BSSNet: Building Subclass Segmentation From Satellite Images Using Boundary Guidance and Contrastive Learning

Haofeng Xie, Xiangyun Hu, Huiwei Jiang, and Jinming Zhang

***Abstract*—Building subclass segmentation, aimed at predicting classes of buildings (high-rise zone, low-rise zone, single high-rise, and single low-rise) from satellite images, is beneficial in numerous applications, including human geography, urban planning, and humanitarian aid. However, problems, such as complex scenes and similar characteristics of different building categories make it difficult for general models to balance the accuracy of localization and classification in building subclass segmentation. Therefore, this article proposes a novel network for building subclass segmentation called building subclass segmentation network (BSSNet), which uses two subnetworks to divide and conquer the problem. The first network guides the building locations through binary building segmentation, called localization network. The spatial gradient fusion module in the localization network improves the binary segmentation result by supervising the spatial gradient map of prediction. The second network is a classification network, which predicts building subclasses. Intermediate features of the second network are optimized by contrastive learning loss to improve feature consistency. Finally, predictions of the two networks are combined to obtain the final result. The experimental results demonstrate that our BSSNet can perform significant improvements on the Hainan dataset we produced and the xBD dataset. In particular, the BSSNet achieves the best performance compared to current methods on the Hainan dataset.**

***Index Terms*—Building subclass segmentation, contrastive learning loss, convolutional neural network (CNN), feature fusion, satellite image, spatial gradient fusion (SGF).**

I. Introduction

BUILDING segmentation is widely studied in the field of remote sensing. Usually, most studies [1], [2], [3], [4] focus on binary building segmentation (whether the pixel is a building). Still, users need to know building subclass information (what type of building the pixel belongs to) in many applications. However, as a meaningful extension of building segmentation, automatic segmentation of building subclasses has rarely been studied. As shown in Fig. 1, building subclasses data provide information, such as building location and category, which can be of great help to many fields, including human geography [5], urban planning [6], and humanitarian aid [7]. But most of the building subclass data used in these fields comes from manual labeling, which is slow, costly, and laborious. Therefore, accurate and efficient automatic segmentation of building subclasses will be convenient for these fields. However, problems, such as within-class feature variation and between-class feature similarity make it difficult for general semantic segmentation networks to maintain localization and classification accuracy simultaneously.

Nowadays, studies on building subclass segmentation either combine images from different angles [8] or incorporate shadow detection with high-resolution images [9]. Damage assessment [10], [11], [12] is also a branch of building subclass segmentation, which classifies the damage level of buildings by using pre and postdisaster images. However, few studies focus

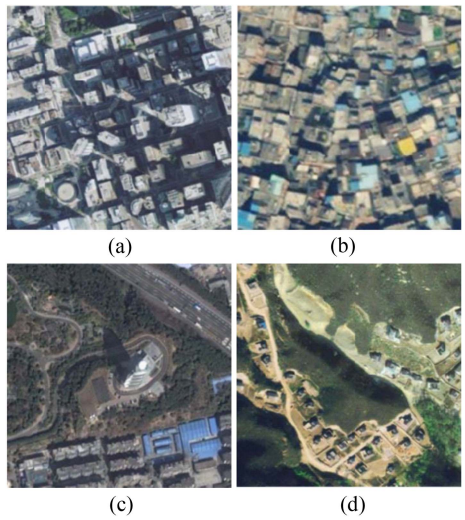

Fig. 1. Examples of the four classes of our building subclass segmentation task. (a) HZ. (b) LZ. (c) SH. (d) SL.

Manuscript received 15 June 2022; revised 8 August 2022; accepted 22 August 2022. Date of publication 29 August 2022; date of current version 15 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41771363 and Grant 92038301, and in part by the Special Fund of Hubei Luojia Laboratory under Grant 220100028. *(Corresponding author: Xiangyun Hu.)*

Haofeng Xie is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: xiehaofeng@whu.edu.cn).

