

Road Extraction With Satellite Images and Partial Road Maps

Qianxiong Xu[✉], Cheng Long[✉], Senior Member, IEEE, Liang Yu, and Chen Zhang, Member, IEEE

Abstract—Road extraction is a process of automatically generating road maps mainly from satellite images. Existing models all target to generate roads from the scratch despite that a large number of road maps, though incomplete, are publicly available [e.g., those from OpenStreetMap (OSM)] and can help with road extraction. In this article, we propose to conduct road extraction based on satellite images and partial road maps, which is new. We then propose a two-branch partial-to-complete network (P2CNet) for the task, which has two prominent components: gated self-attention module (GSAM) and missing part (MP) loss. GSAM leverages a channelwise self-attention module and a gate module to capture long-range semantics, filter out useless information, and better fuse the features from two branches. An MP loss is derived from the partial road maps, trying to give more attention to the road pixels that do not exist in partial road maps. Extensive experiments are conducted to demonstrate the effectiveness of our model, e.g., P2CNet achieves the state-of-the-art performance with the intersection over union (IoU) scores of 70.71% and 75.52%, respectively, on the SpaceNet and OSM datasets.

Index Terms—Attention mechanism, data fusion, partial road maps, remote sensing, road extraction, satellite images, semantic segmentation.

I. INTRODUCTION

ROAD maps are widely used in many applications, including navigation, intelligent transportation, location-based services, and urban design. Conventionally, road maps are made manually via field surveys, which are costly and labor-intensive, and thus, they could not be updated timely. In the last decade, extensive studies have been conducted for automatically extracting roads mainly from satellite images, which is called road extraction [1], [2], [3], [4], [5], [6]. Compared with conventional approaches, road extraction is significantly

Manuscript received 18 November 2022; revised 15 February 2023; accepted 20 March 2023. Date of publication 27 March 2023; date of current version 7 April 2023. This work was supported in part by the Ministry of Education of Singapore through the Academic Research Fund under Tier 2 Award MOE-T2EP20221-0013; and in part by the Alibaba Group through the Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore, under Award AN-GC-2020-006. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore. (*Corresponding author: Cheng Long*)

Qianxiong Xu and Cheng Long are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: qianxiong001@e.ntu.edu.sg; c.long@ntu.edu.sg).

Liang Yu is with Alibaba Cloud Computing, Alibaba, Hangzhou 310030, China (e-mail: liangyu.yl@alibaba-inc.com).

Chen Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: c4zhang@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TGRS.2023.3261332

1558-0644 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

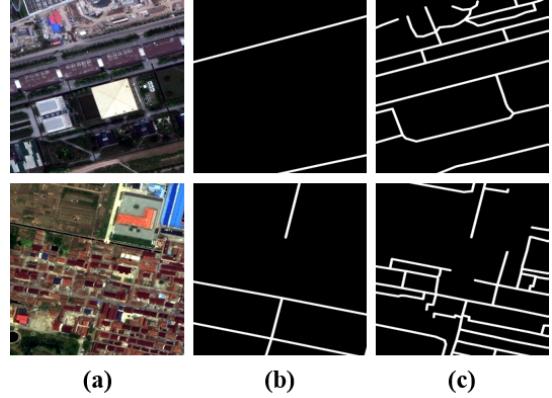


Fig. 1. Data samples. (a) Satellite image. (b) Road map (OSM). (c) Road map (SpaceNet).

cheaper with little or no manual efforts and can be conducted frequently to reflect the updates such as newly built roads and/or destroyed roads timely.

Existing studies on road extraction all aim to conduct the task from an empty road map, i.e., they generate road maps from no road maps but satellite images only. While targeting this problem setting would reveal some scientific insights, e.g., how feasible it is to extract road maps from satellite images solely, it assumes a problem setting that is more difficult than necessary. In reality, a large quantity of road map data, such as OpenStreetMap¹ (OSM), though not complete, has already been collected and made publicly available. For example, in Shanghai, the road maps available at OSM are about 54.19% of those available at SpaceNet. Two samples of satellite images, partial road maps and complete road maps, are shown in Fig. 1. These partial road maps can greatly help with the road extraction task in the following aspects. First, they are already parts of the ground-truth road maps, and thus, there is no need to extract them again. Instead, we can focus more on the remaining parts. Second, the satellite features of these available road maps can directly be obtained, with which the road extraction problem can be converted to one of finding features similar to these available features, which is much easier.

Therefore, we propose to conduct road extraction based on satellite images and partial road maps. There are mainly three possible options of utilizing partial road maps.

- 1) *Outputs Fusion*: We conduct road extraction without the partial road maps with an existing method, fuse

¹<https://www.openstreetmap.org/>

the late-layer features with the partial road maps, and then use extra convolution layers to process the fused information. With this approach, the signal provided by the partial road maps would arrive a bit too late since it provides no guidance on the task of extracting roads from the satellite data, and therefore, it does not help to extract those roads that are missing.

- 2) *Inputs Fusion*: We convert the partial road maps to images, fuse them and the satellite images as different channels, and then input the fused images to an existing method for extracting roads. Nevertheless, the model trained with this fused input would tend to reserve the roads in the partial road maps, but not to effectively extract those missing roads since the partial road maps are parts of the ground truth, i.e., the ability of extracting features would be weakened.
- 3) *Features Fusion*: We extract features from both the satellite images and the partial road maps, fuse the extracted features, and then generate the road map from the fused features. This approach would avoid the issues of the first two options, i.e., the signal from partial road map comes early enough and it would not affect the extraction of satellite features directly, yet how to instantiate this design idea remains nontrivial.

In this article, we design a features fusion approach. Specifically, we introduce a two-branch deep learning framework called partial-to-complete network (P2CNet), where one branch takes satellite images as inputs and the other branch takes partial road maps as inputs. Each branch involves an encoder and a decoder, and for both of them, we employ DeepLabV3+ [7] with the backbone of ResNet-34 as the basic model. We develop modules for better utilizing and fusing the features outputted by the encoders. Specifically, a gated self-attention module (GSAM) is designed, which involves channelwise attentions to better capture the long-dependency semantics from two branches and gate mechanisms to fuse the features from two branches in a complementary manner. Furthermore, apart from using the conventional binary cross entropy (BCE) loss and Dice loss, we introduce a missing part (MP) loss to further exploit the partial road maps by putting more attention on those road pixels that are missing in the partial road maps. With the MP loss, the roads that are missing would be better generated.

Extensive comparative experiments on models, fusion strategies, MP loss strategies, datasets, and parameter settings are conducted and reported. As our setting is new, we train some state-of-the-art models of semantic segmentation both in road extraction and road completion settings to provide better comparisons. The results show that P2CNet achieves the best results among others, as explained in the following. First, the performance improvements of our method over the best baseline method [in terms of the commonly used metric intersection over union (IoU)] is 1.25% (0.7552 versus 0.7427) on the OSM dataset and 0.52% (0.7071 versus 0.7019) on the SpaceNet dataset. In addition, the improvement is 1.85% (0.8028 versus 0.7843) on the OSM dataset for a topology-oriented metric APLS. Second, we emphasize that our method achieves the best results consistently across six

metrics (including the topology-oriented one APLS), on two datasets of different qualities (OSM and SpaceNet), and among 11 methods. Third, our method is robust against different settings of partial road maps. For example, in the case that there are no partial maps (or they are purely black images), which reduces to the conventional setting of road extraction, our method is still slightly better than the state-of-the-art DeepLabV3+ (0.5297 versus 0.5277) on SpaceNet and DLinkNet (0.5243 versus 0.5240) on OSM for IoU.

The innovation of this article is twofold. First, it is a novel idea to conduct the road extraction task with some partial road maps as inputs since they are often available in practice and there is no good reason not to use them for road extraction. We note that the performance with partial maps is around 20% better than without them for IoU on the OSM dataset, from around 0.52 to 0.75. With the improved performance, the road extraction task becomes more meaningful. Second, we have made notable technical contributions for the fusion component and the loss component. For the fusion component, we designed a fused self-attention mechanism based on affinity matrix multiplication and a learnable gate module for deciding the confidence scores of pairs of feature maps from each branch such that the scores are computed in a complementary manner and adaptively adjusted based on data. For the loss component, we developed a new function called MP loss to suit our new setting of using partial maps better. We combined this MP loss with the conventional BCE loss and the Dice loss in the loss component. The experiments validate the benefits of the new MP loss.

In the remainder of this article, we review the related work in Section II and present our method, experiments, and experimental results in Sections III–V, respectively. Finally, we conclude this article in Section VI.

II. RELATED WORK

In this section, we briefly review some related studies about road extraction, data fusion, and road inpainting.

A. Road Extraction

Researchers first adopted histogram, clustering, and morphological techniques to extract roads [8], [9]. Then, Mnih [1] built a patch-based deep convolutional neural network (DCNN) for road extraction. Zhong et al. [10] then utilized fully convolutional network (FCN) [11] for the task. Later, a variant of FCN named DeconvNet gradually became the main stream. With the success of UNet [12] in the field of medical image segmentation, Zhang et al. [13] designed a deep residual UNet model to extract road features from remote sensing imagery data, which guarantees faster convergence speed during training and achieves good results.

Based on standard road extraction models, some auxiliary tasks are adopted to obtain fine-grained features, which are then used to help extract the roads. For example, Cheng et al. [2] proposed a cascade model that is responsible for extracting road areas and road centerlines at the same time. Li et al. [14] combined road extraction with edge detection task, which was used to identify the boundary of roads, and obtained better results. Moreover, Máttyus et al. [3] used a model for

initial road extraction first, then focused on mitigating the connectivity problem of the extracted roads, and improved the topology. Some studies [15], [16], [17] proposed novel loss functions to provide extra regulations and further improved the segmentation performance, e.g., Abdollahi et al. [15] used the Dice loss to mitigate the class imbalance problem and Mosinska et al. [16] and Wei et al. [17] proposed topology-related losses to take topology into consideration.

