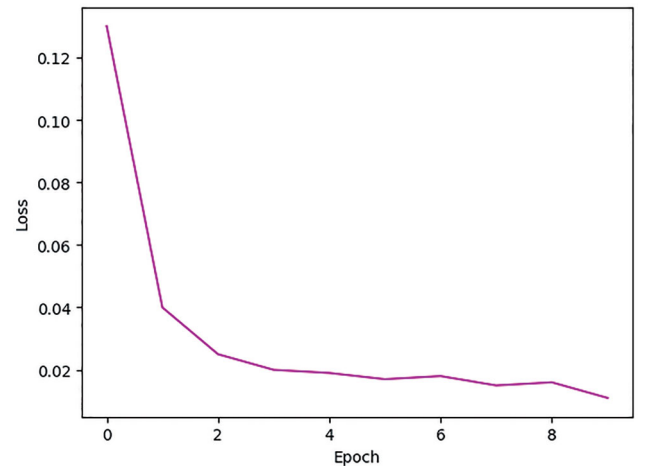


Figure 8. The loss charts of U-Net model in our method.

$$Coincidence = P(Object) * IOU_{predict}^{truth} \quad (12)$$

where $P(x)$ represents the possibility. Intersection Over Union (IOU) is a value to evaluate the proportion of the overlapping area. Next, for each predicted crosswalk with confidence larger than α , we compute the IOU between the predicted bounding box and the

potential road areas. If the computed IOU value is larger than the threshold β , then the prediction is recognized as a crosswalk. Otherwise, we recognize the prediction as not a crosswalk.

Evaluation criteria

Before showing our results, we introduce the performance evaluation criteria used in our method.

Detection accuracy

If the IOU of a predicted box is greater or equal than the threshold which is default as 0.5, we call it True

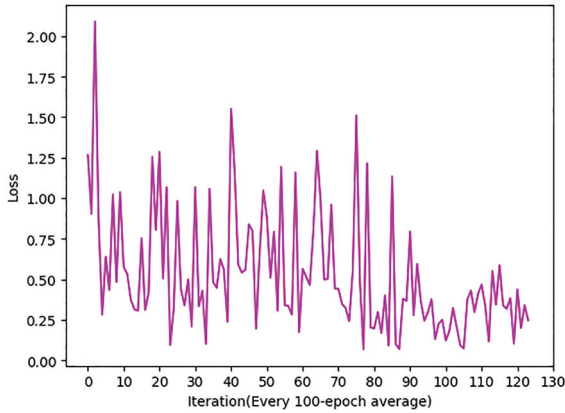


Figure 9. The Loss charts of faster R-CNN model in our method.

Positive (TP), otherwise we call it False Positive (FP). The crosswalks which are lost during detection are called False Negative (FN). Thus, precision, recall, and F1-Score are calculated as follows:

$$\text{Precision} = \frac{FP}{FP + TP}, \quad (13)$$

$$\text{Recall} = \frac{FP}{FP + FN}, \quad (14)$$

$$\text{F1-Score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \quad (15)$$

In order to consider a models' accuracy and recall simultaneously, we can observe the F1-score, which is a comprehensive performance evaluation criterion for a model.

mAP (mean average precision)

It is widely adopted in object detection to evaluate a model's performance. mAP considers a model's accuracy and recall to illustrate a model's performance. However, when calculating mAP, it depends on recall more than precision. The representation of mAP is as follows

$$AP = \sum (r_{n+1} - r_n) * P(r_{n+1}), \quad (16)$$

$$p(r_{n+1}) = \text{MAX}_{m, m \geq r_{n+1}} P(m). \quad (17)$$

Table 5. The experimental results of YoloV3, Faster R-CNN, and YoloV3 based on DenseNet tested on our dataset.