Xiangyun Hu is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China, and also with the Hubei Luojia Laboratory, Wuhan 430079, China (e-mail: huxy@whu.edu.cn).

Huiwei Jiang is with the National Geomatics Center of China, Beijing 100044, China (e-mail: huiwei_jiang@whu.edu.cn).

Jinming Zhang is with the Key Laboratory of Network Information System Technology, Institute of Electronic, and The Aerospace Information Research Institute, Chinese Academic of Sciences, Beijing 100190, China (e-mail: nicnyzjm@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3202524

on using a single image to segment buildings into subclasses. Motivated by the abovementioned circumstances, we study a method specifically for building subclass segmentation, which classifies pixels in five classes, including four classes shown in Fig. 1 and background. The presentation of our method can fill the gap of building subclass segmentation and provide a large amount of accurate data quickly for the fields that need building subclass information for analysis.

Generally, experts identify the building class by the arrangement, the density, the shape, and the texture of buildings. Nevertheless, in automatic segmentation of building subclasses, the complexity of the task makes it difficult for a single network to ensure accurate classification and localization simultaneously. When the problem is decomposed, one network will focus on only one problem and learn more effectively. Therefore, we adopt a divide-and-conquer approach by using two networks to perform different tasks separately and combining them in terms of features and predicted results. Specifically, we divide the task into the following two parts: 1) binary building extraction; 2) building subclass extraction. The purpose of the division is to disassemble the task so that only the accuracy of the building localization is concerned in the first task, not the multiclass extraction accuracy, and vice versa. We add the feature fusion module (FFM) to reinforce the link between these two models. The boundary information can make the localization more accurate, so we propose the spatial gradient fusion (SGF) module to improve the boundary by refining the spatial gradient map. The subtle difference between building classes is also one of the reasons why segmenting building subclasses is difficult. Therefore, we introduce the contrastive learning loss to improve the representation of features in the building subclass segmentation network (BSSNet). Lastly, predictions of the two tasks are merged by intersection.

Our main contributions can be summarized as follows.

1) We propose a novel BSSNet that has two subnetworks for building subclass segmentation by combining binary building segmentation and multiclass building segmentation.
2) In the BSSNet, an SGF module is proposed to refine boundaries of binary building segmentation, while the pixel contrastive learning loss is introduced to enhance the representation of features in multiclass building segmentation.
3) The Hainan building subclass dataset, we proposed enriches the datasets for building subclass segmentation, which can help the research in subclass object extraction or fine-grain land cover classification from remote sensing images. The dataset can be accessed at.[1]
4) We demonstrate that the BSSNet achieves the SOTA performance on the Hainan dataset. On the xBD building damage assessment dataset [10], BSSNet is as good as SOTA methods. Moreover, abundant ablation experiments of BSSNet's components prove their effectiveness.

[1][Online]. Available: https://github.com/Xxxxiahaofeng/The-Hainan-Building-Subclass-Dataset

II. Related Work

A. Building Segmentation

Building segmentation is one of the most popular research focuses in remote sensing information extraction. Novel machine learning and remote sensing technologies have allowed automatic building segmentation, reducing manual work in recent years. Nevertheless, building segmentation remains a long-term challenge in remote sensing because of buildings' complex appearance in complicated environments.

Traditional building segmentation of aerial and remote sensing imagery always uses manual design features, such as color [13], texture [14], edge [15], [16], and spectrum [14], [17]. However, these features may vary significantly due to the indeterminacy of light, shooting angle, and sensors. With the development of CNNs, deep-learning-based methods have been broadly utilized to segment buildings on remote sensing images [1], [2], [3], [4]. With multilayer convolution, CNN can obtain multiscale and more robust features than artificially designed features. Yuan et al. [2] integrated features from multiple scales and combined the building boundaries to improve the performance of building segmentation. Maggiori et al. [3] designed a new architecture and a two-step training approach to solve the inaccurate training data problem. To acquire precise building boundaries, Bischke et al. [4] proposed a multitask network predicting segmentation and distance masks simultaneously.