Recent studies of road extraction focus on exploiting more features. For example, Gao et al. [18] designed a multiple feature pyramid network with dilated convolutional layers to enlarge the receptive field. Some studies [4], [5], [19], [20] took advantages of atrous spatial pyramid pooling [21] to reduce the loss of spatial information and enhance context awareness. In particular, D-LinkNet [4] won the championship of DeepGlobe 2018 [5]. Some studies [6], [22], [23], [24] employed attention modules to capture long-range dependencies and obtain better features, while some other studies focused on the road extraction task with the support of other types of data, e.g., satellite images and GPS data [25], [26], satellite and LiDAR data [27], or pure GPS data [28]. Different from these existing studies, our work targets a new yet practical setting for road extraction, i.e., we make use of partial road maps for the task given that they are of high quality and publicly available.

B. Data Fusion

For tasks where satellite images and some other data types, such as LiDAR data, laser data, and GPS data, are involved, data fusion is normally adopted. There are mainly three types of techniques for data fusion, namely, inputs fusion, outputs fusion, and features fusion. Some studies [25], [29], [30] concatenated satellite images and other data types of image such as aerial LiDAR images as inputs for inputs fusion. Outputs fusion [31] can be achieved either using extra layers to refine predictions from different sources [31], [32] or directly taking an average of the predictions to obtain final outputs. Features fusion fuses feature maps from different branches. For example, Cao et al. [33] embedded features from one branch to another, while studies [26], [27], [34], [35] utilized an extra branch to process the fused features. We design a features fusion approach in this article. Different from the existing features fusion methods on satellite images, we take the branch for satellite images as an auxiliary one and that for partial road maps as the main one and fuse the features from the satellite images to the main branch.

C. Road Inpainting

As introduced in [36], road inpainting is a task of inpainting a road map (e.g., denoising and fixing broken road segments). Batra et al. [37] used road inpainting as a postprocessing step for refining extracted road maps. Zhang et al. [38] made use of satellite images for road inpainting, which concatenates satellite images and some noisy/broken road maps as inputs. We note that the noisy/imperfect road maps that are taken as inputs for the road inpainting task are by nature different from the partial road maps used in this work. For example,

the former usually does not have some entire road segments missing, and thus, the task is easier and feasible even without other data sources such as satellite images taken as inputs, but the latter may involve the majority of road segments missing, and thus, it is not feasible to generate complete road maps solely from the partial ones in general.

III. METHODOLOGY

We first present the framework, then introduce the newly designed GSAM, which could capture long-range semantics and control the information flow from the two data sources in a complementary manner, and finally introduce the employed loss functions, including the new MP loss.

A. Model Architecture

The framework of P2CNet is presented in Fig. 2. P2CNet has two branches: one takes the satellite images as inputs and is called the satellite branch (blue) and the other takes the partial road maps as inputs and is called the partial branch (orange). Each branch involves an encoder and a decoder, and for both of them, we employ DeepLabV3+ [7] with the backbone of ResNet-34 as the basic model due to its small number of parameters and good performance as verified in our experiments.

In particular, we develop a module for better utilizing and fusing the features outputted by the encoders, which is referred as the GSAM.

First, inspired by dual self-attention [39], we decide to apply two channelwise self-attentions to capture global context over features for both satellite and partial branches. Then, we fuse their affinity matrices to provide a mutual guidance and emphasize interdependent feature maps. Different from [39] that applies a self-attention mechanism on both spatial (H, W) and channel (C) dimensions, we only consider channel dimension in this work. There are two reasons that we employ channelwise but not spatial attentions in GSAM. First, channelwise attentions could save much memory, as the shape of its affinity matrix is $C \times C$, while that of spatial attentions is $HW \times HW$. Second, partial road maps usually involve a small number of road pixels, and many extracted feature maps would be close to all-zero matrices, making it less useful to apply spatial attentions. With channelwise self-attentions, various semantics could be highlighted per channel to improve the feature representation of specific semantics.

Second, a gate mechanism is adopted to further control the information flow in a complementary manner. We then take the generated road map from the partial branch as the output, i.e., we take the partial branch as the main branch. We note that the task of generating complete road maps from partial road maps without features from satellite images is not feasible (as verified via experiments) and needs to be guided with features extracted from the satellite images.

B. Gated Self-Attention Module

Self-attention mechanism has shown its advantages in capturing long-range dependencies and improving the feature representation of specific semantics [39], while the gate mechanism

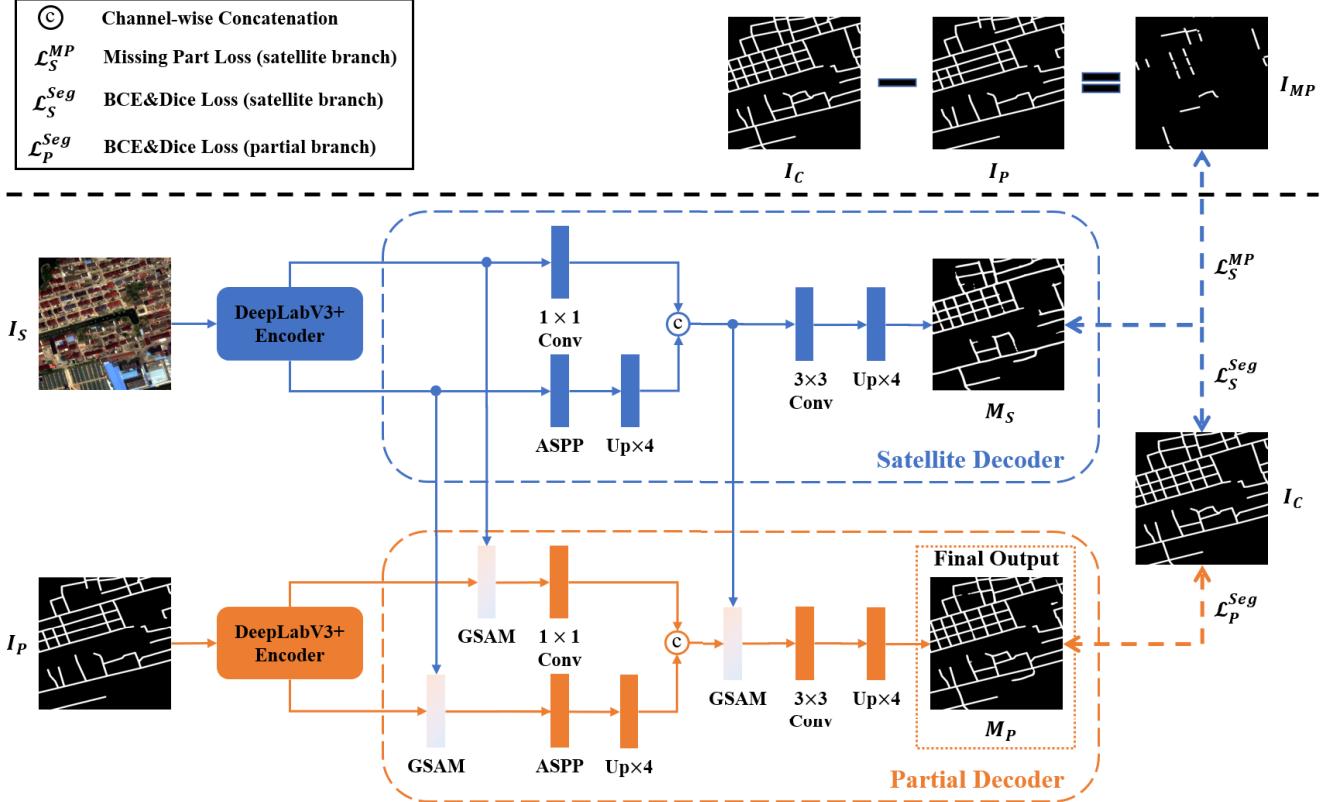


Fig. 2. Overview of the proposed P2CNet model. GSAMs are installed in the partial branch. There are also four GSAMs installed in the partial encoder, which is not explicitly displayed in the figure. The MP loss is applied to the satellite branch only.

is widely adopted in multimodal data fusion to fuse different features. We build the GSAM module to capture rich features from satellite images and partial road maps with self-attentions and then fuse them in a complementary manner with a gate. Note that there are also four GSAMs set right behind the last four stages of ResNet-34 in the partial branch's encoder. GSAM is presented in Fig. 3, and we formally describe it as follows.

1) *Channelwise Self-Attention*: It takes feature maps $F_S \in \mathbb{R}^{C \times H \times W}$ and $F_P \in \mathbb{R}^{C \times H \times W}$ from the satellite branch and the partial branch, respectively, as inputs. The affinity matrices $A_S \in \mathbb{R}^{C \times C}$ and $A_P \in \mathbb{R}^{C \times C}$ are derived directly from the inputs by reshaping F_S and F_P both to $\mathbb{R}^{C \times HW}$, then performing a matrix multiplication between them and their transpose, and finally applying a softmax operation

$$a_X^{ji} = \frac{e^{F_X^i \cdot F_X^j}}{\sum_{i=1}^C e^{F_X^i \cdot F_X^j}} \quad (1)$$

where $X \in \{S, P\}$ and a_X^{ji} is the element with coordinate of (j, i) in A_X , measuring the impact of the i th channel on the j th channel. Then, to achieve the goal of mutual guidance, an elementwise multiplication on A_S and A_P is performed to obtain the fused affinity matrix $A_{\text{fuse}} \in \mathbb{R}^{C \times C}$. After that, two matrix multiplications between the transpose of A_{fuse} and F_S and F_P are performed, and their results are reshaped to $\mathbb{R}^{C \times H \times W}$. The enhanced features $F'_S \in \mathbb{R}^{C \times H \times W}$ and $F'_P \in \mathbb{R}^{C \times H \times W}$ are calculated as follows:

$$A_{\text{fuse}} = A_S \odot A_P \quad (2)$$

$$F'_X^j = \beta_X \sum_{i=1}^C \left(a_{\text{fuse}}^{ji} F_X^i \right) + F_X^j \quad (3)$$

where j means each position of feature maps, β_X represents two learnable weights initialized as 0 for S and P , and a_{fuse}^{ji} is the element with the coordinate of (j, i) in A_{fuse} .