Models	Ground-truth	β -Value	α -Value	False positive	False negative	Recall (%)	Accuracy (%)	mAP (%)	F1-score
YoloV3	871	0.3	0	104	61	93.0	88.62	90.03	0.907563025
			0.1	104	61	93.00	88.62	90.03	0.907563025
			0.2	71	65	92.54	91.90	89.61	0.922196796
			0.3	60	89	89.78	92.87	87.06	0.913018097
			0.4	45	128	85.30	94.29	82.88	0.895720313
			0.5	35	199	77.15	95.05	75.15	0.851711027
			0.6	27	276	66.72	95.66	65.17	0.786108176
			0.7	21	379	56.49	95.91	55.49	0.710982659
			0.8	13	490	43.74	96.70	43.06	0.602371542
			0.9	6	615	29.39	97.71	28.95	0.451897617
FasterR-CNN	871	0.3	0	344	54	93.8	70.37	91.70	0.804133858
			0.1	229	55	93.69	78.09	91.61	0.85177453
			0.2	137	58	93.34	85.58	91.33	0.89291598
			0.3	100	58	93.34	89.05	91.34	0.911434978
			0.4	78	57	93.46	91.26	91.46	0.923425978
			0.5	66	57	93.46	92.50	91.46	0.929754426
			0.6	58	59	93.23	93.33	91.24	0.932797243
			0.7	52	60	93.11	93.97	91.14	0.935409458
			0.8	43	61	93.00	94.96	91.03	0.939675174
			0.9	37	63	92.77	95.62	90.81	0.941724942
DenseNet + YoloV3	871	0.3	0	68	81	90.70	92.07	88.95	0.913823019
			0.1	68	81	90.70	92.07	88.95	0.913823019
			0.2	44	99	88.63	94.61	87.02	0.915234143
			0.3	33	127	85.42	95.75	83.86	0.902912621
			0.4	26	162	81.40	96.46	80.09	0.882938979
			0.5	22	227	73.94	96.70	72.88	0.837996096
			0.6	15	296	66.02	97.46	65.17	0.787132101
			0.7	9	346	60.28	98.31	59.56	0.747330961
			0.8	6	425	51.21	98.67	50.55	0.674225246
			0.9	6	550	36.85	98.17	36.36	0.535893155

The β value is fixed at 0.3 and the α changes ranging from 0 to 0.9.

The results with bold format represent the best results at the corresponding evaluations.

Table 6. The experimental results of YoloV3, Faster R-CNN, and YoloV3 based on DenseNet were tested on our dataset.

Models	Ground-truth	a-value	β -Value	False positive	False negative	Recall (%)	Accuracy (%)	mAP (%)	F1-score
YoloV3	871	0.3	0	70	84	90.36	91.83	87.04	0.91087963
			0.1	61	87	90.01	92.78	87.27	0.913752914
			0.2	60	88	89.90	92.88	87.18	0.913652275
			0.3	60	89	89.78	92.87	87.06	0.913018097
			0.4	59	90	89.67	92.98	86.97	0.912916423
			0.5	59	93	89.32	92.95	86.62	0.911007026
			0.6	56	93	89.32	93.29	86.84	0.912609971
			0.7	55	96	88.98	93.37	86.59	0.911228689
			0.8	51	102	88.29	93.78	85.93	0.909520993
			0.9	51	109	87.49	93.73	85.19	0.904988124
FasterR-CNN	871	0.3	0	142	47	94.60	85.30	91.67	0.897114861
			0.1	103	54	93.80	88.80	91.75	0.912339475
			0.2	101	54	93.80	89.00	91.81	0.913359419
			0.3	100	58	93.34	89.05	91.34	0.911434978
			0.4	98	59	93.23	89.23	91.23	0.911847277
			0.5	98	60	93.11	89.22	91.12	0.911235955
			0.6	96	62	92.88	89.39	90.89	0.911036036
			0.7	96	69	92.08	89.31	90.10	0.906726964
			0.8	92	77	91.16	89.62	89.29	0.903813318
			0.9	90	86	90.13	89.71	88.25	0.899198167
DenseNet + YoloV3	871	0.3	0	38	125	85.65	95.15	84.02	0.901510574
			0.1	32	126	85.53	95.88	84.08	0.904126214
			0.2	32	126	85.53	95.88	84.08	0.904126214
			0.3	33	127	85.42	95.75	83.96	0.902912621
			0.4	33	128	85.30	95.75	83.84	0.902246509
			0.5	33	129	85.19	95.74	83.73	0.901579587
			0.6	33	130	85.07	95.74	83.76	0.900911854
			0.7	31	132	84.85	95.97	83.55	0.900670323
			0.8	30	134	84.62	96.09	83.32	0.8998779
			0.9	29	145	83.35	96.16	82.09	0.89298893

The α value is fixed at 0.3 and the β changes ranging from 0 to 0.9.

The results with bold format represent the best results at the corresponding evaluations.