B. Building Subclass Segmentation

Although the automatic recognition of building subclass is meaningful to urban planning [6], humanitarian aid [7], and other fields [5], few of studies focus on building subclass segmentation. Peng et al. [8] try to detect built-up areas by using stereo imagery incorporates height information. Taoufiq et al. [18] and Huang et al. [19] focus on building subclass classification. Sirmacek et al. [9] incorporate shadow detection with high-resolution images. However, no current methods segment the building subclasses with a single optical remote sensing image.

xBD [10] presents a task to assess building damage level, which can be considered an extension of building subclass segmentation. Various methods have been proposed to evaluate building damages [11], [12], which uses a two-stream CNN architecture for pre and postdisaster images. Nonetheless, building damage assessment uses sequential images to assess the building damage level by comparing images before and after a disaster.

We extend building segmentation from binary to subclass, as shown in Fig. 1. We propose a two-stream end-to-end network. One segment performs multiclass segmentation, and the other predicts binary building location to provide localization guidance.

C. Boundary Detection

Boundary detection is fundamental in various areas, such as semantic segmentation [20], object detection [21], and remote sensing image processing [22], [23], [24], [25]. Xie et al. [26] proposed an end-to-end multiscale boundary detection network.

CASENet [27] claimed a new task called semantic boundary detection, desiring at finding category-aware boundaries. Cheng et al. [21] employed boundary detection as a multitask network to improve the result of object segmentation. Zhen et al. [20] combined semantic boundary detection and semantic segmentation using the spatial gradient to improve the boundary pixel accuracy.

In the field of remote sensing, edge detection is also widely used to improve the effect of building detection. Jung et al. [22] adopted HED [26] and combined the boundary and segmentation mask to obtain an enhanced segmentation result. To improve building extraction, He et al. [24] embedded the boundary detection task into their framework by using spatial variation fusion to couple these two tasks.

Our methods follow the idea of combining boundary and segmentation in a multitask way to enhance the accuracy of building location. Instead of concatenating features or using postprocess, we concatenate the spatial gradient of segmentation and boundary to improve the mask boundary in an efficient way.

D. Contrastive Learning

Contrastive learning is one category of self-supervised learning [28], whose core goal is to discover discriminative representations. Another category of self-supervised learning is generative learning [29], [30], [31], [32], whose primary purpose is to generate feature vectors that can retain essential parts of the original data and reconstruct the original data. Contrastive learning considers representation learning from a different aspect: learn to compare [28]. In this way, contrastive learning avoids pixel-level learning and is more stable. Through noise contrastive estimation [33], contrastive methods learn meaningful representations by attracting positive pairs and repulsing negative pairs. Recently, many methods focus on constructing positive and negative sets [34], [35], [36], [37]. Hadsell et al. [35] first regarded contrastive learning as a dictionary lookup. He et al. [34] developed this method by building a dynamic dictionary with a queue and a moving-averaged encoder. Khosla et al. [38] extended the self-supervised batch contrastive approach to a fully supervised learning task, allowing the effective leverage of label information. Normalized embeddings from the same class are drawn tighter than those from other classes. Latest works address contrastive learning in dense image prediction [39], [40], [41]. Wang et al. al. [39] implemented supervised contrastive learning at the pixel level for semantic segmentation.

There are also some methods that use contrastive learning in remote sensing area tasks [42], [43], [44]. For instance, contrastive learning has been used, for example, in the hyperspectral image (HSI) classification to solve the small-sample problem of HSIs. Meanwhile, it has been adopted in synthetic aperture radar image classification to overcome insufficient labeled data [45], [46]. However, few methods apply contrastive learning to remote sensing image segmentation.

Given the complexity of remote sensing image scenes, even objects in the same class may differ vastly in their embeddings, making the application of semantic segmentation in remote sensing images difficult. Hence, we employ contrastive learning to gather clusters of pixel embeddings belonging to the same category while pushing apart different categories' embeddings.