2) *Gate Mechanism*: Then, F'_S and F'_P are concatenated in the channel dimension to be the input $F_C \in \mathbb{R}^{2C \times H \times W}$ of convolution blocks. The convolution block consists of a convolution layer and a batch normalization layer, followed by a ReLU activation. The kernel sizes of the convolution layers are all 3×3 , and their output channels are all C . After convolution, two feature maps $F_{C_S} \in \mathbb{R}^{C \times H \times W}$ and $F_{C_P} \in \mathbb{R}^{C \times H \times W}$ are generated. Next, F_{C_S} and F_{C_P} are unsqueezed to have a new dimension and concatenated along it. A softmax layer is further attached to offer complementary fusion and result in $F'_C \in \mathbb{R}^{2 \times C \times H \times W}$. Then, two elementwise multiplications are used to obtain the final features $F''_S \in \mathbb{R}^{C \times H \times W}$ and $F''_P \in \mathbb{R}^{C \times H \times W}$. The fused features $F_{\text{fuse}} \in \mathbb{R}^{C \times H \times W}$ are the sums of F''_S and F''_P

$$F'_C[0, :, :, :] + F'_C[1, :, :, :] = 1 \quad (4)$$

$$\begin{aligned} F_{\text{fuse}} = & \underbrace{F'_S \odot F'_C[0, :, :, :]}_{F''_S} \\ & + \underbrace{F'_P \odot F'_C[1, :, :, :]}_{F''_P} \end{aligned} \quad (5)$$

where F'_C is the output of the convolution blocks and $[0, :, :, :]$ and $[1, :, :, :]$ mean the 1st and 2nd channels. Finally, F_{fuse} is forwarded back to the partial branch for further processing.

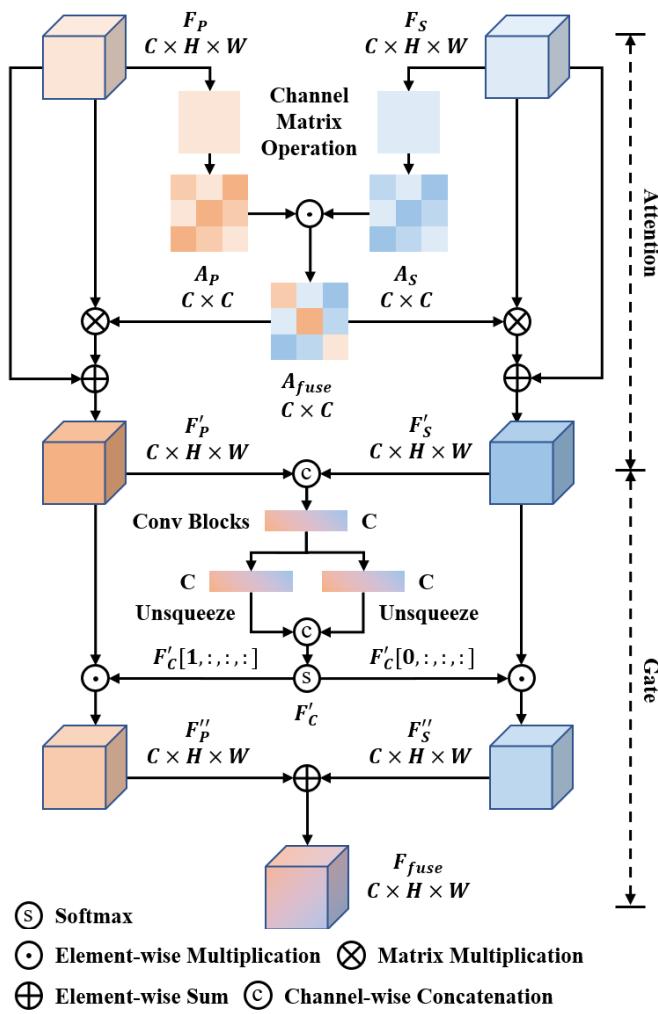


Fig. 3. Details of GSAM.

C. Loss Functions

We employ a combination of BCE loss and Dice loss as the basic loss functions of our model. The BCE loss is widely accepted in binary semantic segmentation tasks, while the Dice loss is usually used when the classes are imbalanced. In the road extraction task, the number of road pixels is far less than that of background pixels. These loss functions are defined as follows:

$$\mathcal{L}_X^{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N (I_C^i \log M_X^i + (1 - I_C^i)(1 - \log M_X^i)) \quad (6)$$

$$\mathcal{L}_X^{\text{Dice}} = 1 - \frac{2|M_X \cap I_C|}{|M_X| + |I_C|} \quad (7)$$

where $X \in \{S, P\}$, $N = H \times W$, $I_C \in \mathbb{R}^{1 \times H \times W}$ is the complete road map, and $M_S \in \mathbb{R}^{1 \times H \times W}$ and $M_P \in \mathbb{R}^{1 \times H \times W}$ are the generated road maps from satellite and partial branches, respectively.

In addition, we propose a new loss called the MP loss to further exploit the partial road maps. The MP loss is defined based on those missing road pixels from the partial road maps as follows:

$$I_{\text{MP}} = I_C - I_P \quad (8)$$

$$\mathcal{L}_S^{\text{MP}} = -\frac{1}{|\Omega_{\text{MP}}|} \sum_{i \in \Omega_{\text{MP}}} (I_{\text{MP}}^i \log M_S^i) \quad (9)$$

where I_{MP} is the MP of the road map, I_P is the partial road map, Ω_{MP} is the road pixels set of I_{MP} , and $|\cdot|$ means the number of elements in the set. The rationale of MP loss is to assign more weights to the missing road pixels when performing backpropagation. Note that the MP loss corresponds to a BCE loss on the missing road pixels. We do not define the MP loss as a Dice loss on the missing road pixels since the latter is sensitive to small regions and makes the training process unstable [40]. For the complete road map, which has 1024^2 pixels, the Dice loss works well, but for the MP, which occupies a very small number of pixels only, e.g., several hundreds or fewer pixels, a Dice loss would not work well.

Finally, the loss of satellite branch \mathcal{L}_S , the loss of partial branch \mathcal{L}_P , and the total loss $\mathcal{L}_{\text{total}}$ are given as follows:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_S^{\text{BCE}} + \mathcal{L}_S^{\text{Dice}} + \lambda \mathcal{L}_S^{\text{MP}}}_{\mathcal{L}_S} + \underbrace{\mathcal{L}_P^{\text{BCE}} + \mathcal{L}_P^{\text{Dice}}}_{\mathcal{L}_P} \quad (10)$$

where λ is a hyperparameter. Note that we incorporate the MP loss in the satellite branch only, which we explain as follows. The MP loss would help to put more weights on those missing road pixels and thus extract them out correctly, but at the same time, it would lower down the weights on other pixels, including those near the roads in the partial road maps, resulting in some potential misclassifications. Specifically, for those pixels near to roads, they tend to be classified to be road pixels (since they correspond to roads in the satellite image), but their corresponding labels are background pixels (since only those pixels on the centerlines of roads are labeled as road pixels and others as background pixels). By incorporating the MP loss in the satellite branch only, we achieve the following: 1) some additional weights have been put on those missing road pixels as we aim to and 2) the misclassifications are avoided since they can only happen in the satellite branch and the generated road map in the partial branch is taken as the output. We conduct detailed experiments on different strategies of using the MP loss strategies in Section V-B, which verify the superiority of the strategy as implemented by (10).

IV. EXPERIMENTS

A. Dataset

By following the existing studies [37], [41], [42], [43], we use the SpaceNet² [44] and OSM³ datasets. SpaceNet contains 1300×1300 pixels satellite images and their corresponding complete road maps, which cover four cities, including Las Vegas, Paris, Shanghai, and Khartoum. After removing those satellite images, for which there are no complete road maps, we have 283, 257, 1028, and 981 satellite images for Khartoum, Paris, Shanghai, and Las Vegas, respectively. In total, we have 2549 satellite images and convert them to 8-bit RGB images. Also, the OSM dataset contains

²<https://spacenet.ai/spacenet-roads-dataset/>

³<https://github.com/mitroadmaps/roadtracer/>

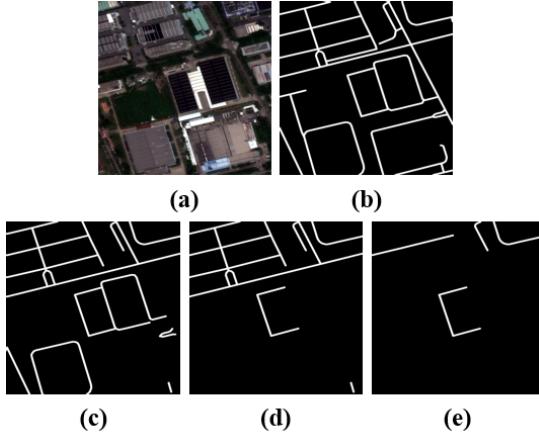


Fig. 4. Sample of the datasets. (a) Satellite image. (b) Complete road map. (c) 75% partial road map. (d) 50% partial road map. (e) 25% partial road map.