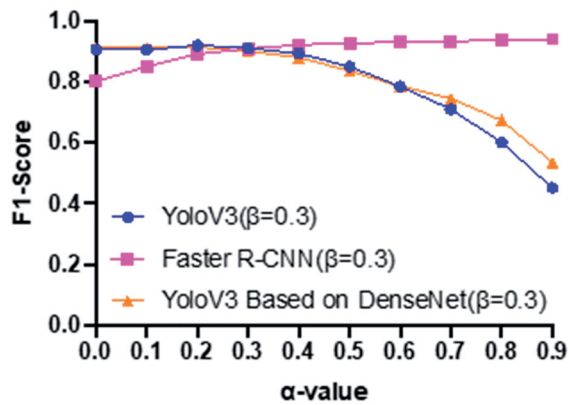


Figure 10. The influences of α on F1-Scores for three models.

Experimental results

Analysis about α -value

In this section, we analyze the influence of threshold α . We fix the β value at 0.3 and change the α ranging from 0 to 0.9. Table 5 shows the detailed experimental results tested on our dataset. Figure 10 shows the influences of α on models' F1-Score for three models. From Table 5, we can see that the F1-scores increase when increasing the threshold of α for YoloV3 and Faster R-CNN. For YoloV3 based on DenseNet, the best α is 0.3.

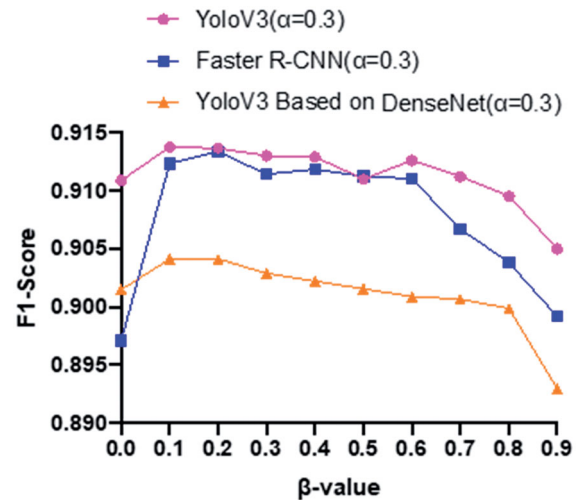


Figure 11. The influences of β on F1-Scores for three models.

Analysis about β -value

In this section, we analyze the influence of β for YoloV3, Faster R-CNN, and YoloV3 based on DenseNet tested on our dataset. During the experiment, we fix the α at 0.3. Then, we change the β ranging from 0 to 0.9. Table 6 shows the experimental results. Figure 11 shows the F1-scores about the experiments for the three models.

From Table 6, we can see that the performance is not continuously improved when β increases.

Table 7. The final comparison results tested on our dataset among the original YoloV3, Faster R-CNN, YoloV3 based on DenseNet and our framework combined with the above three CNN models, respectively.

Models	Ground-truth number	False positive	False negative	Recall (%)	Accuracy (%)	mAP (%)	F1-score
YoloV3	871	130	56	93.57	86.24	89.96	0.897577093
Road guidance combined with YoloV3($\alpha=0.3, \beta=0.1$)		61	87	90.01	92.78	87.27	0.913752914
Faster R-CNN	871	483	41	95.29	63.21	92.19	0.76007326
Road guidance combined with faster R-CNN($\alpha=0.3, \beta=0.2$)		101	54	93.80	89.00	91.81	0.913359419
YoloV3 based on DenseNet	871	77	78	91.04	91.15	89.08	0.910970706
Road guidance combined with YoloV3 base on DenseNet($\alpha=0.2, \beta=0.3$)		44	99	88.63	94.61	87.02	0.91522422

The results with bold format represent the best results at the corresponding evaluations.

A suitable β selection is important to our method. Besides, we find that we cannot impose strong limitations of the road information guidance for the final recognition. Otherwise, many true crosswalks will be recognized as false negatives. The reason may be that the road extraction by U-Net fails in several road area extractions.

Comparison with the original three CNN models

From the above analysis, we finally select the parameters $\alpha=0.3$ and $\beta=0.1$ for YoloV3, $\alpha=0.3$ and $\beta=0.2$ for Faster R-CNN and $\alpha=0.2$ and $\beta=0.3$ for YoloV3 based on DenseNet respectively. Table 7 shows the comparison results tested on our dataset among YoloV3, Faster R-CNN, YoloV3 based on DenseNet and road guidance combined with the above three CNN models respectively. From Table 7, we can observe that when combined with road guidance, the YoloV3, Faster R-CNN, and DenseNet based Yolov3 can improve their F1-score performance by about 1.6, 15.3, and 0.4%, respectively. The experimental results prove that the road guidance information is useful for crosswalk detection from remote sensing images.