III. Proposed Method

A. Network Architecture

1) Architecture Overview: Fig. 2 gives an overview of the procedure of BSSNet, which consists of te following two parts: the classification network (i.e., the upper one) and the localization network (i.e., the lower one). We exploit HRNet [47] as the backbone for the localization and classification network, respectively. HRNet can be divided into four stages according to the number of branches and resolutions. The stage n includes n branches corresponding to n resolutions. For ease of presentation, we simplify each stage to its number without showing the details of each stage in Fig. 2.

The localization network predicts the binary mask of building objects from images, in which building objects can be predicted intactly and shapely. The localization network first concatenates feature maps from the HRNet backbone. It feeds them into the predictor, which uses the 3×3 convolution (3×3 Conv), followed by a batch normalization (BN) layer and ReLU to reduce the feature dimension to 256, and then a 1×1 convolution is used to acquire mask predictions. Although the predicted masks can provide relatively accurate location of buildings, the predictions can still be rough and fuzzy due to ignoring of boundary information.

The before-mentioned issue could be primely alleviated by providing improved localization and guidance while employing building boundaries. Therefore, to utilize boundary information, we propose a boundary-predicting head. It realizes boundary prediction using the same predictor as the binary building mask predictor and boundary of binary ground truth as ground truth. However, simply adding a boundary-predicting head cannot potently pass boundary information to mask predictions. Thus, the SGF module is proposed, which combine the spatial gradient of mask predictions and predictions of the boundary-predicting head to obtain the final boundary predictions. It will be explained in detail later.

Likewise, the classification network employ the same predictor with different class numbers to generate the building subclass segmentation prediction. After the backbone network, we add up the projection head. The projection head outputs 256-dimensional features, and these features will be used in the contrastive learning loss. The role of contrastive learning loss is clustering features from the same class and scattering features from different classes.

In addition, we fuse two features of each HRNet in the same resolution with a simple FFM. The FFM consists of concatenation and two convolution blocks (3×3 Conv+BN+ReLU). It is a simple yet effective module to exchange information between the localization network and the classification network.

Finally, the binary building prediction and the building subclass prediction are combined to get the final prediction. We simply combine the intersection of nonbackground parts of the two predictions, and take the predicted values of building