1024 × 1024 pixels satellite images of 40 cities from Google Maps and the road maps from OSM.

We define a parameter called completeness ratio α of a partial road map, which represents the ratio of the number of road pixels in the partial road map and that in the complete road map. To obtain the partial road maps, we randomly erase complete road segments until the remaining road pixels are below α .

The preprocessing step is uniform, and we only take SpaceNet as an example. For partial road maps, we choose to generate them from the complete ones by erasing some roads, but not to use the road maps from OSM, which are incomplete for two reasons. First, the road maps from SpaceNet and those from OSM are from two different sources and extra efforts are needed to align them to each other, which could be tedious. Second, with the road maps from OSM used as partial road maps, we cannot control the extent to which the roads are missing for studying the effects of the missing roads on the performance of the proposed models.

To generate a partial road map with the completeness ratio α , we iteratively drop a road from the road map until the ratio of road pixels in the remaining road map and that in the original one becomes below α . Note that we do not define the completeness ratio as the ratio of the numbers of roads since roads can have different lengths and a partial road map with one long road missing would be significantly different from that with one short road missing. We prepare five sets of partial road maps as follows: for the first four sets, we generate them by setting α as 0%, 25%, 50%, and 75%, and for the fifth set, we generate each partial road map by randomly sampling it from one of the first four sets except for the 0% set and call the set mix. An illustration of some partial road maps is shown in Fig. 4.

B. Training Details

We randomly split the dataset into the train, validation, and test sets, with ratios of 0.8, 0.1, and 0.1, respectively. We use random crops with a size of 1024 × 1024 for the train set and the crop with a size of 1024 × 1024 in the center for

validation and test. Some data augmentation methods, e.g., random horizontal flip, shift, Gaussian noises, and brightness, are applied to the train set. The proposed model is implemented using PyTorch and adopts the Adam optimizer with an initial learning rate of 2×10^{-4} and a momentum of 0.5 for optimization. The training batch size is set as 4. After training each epoch, validation is performed, and a learning rate scheduler is further utilized based on this. If the mean IoU (mIoU) score does not increase continuously for five epochs, the learning rate would be multiplied by 0.2. In addition, the training process would be terminated if the mIoU score does not improve for 20 consecutive epochs. Then, we use the model, whose validation mIoU score is the best, to perform testing.

C. Baselines

We select ten baselines, including PSPNet [45], PAN [46], FPN [47], CasNet [2], LinkNet [48], DLinkNet [4], DeepLabV3 [49], DeepLabV3+ [7], RoadCon [37], and SGCN [50], for comparisons. In addition, we consider McGAN [38], which was originally designed for road inpainting, for comparison, for which we use partial road maps as noisy ones. For baseline methods, we adapt them to the proposed setting with inputs fusion. All methods use the same inputs (namely, partial maps and satellite images) for a fair comparison. Moreover, we take ResNet-34, which is pretrained on ImageNet, as the backbone for all methods, and each model is trained and tested with 1024 × 1024 image patches.

D. Evaluation Metrics

Six metrics are applied in the experiments: precision P , recall R , F1 score $F1$, APLS score APLS [44], IoU, and mean IoU (mIoU). P measures the ratio of correctly extracted road pixels among all extracted road pixels, R measures the ratio of correctly extracted road pixels among all actual road pixels, and $F1$ is a harmonic mean of P and R . APLS is a topology-oriented metric and measures the differences of optimal path lengths between nodes in generated and ground-truth road graphs. IoU calculates the ratio of the overlapping area to the union area of the actual and extracted road pixels and mIoU is the average of IoU's for road pixels and background pixels.

Road extraction is a binary segmentation task, where only two classes involve, i.e., foreground and background. All metrics except APLS could be used as segmentation-related metrics and could be calculated via confusion matrix, which are shown as follows:

$$P = \frac{TP}{TP + FP} \quad (11)$$

$$F = \frac{TP}{TP + FN} \quad (12)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (13)$$

$$\text{IoU} = \text{IoU}_f = \frac{TP}{TP + FP + FN} \quad (14)$$

$$\text{IoU}_b = \frac{FN}{FN + TP + TN} \quad (15)$$

TABLE I

COMPARISONS OF MODELS FOR ROAD EXTRACTION. ALL MODELS ARE TRAINED AND TESTED ON THE MIX SPACENET DATASET; THE TABLE IS SORTED BASED ON MIoU IN ASCENDING ORDER. (SAT)/(PAR) MEANS THE OUTPUTS FROM THE SATELLITE/PARTIAL BRANCH

Method	<i>P</i>	<i>R</i>	SpaceNet			<i>mIoU</i>
			<i>F1</i>	<i>APLS</i>	<i>IoU</i>	
CasNet [2]	0.8056	0.7653	0.7849	0.6105	0.6460	0.8094
PSPNet [45]	0.8275	0.7679	0.7966	0.6597	0.6619	0.8183
RoadCon [37]	0.8419	0.7758	0.8075	0.7040	0.6772	0.8266
McGAN [38]	0.8388	0.7990	0.8178	0.7420	0.6917	0.8345
FPN [47]	0.8332	0.8075	0.8202	0.7297	0.6952	0.8361
DeepLabV3 [49]	0.8371	0.8052	0.8208	0.7486	0.6961	0.8367
PAN [46]	0.8365	0.8082	0.8221	0.7078	0.6979	0.8376
DeepLabV3+ [7]	0.8378	0.8085	0.8229	0.7359	0.6990	0.8382
SGCN [50]	0.8377	0.8100	0.8237	0.7415	0.7002	0.8388
LinkNet [48]	0.8412	0.8075	0.8240	0.7384	0.7007	0.8392
DLinkNet [4]	0.8385	0.8116	0.8248	0.7445	0.7019	0.8398
P2CNet	0.8424	0.8150	0.8285	0.7489	0.7071	0.8426

TABLE II

COMPARISONS OF MODELS FOR ROAD EXTRACTION. ALL MODELS ARE TRAINED AND TESTED ON THE MIX OSM DATASET; THE TABLE IS SORTED BASED ON MIoU IN ASCENDING ORDER. (SAT)/(PAR) MEANS THE OUTPUTS FROM THE SATELLITE/PARTIAL BRANCH

Method	<i>P</i>	<i>R</i>	OSM			<i>mIoU</i>
			<i>F1</i>	<i>APLS</i>	<i>IoU</i>	
PSPNet [45]	0.8365	0.8044	0.8201	0.7234	0.6851	0.8311
PAN [46]	0.8579	0.7851	0.8199	0.6449	0.6947	0.8313
DeepLabV3 [49]	0.8361	0.8054	0.8205	0.7368	0.6956	0.8314
CasNet [2]	0.8701	0.8030	0.8331	0.6871	0.7140	0.8424
McGAN [38]	0.8815	0.8013	0.8395	0.7843	0.7234	0.8474
FPN [47]	0.8655	0.8192	0.8417	0.7625	0.7267	0.8490
RoadCon [37]	0.8762	0.8167	0.8454	0.7779	0.7322	0.8522
DeepLabV3+ [7]	0.8662	0.8353	0.8505	0.7821	0.7398	0.8562
DLinkNet [4]	0.8772	0.8258	0.8507	0.7793	0.7402	0.8566
SGCN [50]	0.8760	0.8270	0.8508	0.7803	0.7403	0.8566
LinkNet [48]	0.8835	0.8233	0.8523	0.7801	0.7427	0.8580
P2CNet	0.8788	0.8430	0.8605	0.8028	0.7552	0.8648

$$\text{mIoU} = \frac{1}{2}(\text{IoU}_f + \text{IoU}_b) \quad (16)$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. IoU_f represents the IoU of road, while IoU_b is that of background.

In particular, for APLS, it does not treat the extracted road maps as an image. Instead, it extracts the image's skeleton and builds a graph based on the skeleton. Then, it measures the sum-up differences in the lengths of the shortest paths between graph nodes of the newly built graph and the exact ground truth. In this way, APLS could measure the topology difference between the two graphs well. For example, if there exists a road segment between two points in the road map and the road segment is broken in the extracted graph, then we might still be able to travel between two points, but the length of the shortest path would become much larger.

V. RESULTS

A. Performance Comparisons

1) *Quantitative Results*: The results are shown in Tables I and II.

For the selected models, we set the number of input channels to 4 and concatenate satellite images and partial road maps along channel dimension as inputs, as inspired by McGAN.

Besides, inputs fusion could obtain quite good results, according to later experiments on the ways of fusion. In particular, for McGAN, we use a pretrained DeepLabV3+ model as the generator. The results show that its discriminators do not work well in our setting, which causes a slight performance decrease of 1.04% on mIoU, compared with DeepLabV3+. According to the results, our P2CNet model achieves the best results consistently across six metrics (including the topology-oriented one APLS), on two datasets from different sources and of different qualities (OSM and SpaceNet), and among 11 methods, e.g., for IoU score, P2CNet could outperform the best baseline, say, LinkNet, by 1.27% on the OSM mix dataset. Besides, the APLS score of P2CNet is 1.85% better than the second one.

In addition, in our P2CNet, the IoU scores of the predictions based on the partial branch are 18.60% and 23.05% higher than those based on the satellite branch in SpaceNet and OSM, respectively. Such huge gaps could show the great importance of partial road maps that could help to reduce the difficulty of road extraction greatly.