Figure 12 shows the effectiveness of U-Net on our dataset. Row 1 and 3 show the original pictures, row 2 and 4 show the extraction results by U-Net. Owing to the effectiveness of road potential areas extraction by U-Net, the road information guidance for crosswalk detection presents satisfactory improvements in our experiments.

When only using YoloV3, Faster R-CNN, or YoloV3 Based on DenseNet, the F1-scores in the experiment are 0.8976, 0.7600, and 0.9109, respectively. As a contrast, when combing the above three models with potential road area guidance, the F1-scores are improved to 0.9137, 0.9133, and 0.9152, respectively. Through this experiment, it convincingly proves that the Mixture Classification Strategy is effective.

Conclusion

In this paper, we proposed a road information guidance based crosswalk detection method from remote sensing images. We first used CNN models to detect crosswalks. In our method, we used YoloV3, Faster R-CNN and DenseNet based YoloV3 for detection. Then, we dropped detections with low confidence. Third, we used U-Net to extract potential road areas. Based on road area guidance, we further dropped detections which were not located on road areas. To

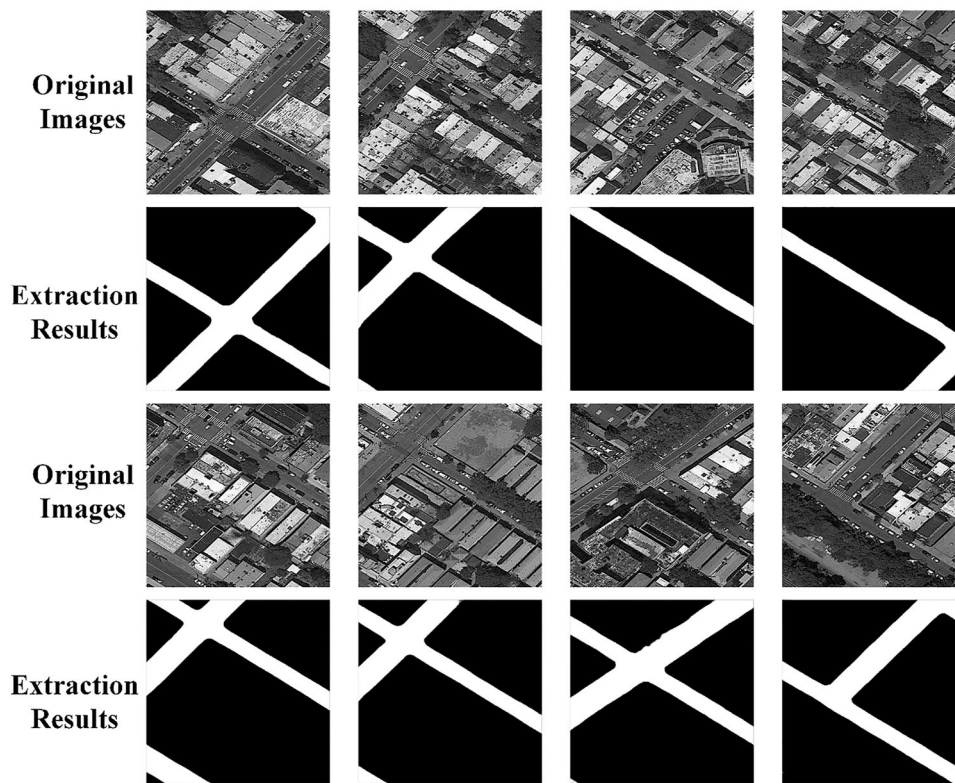


Figure 12. Several road extractions results of U-Net on our dataset.

test our method, we made a crosswalk detection dataset from remote sensing images. The dataset was publicly accessible. We tested the original YoloV3, Faster R-CNN and DenseNet based YoloV3 on our dataset. We also tested the YoloV3, Faster R-CNN, and DenseNet based YoloV3 combined with road information guidance respectively. The experimental results proved that combing with road information guidance, the YoloV3, Faster R-CNN, and DenseNet based YoloV3 can improve F1-scores' performance about 1.6, 15.3, and 0.4%, respectively. The experimental results proved the effectiveness of our method.