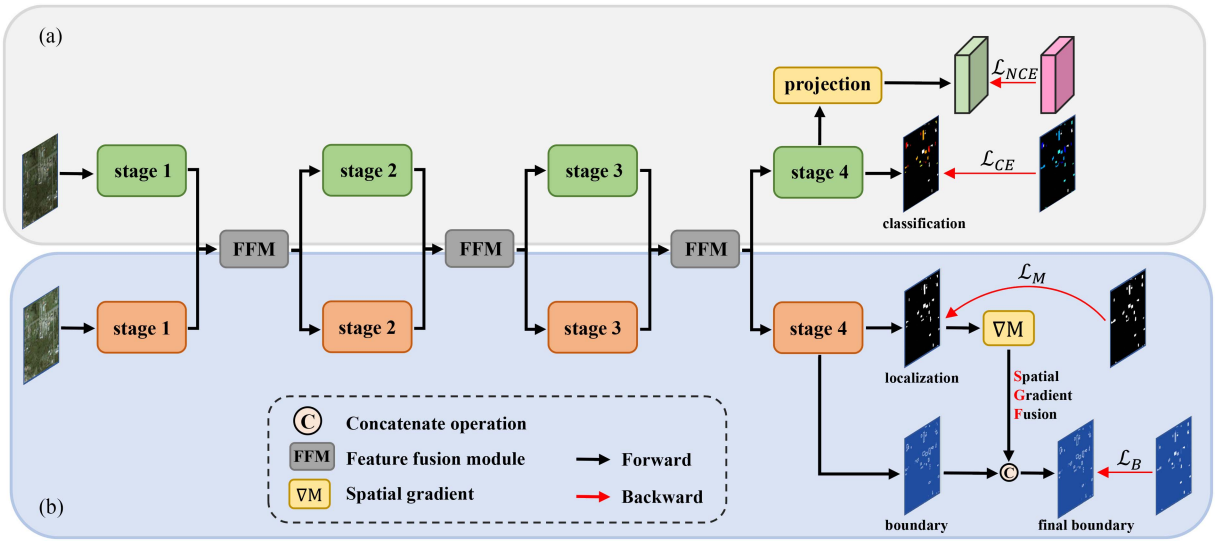

Fig. 2. Overview of our proposed method. There are four stages in our network. (a) Classification network for building subclass segmentation. (b) Localization network predicting binary mask and boundary of buildings. The FFM module is used to interact between the two networks. The spatial gradient ∇M derived from the building mask is concatenated with boundary prediction to get the final boundary prediction.

subclasses corresponding to these nonbackground pixels as the final prediction result.

2) SGF Module: An accurate boundary is essential to ensure the building binary segmentation result, which can make neighboring buildings effectively separated. Recently, most methods using boundary information to improve the segmentation effect have added boundary-predicting branches. However, these methods do not use boundary ground truth to supervise building binary segmentation, which makes the use of the boundary information ineffective. Therefore, we propose to combine the results of the boundary prediction branch with the spatial gradient of binary segmentation results to obtain the final boundary prediction results to learn the boundary information simply and directly.

From the boundary-predicting head and the mask-predicting head of the localization network, we can generate the boundary probability map $B \in \mathbb{R}^{H\times W\times 1}$, and the mask probability map $M \in \mathbb{R}^{H\times W\times 1}$, respectively. Then, we can obtain the mask boundary easily by spatial gradient deriving. Here, we use adaptive pooling to derive spatial gradient ∇M, which is

$$\nabla M(i,j) = |M(i,j) - \text{pool}_k(M(i,j))| \tag{1}$$

where i and j are the coordinates of the mask probability prediction and $|\cdot|$ denotes the norm function. pool_k is an adaptive average pooling operation with kernel size k. k can control the width of generated boundary ground truth. The default setting of k is 3.

To supervise the mask boundary directly and efficiently, we concatenate the boundary probability map B and derived boundary map ∇M. The concatenated map is assigned into a convolution layer to get the final boundary map, which will calculate loss with ground truth in the boundary loss function. This process can be formulated as

$$\boldsymbol{b} = \text{conv}(B \oplus M) \tag{2}$$

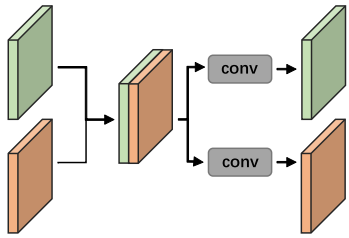

Fig. 3. FFM links the localization and classification networks. Conv denotes that a convolution block (3×3 Conv + BN+ReLU) does not change feature dimensions.

where $\oplus$ is the concatenation operation, and conv is a simple convolution layer. The final boundary prediction map is $\boldsymbol{b}$. In this way, we can simultaneously supervise the boundary accuracy of mask prediction and boundary prediction. Moreover, few impurities exist in the building outline, which can also be continuous.

3) FFM: As shown in Fig. 3, FFM uses concatenation and two convolution blocks (3×3 Conv+BN+ReLU). The output features can be formulated as

$$\tilde{X}^s_{\text{loc}} = f^s_{\text{loc}}(X^s_{\text{cls}} \oplus X^s_{\text{loc}}) \tag{3}$$

$$\tilde{X}^s_{\text{cls}} = f^s_{\text{cls}}(X^s_{\text{cls}} \oplus X^s_{\text{loc}}) \tag{4}$$

where $\tilde{X}^s_{\text{loc}}$ and $\tilde{X}^s_{\text{cls}}$ are the localization feature and the classification feature after FFM in stage s. X^s_{loc} and X^s_{cls} remark the localization feature and the classification feature before FFM in stage s. f^s_{loc} and f^s_{cls} denote two convolution blocks of FFM in stage s. $\oplus$ means the concatenation operation.