2) *Model Complexity*: As shown in Table III, we use *thop*⁴ to calculate the number of parameters and multiply–accumulate (MAC) operations. Both satellite images and

⁴<https://github.com/Lyken17/pytorch-OpCounter>

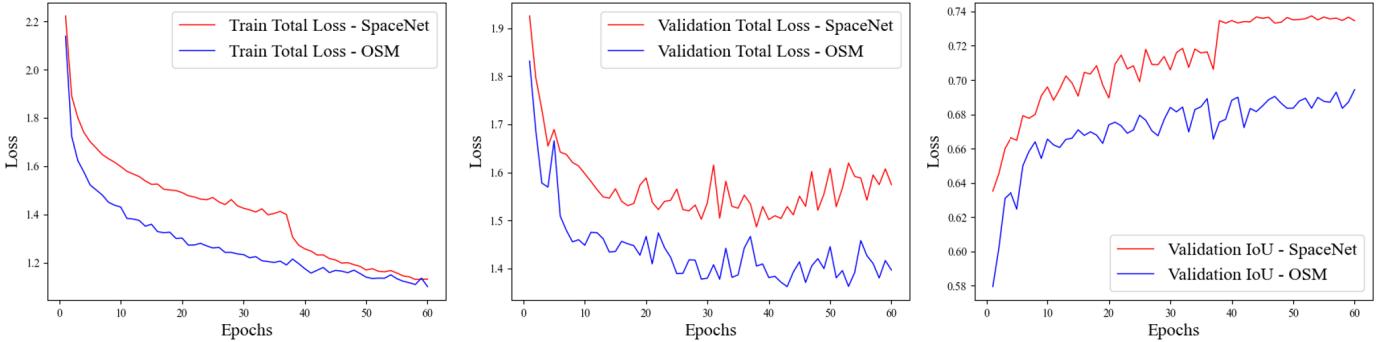


Fig. 5. Loss and IoU curves of P2CNet on mix SpaceNet and OSM datasets.

TABLE III
COMPARISONS OF MODEL COMPLEXITY

Method	#Branch	#Parameters (M)	MACs (G)
DLinkNet [4]	1	31.21	4.68
DeepLabV3+ [7]	1	22.44	6.10
SGCN [50]	1	32.78	26.04
P2CNet	2	61.04	28.04

partial road maps are taken as the inputs, and their spatial resolutions are uniformly set as 224×224 . We have the following observations.

- 1) Our P2CNet consists of two branches. The satellite branch is exactly a DeepLabV3+ model, and the partial branch is built upon DeepLabV3+ (with novel attention-based GSAM modules). Hence, our model has more parameters and MACs than DeepLabV3+.
- 2) The number of MACs of our P2CNet (two branches) is larger than that of SGCN (one branch) by 2G only, which shows the effectiveness of our design.

3) *Model Convergence:* To show the convergence speed of our P2CNet, we show the results of the total training and validation losses, as well as the validation IoU scores, on both mix SpaceNet and OSM datasets in Fig. 5. We could observe that P2CNet could converge fast, e.g., P2CNet could converge within 40 epochs on both datasets.

4) *Qualitative Results:* We visualize four sets of satellite images, partial road maps, complete road maps, and the generated road maps by different models from SpaceNet in Fig. 6 and OSM in Fig. 7. We use red rectangles to highlight some areas that are difficult to be detected, and it could be observed that our P2CNet works well on these areas. This is due to the effectiveness of our proposed modules, say, GSAM and MP loss. With the help of channelwise self-attention and the gate mechanism, the proposed GSAM could filter out the useless and sparse features in partial road maps, and the remaining useful road features could be better referred by the module and then build up stronger relationship with the features extracted from the satellite images.

An explanation for this is given as follows, and taking the third row of Fig. 6 as an example, the area inside the red rectangle is quite hard to be classified as roads. We could observe that in column (b), there are some existing roads in the left side of rectangle, and these existing parts have perceptually similar features as the other parts inside the

rectangle according to column (a). Then, the existing partial road maps appear to act as clues and could encourage the model to analyze the relationships between these areas and other areas, say, the right part of the rectangle has similar features as parts with partial road maps, and hence, these areas would probably contain roads.

B. Ablation Study

To study the P2CNet model in detail, extensive experiments are conducted in this section, including different ways of fusion, MP loss strategies, ablation study on modules, and robustness of the proposed model.

1) *Fusion Strategies:* We investigate four different fusion strategies based on DeepLabV3+ with the backbone of ResNet-34.

- 1) Outputs augmentation performs fusion directly by taking the union of binarized outputs and partial road maps as the final outputs.
- 2) Inputs fusion feeds the model with the concatenations of satellite images and partial road maps.
- 3) Outputs fusion is similar to outputs augmentation, but instead of using sigmoid activation at last, it uses two 3×3 convolution layers to convert the inactivated outputs and partial road maps into the same space, concatenates the outputs of convolution layers, and uses another 1×1 convolution layer to obtain the final outputs, followed by a sigmoid activation.
- 4) Features fusion is our P2CNet with all GSAM replaced with 3×3 convolution layers. The model of features fusion is two-branch, while others are one branch.

More details of these four fusion strategies can be found in the Appendix.

We uniformly train these models on the SpaceNet mix dataset and then test them on both mix and 0% datasets. The results are shown in Table IV. We observe that outputs augmentation outperforms others on the 0% dataset but does not perform well on mix dataset because the partial road maps are not involved in the training process in this case. In addition, inputs fusion does not have a good generalization ability, and the reason is that partial road maps are binary images, and if they are directly used as parts of the inputs, as inputs fusion does, the model would focus on how to reserve such information instead of inferring the road maps from satellite

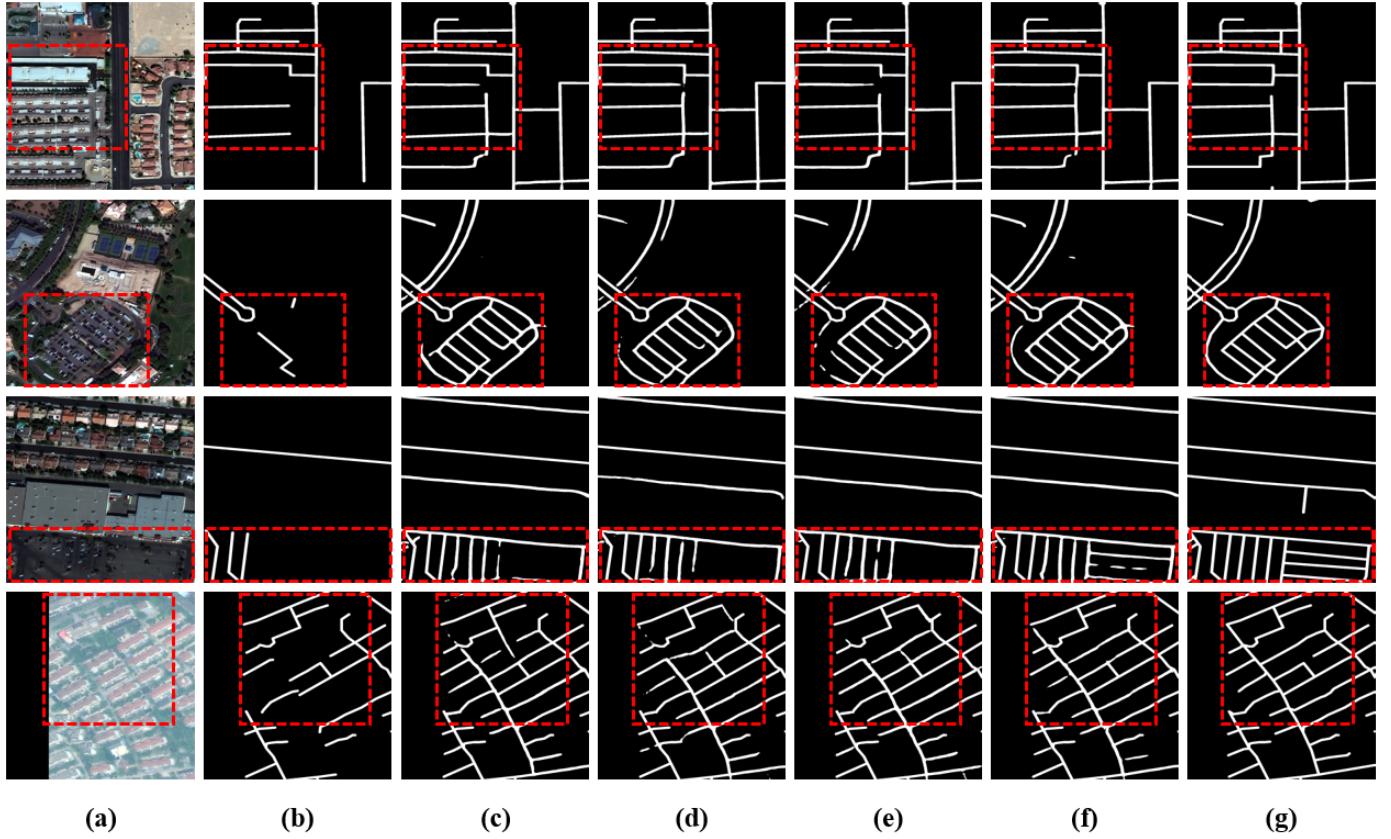


Fig. 6. Qualitative results of mix SpaceNet. (a) Satellite image. (b) Partial road map (SpaceNet). (c) DeepLabV3+. (d) LinkNet. (e) DLinkNet. (f) P2CNet (ours). (g) Complete road map (SpaceNet).