In our future study, several aspects of potential improvements are going to be tried in our work. Firstly, we will change our method into an end-to-end training framework. Secondly, we will combine more models with our method to improve efficiency. Thirdly, we will try to rotate the predicted boxes and ground truth, making the predicted boxes closer to crosswalks.

Funding

This study was financially supported by Natural Science Foundation of Fujian Province [No.2019J01081], National Natural Science Foundation of China [No. 62001175],

United National Natural Science Foundation of China [No.U1605254], National Natural Science Foundation of China [No. 6187606, 61972167 and 61673186], and the Special National Key Research and Development Plan [No. 2019YFC1604705].

ORCID

Jonathan Li  <http://orcid.org/0000-0001-7899-0049>

References

- Ahmetovic, D., Manduchi, R., Coughlan, J.M., and Mascetti, S. 2017. "Mind your crossings: Mining GIS imagery for crosswalk localization." *ACM Transactions on Accessible Computing*, Vol. 9(No. 4): pp. 1–25. doi:10.1145/3046790.
- Berriel, R.F., Lopes, A.T., Souza, A.F.D., and Oliveira-Santos, T. 2017. "Deep learning based large-scale automatic satellite crosswalk classification." *IEEE Geoscience and Remote Sensing Letters*, Vol. 14(No. 9): pp. 1513–1517. doi:10.1109/LGRS.2017.2719863.
- Berriel, R.F., Rossi, F.S., de Souza, A.F., and Oliveira-Santos, T. 2017. "Automatic large-scale data acquisition via crowdsourcing for crosswalk classification: A deep learning approach." *Computers & Graphics*, Vol. 68: pp. 32–42. doi:10.1016/j.cag.2017.08.004.
- Bochkovskiy A., Wang C.-Y., and Liao H.-Y.M. 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv:2004.10934

- Cao, Z., Li, X., and Zhao, L. 2019. "Object detection in VHR image using transfer learning with deformable convolution." Paper presented at the IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, July 2019.
- Chen, Z., Fan, W., Zhong, B., Li, J., Du, J., and Wang, C. 2020. "Coarse-to-fine road extraction based on local Dirichlet mixture models and multiscale-high-order deep learning." *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21(No. 10): pp. 4283–4293. doi:10.1109/TITS.2019.2939536.
- Chen, J., Wan, L., Zhu, J., Xu, G., and Deng, M. 2020. "Multi-scale spatial and channel-wise attention for improving object detection in remote sensing imagery." *IEEE Geoscience and Remote Sensing Letters*, Vol. 17(No. 4): pp. 681–685. doi:10.1109/LGRS.2019.2930462.
- Cheng, G., Han, J., Zhou, P., and Xu, D. 2019. "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection." *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, Vol. 28(No. 1): pp. 265–278. doi:10.1109/TIP.2018.2867198.
- Christodoulou, P. 2019. *Crosswalk identification for decision making*. Thessaloniki, Greece: International Hellenic University.
- Danilo, C.H., Laksono, K., Alexander, F., and Kang, J. 2016. "Real-time lane region detection using a combination of geometrical and image features." *Sensors*, Vol. 16(No. 11): pp. 1935. doi:10.3390/s16111935.
- Dong, Z., Wang, M., Wang, Y., Zhu, Y., and Zhang, Z. 2020. "Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features." *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 58(No. 3): pp. 2104–2114. doi:10.1109/TGRS.2019.2953119.
- Girshick, R. 2015. "Fast R-CNN." Paper presented at the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, December 2015.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation." Paper presented at the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, June 2014.
- Cheng, G., and Han, J. 2016. "A survey on object detection in optical remote sensing images." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 117: pp. 11–28. doi:10.1016/j.isprsjprs.2016.03.014.
- Guan, H., Li, J., Yu, Y., Chapman, M., and Wang, C. 2014. "Automated road information extraction from mobile laser scanning data." *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16(No. 1): pp. 194–205. doi:10.1109/TITS.2014.2328589.
- Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., and Yang, B. 2014. "Using mobile laser scanning data for automated extraction of road markings." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 87: pp. 93–107. doi:10.1016/j.isprsjprs.2013.11.005.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. 2017. "Mask R-CNN." Paper presented at the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 2017.
- He, K., Zhang, X., Ren, S., and Sun, J. 2016. "Deep Residual Learning for Image Recognition." Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016.
- Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. 2017. "Densely Connected Convolutional Networks." Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 2017.
- Li, K., Wan, G., Cheng, G., Meng, L., and Han, J. 2020. "Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 159: pp. 296–307. doi:10.1016/j.isprsjprs.2019.11.023.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. "ImageNet classification with deep convolutional neural networks." Paper presented at the NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems, New York, U.S., December 2012.
- Li, K., Cheng, G., Bu, S., and You, X. 2018. "Rotation-insensitive and context-augmented object detection in remote sensing images." *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 56(No. 4): pp. 2337–2348. doi:10.1109/TGRS.2017.2778300.
- Li, S., Xu, Y., Zhu, M., Ma, S., and Tang, H. 2019. "Remote sensing airport detection based on end-to-end deep transferable convolutional neural networks." *IEEE Geoscience and Remote Sensing Letters*, Vol. 16(No. 10): pp. 1640–1644. doi:10.1109/LGRS.2019.2904076.
- Liu, W. 2016. "SSD: single shot multibox detector." Paper presented at the European Conference on Computer Vision, Amsterdam, The Netherlands, October 2016.
- Liu, Z., Hu, J., Weng, L., and Yang, Y. 2017. "Rotated region based CNN for ship detection." Paper presented at the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, September 2017.
- Prakash, T., Comandur, B., Chang, T., Elfiky, N., and Kak, A. 2015. "A generic road-following framework for detecting markings and objects in satellite imagery." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 8(No. 10): pp. 4729–4741. doi:10.1109/JSTARS.2015.2495142.
- Redmon, J., and Farhadi, A. 2017. "YOLO9000: better, faster, stronger." Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 2017.
- Redmon J. and Farhadi A. 2018. "YOLOv3: An Incremental Improvement," *arXiv* (e-prints). arXiv:1804.02767
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. 2016. "You only look once: Unified, real-time object detection." Paper presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016.
- Ren, S., He, K., Girshick, R., and Sun, J. 2017. "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39(No. 6): pp. 1137–1149. doi:10.1109/TPAMI.2016.2577031.
- Ronneberger, O., Fischer, P., and Brox, T. 2015. "U-net: Convolutional networks for biomedical image segmentation." Paper presented at the International