FFM links two subnetworks to build bilateral information exchanges. This module makes the classification network focus more on areas predicted as buildings and lets the localization network be more robust to various constructions.

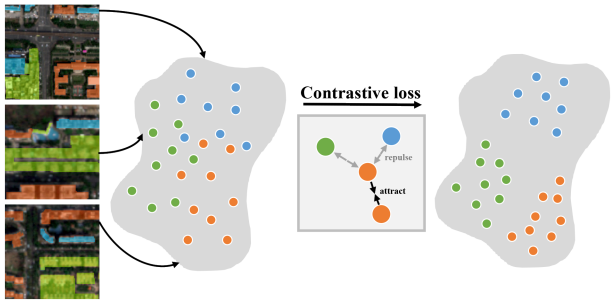

Fig. 4. Main idea of pixel contrastive loss is to extract embeddings of different classes with randomly selected pixels. Then pixel contrastive loss repels the embeddings of different classes while attracting the embeddings of the same class.

B. Loss Function

1) Pixel Cross-Entropy Loss: We can obtain the logit prediction $\boldsymbol{y} \in \mathbb{R}^{HW \times C}$, i.e., the unnormalized prediction vector from the last layer of the network. In the classical semantic segmentation cross-entropy loss function, $\boldsymbol{y}$ is normalized using softmax. Then, it is multiplied with the one-hot vector of ground truth $\hat{\boldsymbol{y}} \in \mathbb{R}^{HW \times C}$

$$\mathcal{L}_{\text{CE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\hat{\boldsymbol{y}}^T \log(\text{softmax}(\boldsymbol{y})). \tag{5}$$

However, it computes loss pixel by pixel, so it does not consider the relationship between pixels. This may result in different classes of pixels with very similar characteristics that are difficult to distinguish. Thus, we propose pixel contrastive loss to cluster pixels in the same class and push away pixels in different classes, as shown in Fig. 4.

2) Pixel Contrastive Loss: First, we introduce InfoNCE Loss in unsupervised representation learning. Unsupervised representation learning aims to train an encoder, which generates effective image embedding (feature vectors) $\boldsymbol{v}_I$ of image I. Contrastive learning is the current mainstream way to achieve this goal.

In contrastive learning, $\boldsymbol{v}_I$ should be similar to the positive embedding $\boldsymbol{v}_I^+$ (feature vectors of the same augmented image I) and dissimilar to embedding $\boldsymbol{v}^-$ in negative embedding set $\mathcal{N}_I$ (feature vectors of other images). Driven by this motivation, InfoNCE is the commonly used contrastive learning loss function

$$\mathcal{L}_{\text{NCE}}^I = -\log \frac{\exp(\boldsymbol{v}_I \cdot \boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{v}_I \cdot \boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^- \in \mathcal{N}_I} \exp(\boldsymbol{v}_I \cdot \boldsymbol{v}^-/\tau)} \tag{6}$$

where $\boldsymbol{v}^-$ is the negative embedding in $\mathcal{N}_I$, $\cdot$ is the inner product operation, and τ is the temperature hyperparameter.

Now, we extend the InfoNCE loss to the pixel level. Thus, positive embeddings imply pixel embeddings in the same class, while negative embeddings are pixel embeddings of different classes. Our goal is to attract these positive embeddings and repulse negative embeddings.

We assume the embedding of pixel i as $\boldsymbol{v}_i \in \mathbb{R}^D$, where D means the dimension of the embedding. $\mathcal{P}_i$ and $\mathcal{N}_i$ are pixel embedding sets of the positive and negative samples for pixel i, respectively; i.e., $\mathcal{P}_i$ is the embedding set that the class of samples is the same as pixel i, and vice versa. Accordingly, our pixel contrastive loss is defined as

$$\mathcal{L}_{\text{NCE}} = \sum_{i \in I} \sum_{\boldsymbol{v}^+ \in \mathcal{P}_i} -\log \frac{\exp(\boldsymbol{v}_i \cdot \boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{v}_i \cdot \boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^- \in \mathcal{N}_i} \exp(\boldsymbol{v}_i \cdot \boldsymbol{v}^-/\tau)}. \tag{7}$$

Note that the positive and negative samples come from an identical batch of pixel i. Furthermore, all embeddings are normalized before sending into the pixel contrastive loss.