TABLE IV

RESULTS OF FUSION STRATEGIES. ALL MODELS ARE TRAINED ON THE MIX DATASET; α_{TEST} MEANS α ADOPTED FOR TESTING. (SAT)/(PAR) IN FEATURE FUSION MEANS THE OUTPUTS FROM THE SATELLITE/PARTIAL BRANCH. **BOLD** SCORES ARE THE BEST, WHILE SCORES WITH UNDERLINE ARE THE SECOND BEST

Method	α_{test}	F1	IoU
Outputs Augmentation	mix	0.7711	0.6274
Outputs Fusion	mix	0.7782	0.6369
Inputs Fusion	mix	<u>0.8229</u>	<u>0.6990</u>
Features Fusion (sat)	mix	0.6843	0.5201
Features Fusion (par)	mix	0.8248	0.7019
Outputs Augmentation	0%	0.6909	0.5277
Outputs Fusion	0%	0.6597	0.4922
Inputs Fusion	0%	0.6556	0.4876
Features Fusion (sat)	0%	0.6843	0.5201
Features Fusion (par)	0%	0.6688	0.5024

features. We employ features fusion since it achieves the best performance when considering both mix and 0% datasets, i.e., it generalizes well on both cases, because individual loss is calculated and backpropagated on the satellite branch, so the features of satellite images could be well captured and the influence of partial road maps on satellite images would not be as severe as that in inputs fusion.

2) *Loss Strategies*: The two-branch features fusion model is used as the basic model. There are three possible ways to employ the MP loss: 1) Both, add the MP loss to both the satellite and partial branches; 2) Par, only add the loss to the partial branch; and 3) Sat, only add to the satellite branch. As the MPs of a road map only occupy a very small

TABLE V
EXPERIMENTS ON MP LOSS STRATEGIES. ALL MODELS ARE TRAINED AND TESTED ON MIX DATASET (SPACENET)

Method	λ	R	F1	IoU
Both	1	0.8241	0.8236	0.7001
	10	0.8747	0.8057	0.6746
	20	0.8978	0.7594	0.6121
	30	0.9179	0.7353	0.5814
Par	1	0.8243	0.8248	0.7019
	10	0.8750	0.8063	0.6755
	20	0.8992	0.7615	0.6148
	30	0.9164	0.7532	0.6041
Sat	1	0.8105	0.8254	0.7027
	10	0.8095	0.8258	0.7033
	20	0.8118	0.8258	0.7033
	30	0.8108	0.8261	0.7037

portion of pixels, small changes of the weight λ , e.g., from 0.9 to 1, might not affect the performance much and the influence of randomness caused by data augmentation might blow up, we only try weights with relatively large intervals in this article. To amplify the impact of the MP loss on metrics, we use $\lambda = 1, 10, 20$, and 30 to demonstrate the differences among the three loss strategies.

The results are shown in Table V, where the R scores are also included since the direct target of introducing the MP loss is to improve R scores. We observe that Both and Par have similar patterns. With the increase of λ , the R scores are improved greatly, but at the cost of wrongly classifying many background pixels as road pixels. An example is shown in Fig. 8 for illustration.

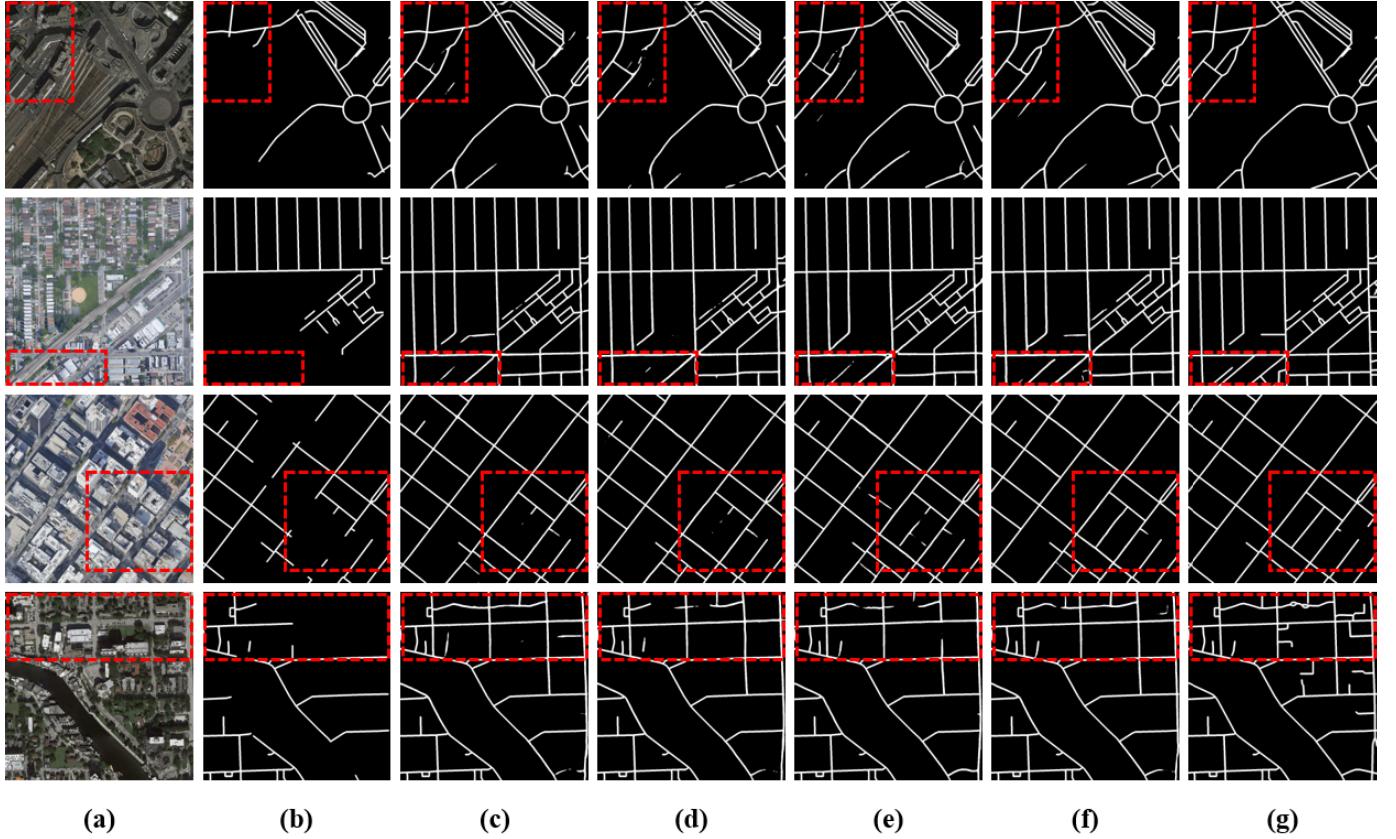


Fig. 7. Qualitative results of mix OSM. (a) Satellite image. (b) Partial road map (OSM). (c) DeepLabV3+. (d) LinkNet. (e) DLinkNet. (f) P2CNet (ours). (g) Complete road map (OSM).

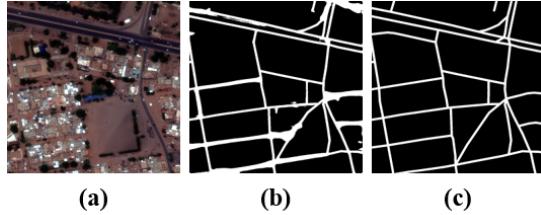


Fig. 8. Illustration of the effects of the MP loss (Both, $\lambda = 30$). (a) Satellite image. (b) Generated road map (from the partial branch). (c) Complete road map.

This phenomenon is caused by adding the MP loss to the main branch, i.e., the partial branch. The purpose of the MP loss is to give more attention to the MPs, which would have a weight of $1 + \lambda$, while other pixels have a weight of 1. Therefore, once λ is set too large, the model would focus on the MPs and neglect other pixels (as their weights are relatively small). As for Sat, the outputs of it would suffer from the same problem, but its feature maps would contain more coarse and richer information, which could be filtered during fusion to produce better outputs in the partial branch. From Table V, Sat appears to be more stable, and the scores of $F1$ and IoU are the highest when $\lambda = 30$.

3) *Efficacy of GSAM and MP Loss:* Our P2CNet is built on the two-branch features fusion model, with the MP loss added to the satellite branch only (Sat). P2CNet has two prominent components: GSAM and MP loss. We study their effects by incorporating them to the basic model gradually

TABLE VI
ABLATION STUDY ON P2CNET (MIX DATASET, SPACENET)

Method	$F1$	IoU
Basic	0.8248	0.7019
+ MP	0.8261	0.7037
+ MP + GSAM (Gate (exist))	0.8257	0.7032
+ MP + GSAM (Gate)	0.8266	0.7045
+ MP + GSAM (Gate + Attention (exist))	0.8272	0.7053
+ MP + GSAM (Gate + Attention)	0.8285	0.7071

and report on $F1$ and IoU metrics. The results are shown in Table VI, where all models are trained and tested on the mix dataset.

We first employ the MP loss to the basic model, and the IoU score is improved from 70.19% to 70.37%. Then, we study our proposed GSAM, which consists of attention and gate mechanism. We highlight the differences among our designed modules and existing ones as follows.

- 1) For gate mechanism, the shape of F'_C in Fig. 3 is $2 \times C \times H \times W$, but that of existing works is $2 \times H \times W$, which means that we would have C confidence maps for each pair of feature maps of F'_S and F'_P for each channel, while existing work would uniformly have one confidence map for all channels. To show the effect of such design, we conduct an experiment named (+ MP + GSAM (Gate (exist))), in which a shared confidence map is used for all channels, and the IoU score is 70.32% and it is 0.13% less than that of (+ MP + GSAM (Gate)).