- Conference on Medical Image Computing and Computer-assisted Intervention, Munich, Germany, October 2015.
- Simonyan K. and Zisserman A. 2015. "Very deep convolutional networks for large-scale image recognition." Paper presented at the ICLR2015, San Diego, CA, USA, May 2015.
- Tümen, V., and Ergen, B. 2020. "Intersections and cross-walk detection using deep learning and image processing techniques." *Physica A: Statistical Mechanics and Its Applications*, Vol. 543: pp. 123510. doi:[10.1016/j.physa.2019.123510](https://doi.org/10.1016/j.physa.2019.123510).
- Wang H., Liang W., and Shan G. 2019. "An efficient method of detection and recognition in remote sensing image based on multi-angle region of interests." *arXiv* (preprint arXiv:1907.09320).
- Wei, W., and Zhang, J. 2019. "Remote sensing image aircraft detection technology based on deep learning." Paper presented at the 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, August 2019.
- Yao, Q., Hu, X., and Lei, H. 2019. "Geospatial object detection in remote sensing images based on multi-scale convolutional neural networks." Paper presented at the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, July 2019.
- Zhang, W., Jiao, L., Liu, X., and Liu, J. 2019. "Multi-scale feature fusion network for object detection in VHR optical remote sensing images." Paper presented at the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, July 2019.
- Zhang, Z., Guo, W., Zhu, S., and Yu, W. 2018. "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks." *IEEE Geoscience and Remote Sensing Letters*, Vol. 15(No. 11): pp. 1745–1749. doi:[10.1109/LGRS.2018.2856921](https://doi.org/10.1109/LGRS.2018.2856921).
- Zhou, K., Zhang, Z., Gao, C., and Liu, J. 2021. "Rotated feature network for multiorientation object detection of remote-sensing images." *IEEE Geoscience and Remote Sensing Letters*, Vol. 18(No. 1): pp. 33–37. doi:[10.1109/LGRS.2020.2965629](https://doi.org/10.1109/LGRS.2020.2965629).