TABLE VII

COMPARISONS OF MODELS FOR ROAD EXTRACTION. ALL MODELS ARE TRAINED AND TESTED ON THE 0% SPACENET DATASET; THE TABLE IS SORTED BASED ON MIoU IN ASCENDING ORDER. (PAR) MEANS THE OUTPUTS FROM THE PARTIAL BRANCH

Method	<i>P</i>	<i>R</i>	SpaceNet			
			<i>F1</i>	<i>APLS</i>	<i>IoU</i>	<i>mIoU</i>
CasNet [2]	0.5991	0.6561	0.6263	0.4266	0.4559	0.7028
PSPNet [45]	0.6330	0.6512	0.6420	0.4825	0.4727	0.7130
SGCN [50]	0.6344	0.6940	0.6629	0.5557	0.4957	0.7251
DLinkNet [4]	0.6530	0.6721	0.6624	0.5743	0.4953	0.7256
LinkNet [48]	0.6340	0.7013	0.6659	0.5588	0.4992	0.7269
DeepLabV3 [49]	0.6621	0.6794	0.6706	0.5807	0.5045	0.7307
RoadCon [37]	0.6763	0.6705	0.6734	0.5719	0.5076	0.7329
PAN [46]	0.6747	0.6913	0.6829	0.5986	0.5185	0.7386
FPN [47]	0.6787	0.6875	0.6831	0.6044	0.5187	0.7388
DeepLabV3+ [7]	0.6717	0.7112	0.6909	0.6192	0.5277	0.7433
P2CNet (ours, par)	0.6944	0.6902	0.6923	0.6061	0.5297	0.7451

TABLE VIII

COMPARISONS OF MODELS FOR ROAD EXTRACTION. ALL MODELS ARE TRAINED AND TESTED ON THE 0% OSM DATASET; THE TABLE IS SORTED BASED ON MIoU IN ASCENDING ORDER. (PAR) MEANS THE OUTPUTS FROM THE PARTIAL BRANCH

Method	<i>P</i>	<i>R</i>	OSM			
			<i>F1</i>	<i>APLS</i>	<i>IoU</i>	<i>mIoU</i>
PSPNet [45]	0.6317	0.6559	0.6436	0.5245	0.4745	0.7039
PAN [46]	0.6425	0.6882	0.6481	0.5874	0.4794	0.7053
CasNet [2]	0.6226	0.6888	0.6540	0.5701	0.4859	0.7094
DeepLabV3 [49]	0.6463	0.6975	0.6709	0.6334	0.5048	0.7209
RoadCon [37]	0.6679	0.6871	0.6774	0.6723	0.5121	0.7259
SGCN [50]	0.6644	0.7009	0.6821	0.6485	0.5176	0.7287
FPN [47]	0.6767	0.6892	0.6829	0.6468	0.5184	0.7297
DeepLabV3+ [7]	0.6573	0.7147	0.6848	0.6639	0.5207	0.7300
LinkNet [48]	0.6735	0.7015	0.6872	0.6678	0.5235	0.7323
DLinkNet [4]	0.6684	0.7080	0.6876	0.6575	0.5240	0.7323
P2CNet (ours, par)	0.6899	0.6860	0.6879	0.6613	0.5243	0.7335

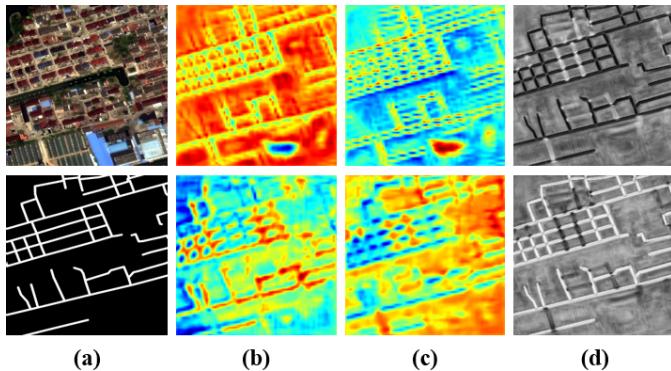


Fig. 9. Visualization of gate mechanism. The upper row is related to the satellite branch, while the bottom row is related to the partial branch. (a) Inputs. (b) 5th channel maps from F'_S and F'_P . (c) 27th channel maps from F'_S and F'_P . (d) Gate values.

- 2) Also, for attention mechanism, the one that does not use fused affinity matrix (+ MP + GSAM (**Gate + Attention (exist)**)) would result in an IoU score of 70.53%, which is 0.18% less than the model with it (+ MP + GSAM (**Gate + Attention**)), and this final model could achieve the best performance of 82.85% and 70.71% for F1 and IoU, respectively.

We also visualize some feature maps of the gate mechanism in GSAM in Fig. 9, where in Fig. 9(b) and (c), red values represent large values, while blue values are smaller ones.

As there are hundreds of channels per group of feature maps in the gate, we select the 5th and 27th channels of F'_S and F'_P to indicate that our proposed gate mechanism could highlight clear semantic areas, e.g., road and background in this case. In addition, the figures in Fig. 9(d) are from two channels of F'_C , which could fuse features from satellite and partial branches with different confidences in a complementary manner.

As we could observe from the second row, the feature map of Fig. 9(c) has larger values on the “black” part of Fig. 9(a), which means that this feature map focuses on extracting features from the useless part. Such information would be redundant during fusion in GSAM, and by utilizing the channelwise self-attention, the confusable information could be safely filtered and the fusion process could work better.

4) *Robustness Against No Partial Road Maps*: To show the robustness of different models, we also train all models except for McGAN on the 0% dataset and then test them on the 0% dataset. The reason behind is that in platforms such as OSM, there are also many places that have few or even no partial road maps, and our proposed P2CNet is expected to behave well on such cases.

We did not train the McGAN model because its discriminators require the partial road maps as the inputs of noisy road maps. As our P2CNet is a two-branch model, we use all-zero maps as the inputs of the partial branch, while other

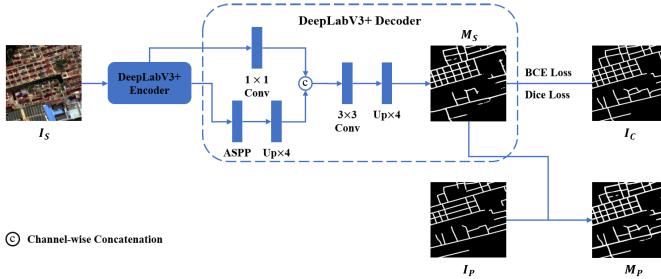


Fig. 10. Model structure of outputs augmentation.

models follow their original structure, i.e., they act as normal road extraction models and do not use partial road maps. For our P2CNet, the MPs of MP loss are road pixels in complete road maps, and the weights of road pixels appear to be twice as large as those of background pixels. The results are shown in Tables VII and VIII and show that P2CNet is still slightly better than the state-of-the-art DeepLabV3+ (0.5297 versus 0.5277) on SpaceNet and DLinkNet (0.5243 versus 0.5240) on OSM for IoU.

VI. CONCLUSION

In this article, we study the road extraction task based on satellite images and partial road maps. We introduce a two-branch framework called P2CNet for the task. P2CNet involves two prominent components: GSAM and MP loss. GSAM fuses the feature maps from two branches and MP loss helps to put more attention to missing road pixels. Extensive comparative experiments were conducted, which shows that P2CNet achieves the best performance compared with existing state-of-the-art models. One interesting research direction is to use publicly available richer data such as different layers of OSM for other related tasks such as semantic map labeling. Moreover, the strategy could be further improved by taking road types into consideration, which could be studied in future work.

APPENDIX

Four basic fusion methods that could be applied to our setting, e.g., fusing satellite images and partial road maps, are briefly introduced as follows. Note that we show the models with a uniform basic model, DeepLabV3+ [7].

A. Outputs Augmentation

Fig. 10 shows the model structure of outputs augmentation. In this case, the partial road maps are fused in a coarse way, and it does not require any extra parameters. Specifically, we train a basic DeepLabV3+, with the backbone of ResNet-34, and then take an average of the binarized outputs of DeepLabV3+ and partial road maps to obtain the fused outputs.

B. Outputs Fusion

As shown in Fig. 11, there are three more convolution layers than DeepLabV3+, where the two 3×3 convolution layers are used to convert the inactivated predictions and partial road maps into the same space, and another 1×1 convolution layer is used to fuse them.

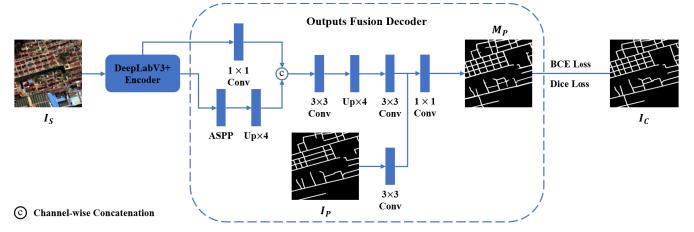


Fig. 11. Model structure of outputs fusion.

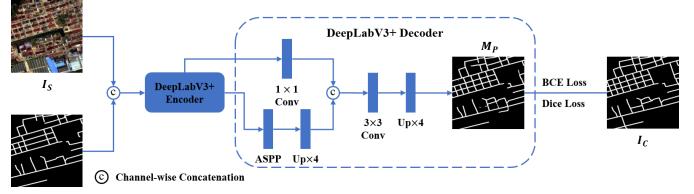


Fig. 12. Model structure of inputs fusion.

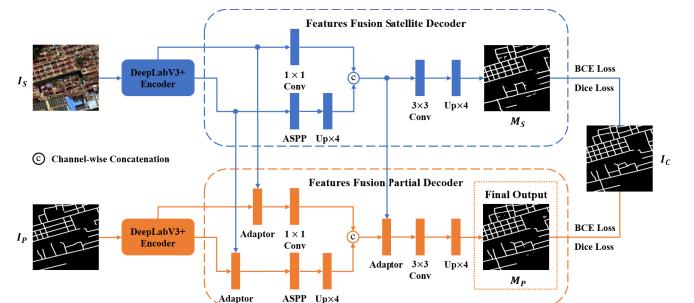


Fig. 13. Model structure of features fusion.

C. Inputs Fusion

The only difference between the basic DeepLabV3+ and the model of inputs fusion lies in the first convolution layers, where their numbers of input channels are 3 and 4. In inputs fusion, the satellite images and the corresponding partial road maps are concatenated along the channel to be the inputs. The partial road maps would influence the ability of extracting features significantly, as they are binary images and would be regarded as a kind of strong condition, using which the model would pay more attention on reserving existing roads. The model of inputs fusion is shown in Fig. 12.

D. Features Fusion

The number of parameters of features fusion is more than twice as large as that of DeepLabV3+. As shown in Fig. 13, the middle-layer low-level features are fused together to generate better results. In this case, it is clear that there are two branches, one for satellite images and another for partial road maps, and they could be jointly trained by two sets of loss functions. The satellite branch acts as a dedicated feature extractor, while the partial branch aims to not only extract useful information from partial road maps but also fuse the information of satellite images and partial road maps, to yield better outputs.

REFERENCES

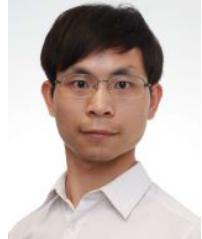
- [1] V. Mnih, *Machine Learning for Aerial Image Labeling*. Toronto, ON, Canada: Univ. of Toronto, 2013.
- [2] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, “Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.
- [3] G. Mattyus, W. Luo, and R. Urtasun, “DeepRoadMapper: Extracting road topology from aerial images,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3438–3446.
- [4] L. Zhou, C. Zhang, and M. Wu, “D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 182–186.
- [5] I. Demir et al., “DeepGlobe 2018: A challenge to parse the earth through satellite images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 172–181.
- [6] X. Li, Y. Wang, L. Zhang, S. Liu, J. Mei, and Y. Li, “Topology-enhanced urban road extraction via a geographic feature-enhanced network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8819–8830, Dec. 2020.
- [7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
- [8] P. P. Singh and R. D. Garg, “Automatic road extraction from high resolution satellite image using adaptive global thresholding and morphological operations,” *J. Indian Soc. Remote Sens.*, vol. 41, no. 3, pp. 631–640, 2013.
- [9] P. P. Singh and R. D. Garg, “A two-stage framework for road extraction from high-resolution satellite images by using prominent features of impervious surfaces,” *Int. J. Remote Sens.*, vol. 35, no. 24, pp. 8074–8107, Dec. 2014.
- [10] Z. Zhong, J. Li, W. Cui, and H. Jiang, “Fully convolutional networks for building and road extraction: Preliminary results,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1591–1594.
- [11] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [12] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.
- [13] 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.
- [14] R. Li, B. Gao, and Q. Xu, “Gated auxiliary edge detection task for road extraction with weight-balanced loss,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 5, pp. 786–790, May 2021.
- [15] 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.
- [16] A. Mosinska, P. Marquez-Neila, M. Kozinski, and P. Fua, “Beyond the pixel-wise loss for topology-aware delineation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3136–3145.
- [17] Y. Wei, Z. Wang, and M. Xu, “Road structure refined CNN for road extraction in aerial image,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 709–713, May 2017.
- [18] X. Gao et al., “An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network,” *IEEE Access*, vol. 6, pp. 39401–39414, 2018.
- [19] Q. Wu, F. Luo, P. Wu, B. Wang, H. Yang, and Y. Wu, “Automatic road extraction from high-resolution remote sensing images using a method based on densely connected spatial feature-enhanced pyramid,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3–17, 2021.
- [20] M. Zhou, H. Sui, S. Chen, J. Wang, and X. Chen, “BT-RoadNet: A boundary and topologically-aware neural network for road extraction from high-resolution remote sensing imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 168, pp. 288–306, Oct. 2020.
- [21] 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. 2017.
- [22] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, “The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 11–19.
- [23] 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, p. 2866, Sep. 2020.
- [24] Y. Xu, Z. Xie, Y. Feng, and Z. Chen, “Road extraction from high-resolution remote sensing imagery using deep learning,” *Remote Sens.*, vol. 10, no. 9, p. 1461, Sep. 2018.
- [25] T. Sun, Z. Di, P. Che, C. Liu, and Y. Wang, “Leveraging crowdsourced GPS data for road extraction from aerial imagery,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7509–7518.
- [26] H. Wu, H. Zhang, X. Zhang, W. Sun, B. Zheng, and Y. Jiang, “Deep-DualMapper: A gated fusion network for automatic map extraction using aerial images and trajectories,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1037–1045.
- [27] B. Parajuli, P. Kumar, T. Mukherjee, E. Pasiliao, and S. Jambawalikar, “Fusion of aerial LiDAR and images for road segmentation with deep CNN,” in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 548–551.
- [28] S. Ruan et al., “Learning to generate maps from trajectories,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 890–897.
- [29] F. H. Nahhas, H. Z. Shafri, M. I. Sameen, B. Pradhan, and S. Mansor, “Deep learning approach for building detection using LiDAR–orthophoto fusion,” *J. Sensors*, vol. 2018, Aug. 2018, Art. no. 7212307.
- [30] T. Sun, Z. Di, and Y. Wang, “Combining satellite imagery and GPS data for road extraction,” in *Proc. 2nd ACM SIGSPATIAL Int. Workshop AI Geographic Knowl. Discovery*, Nov. 2018, pp. 29–32.
- [31] K. K. Gadiraju, B. Ramachandra, Z. Chen, and R. R. Vatsavai, “Multimodal deep learning based crop classification using multispectral and multitemporal satellite imagery,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3234–3242.
- [32] T. G. Rudner et al., “Multi3Net: Segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 702–709.
- [33] R. Cao et al., “Integrating aerial and street view images for urban land use classification,” *Remote Sens.*, vol. 10, no. 10, p. 1553, 2018.
- [34] P. Benedetti, D. Ienco, R. Gaetano, K. Osé, R. Pensa, and S. Dupuy, “M3Fusion: A deep learning architecture for multiscale multimodal multitemporal satellite data fusion,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 2, pp. 4939–4949, 2018.
- [35] X. Li, L. Lei, Y. Sun, M. Li, and G. Kuang, “Collaborative attention-based heterogeneous gated fusion network for land cover classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3829–3845, May 2021.
- [36] H. Chen, M. Valerio Giuffrida, P. Doerner, and S. A. Tsaftaris, “Blind inpainting of large-scale masks of thin structures with adversarial and reinforcement learning,” 2019, *arXiv:1912.02470*.
- [37] A. Batra, S. Singh, G. Pang, S. Basu, C. V. Jawahar, and M. Paluri, “Improved road connectivity by joint learning of orientation and segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10385–10393.
- [38] Y. Zhang, X. Li, and Q. Zhang, “Road topology refinement via a multi-conditional generative adversarial network,” *Sensors*, vol. 19, no. 5, p. 1162, Mar. 2019.
- [39] J. Fu et al., “Dual attention network for scene segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.
- [40] Y. Zhang, S. Liu, C. Li, and J. Wang, “Rethinking the dice loss for deep learning lesion segmentation in medical images,” *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 26, no. 1, pp. 93–102, Feb. 2021.
- [41] A. Tran, A. Zonoozi, J. Varadarajan, and H. Kruppa, “Pp-linknet: Improving semantic segmentation of high resolution satellite imagery with multi-stage training,” in *Proc. 2nd Workshop Structuring Understand. Multimedia Heritage Contents*, 2020, pp. 57–64.
- [42] A. V. Etten, “City-scale road extraction from satellite imagery v2: Road speeds and travel times,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1775–1784.
- [43] F. Bastani et al., “RoadTracer: Automatic extraction of road networks from aerial images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4720–4728.
- [44] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, “SpaceNet: A remote sensing dataset and challenge series,” 2018, *arXiv:1807.01232*.
- [45] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

- [46] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*.
- [47] A. Kirillov, K. He, R. Girshick, and P. Dollár, "A unified architecture for instance and semantic segmentation," Universität Heidelberg, Heidelberg, Germany, Tech. Rep., 2017.
- [48] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
- [49] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.
- [50] G. Zhou, W. Chen, Q. Gui, X. Li, and L. Wang, "Split depth-wise separable graph-convolution network for road extraction in complex environments from high-resolution remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2021.



Qianxiong Xu received the bachelor's degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore.

His research interests include data mining, remote sensing, and computer vision.



Cheng Long (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2015.

From 2016 to 2018, he worked at Queen's University Belfast, Belfast, U.K. He is currently an Assistant Professor at the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore. His research interests include data management, data mining, and big data analytics.



Liang Yu received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2008.

He did the post-doctoral research at the National University of Singapore, Singapore; the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Champaign, IL, USA; and the Singapore-MIT Alliance for Research and Technology, Singapore. He had been leading the Smart Mobility Group, Institute of Infocomm Research, Singapore, where he initiated many research efforts in transportation and urban planning. He is currently a Senior Data Scientist with Alibaba Cloud Computing, Alibaba, Hangzhou, China, where he is leading the algorithm development for the City Brain Project. His research interest has covered a wide range of smart city topics such as geospatial data integration, data semantics, and data analytics.



Chen Zhang (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2015, supervised by Prof. Lei Chen.

He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University (PolyU), Hong Kong. Before joining PolyU, he worked as a Senior Manager at the Big Data Institute, HKUST. He is broadly interested in crowdsourcing, fintech, and machine learning.