Finally, we generate the overall loss function of the classification network by adding $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{NCE}}$

$$\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{NCE}} \mathcal{L}_{\text{NCE}} \tag{8}$$

where λ_{NCE} is the weight to control the importance of $\mathcal{L}_{\text{NCE}}$.

3) Boundary Loss: We view boundary prediction as a binary semantic segmentation problem, similar to the practice of joint boundary detection. We concatenate the prediction of the boundary-predicting head and the spatial gradient of mask prediction to acquire the final boundary prediction. We can obtain complete results if we simply supervise it by using binary cross-entropy loss. However, because the proportion of boundary pixels in each image is changing, even if the positive sample weight is carefully set, the response degree of boundary prediction is still not high. Dice loss [48] avoids the difficulty of setting positive sample weight by directly optimizing the $F1$ score, but due to its instability, its trained boundary is often incomplete. Therefore, we combine these two types of loss and use their complementary to improve the effect of boundary prediction.

To generate soft boundaries from the ground truth of the binary mask, we utilize the Laplacian operator. The Laplacian operator is a second-order gradient operator for generating boundaries. The generated soft boundary maps are converted to binary maps by a threshold value of 0.

We utilize binary cross-entropy loss and dice loss to improve the learning of boundaries. Dice loss calculates the ratio of overlaps between prediction and ground truth, independent of the number of foreground/background pixels. We define boundary loss $\mathcal{L}_B$ as follows:

$$\mathcal{L}_{\text{boundary}}(\boldsymbol{b}, \hat{\boldsymbol{b}}) = \mathcal{L}_{\text{BCE}}(\boldsymbol{b}, \hat{\boldsymbol{b}}) + \lambda_{\text{Dice}} \mathcal{L}_{\text{Dice}}(\boldsymbol{b}, \hat{\boldsymbol{b}}) \tag{9}$$

where λ_{Dice} is the weight to control the importance of $\mathcal{L}_{\text{Dice}}$. $\boldsymbol{b}, \hat{\boldsymbol{b}} \in \mathbb{R}^{H \times W}$ are the prediction and ground truth of boundary, respectively. Dice loss λ_{Dice} is given as follows.

$$\mathcal{L}_{\text{Dice}}(\boldsymbol{b}, \hat{\boldsymbol{b}}) = 1 - \frac{\sum_i^{\text{HW}} b_i \cdot \hat{b}_i + \epsilon}{\sum_i^{\text{HW}} b_i^2 + \sum_i^{\text{HW}} \hat{b}_i^2 + \epsilon} \tag{10}$$

where b_i and $\hat{b}_i$ are the ith pixel in boundary prediction and ground truth, respectively. $\cdot$ is the inner product operation. ϵ is added in numerator and denominator to ensure no zero division (default $\epsilon = 1$). This formula is similar to the $F1$ score at the pixel level.

4) Multitask Learning Loss: Our network accomplishes three tasks, namely binary building segmentation, building subclass segmentation, and building boundary detection. To train the network efficiently, we choose to train in a multitask learning

TABLE I
DISTRIBUTION OF BUILDING SUBCLASSES IN THE HAINAN DATASET

	HZ	LZ	SH	SL
Number(M)	7.622	15.291	11.389	1.310
%	21.10	42.92	32.36	3.62

Considering the way ground truth are labeled, we calculate the number at the pixel level.

TABLE II
NUMBER AND DISTRIBUTION OF BUILDING DAMAGE ANNOTATION IN THE XBD DATASET

	No damage	Minor	Major	Destroyed
Number	313003	36860	29904	31560
%	76.04	8.98	7.29	7.69

The number of each damage level's polygons is reported.

way. The definition of the overall multitask learning loss is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{boundary}}. \tag{11}$$

Note that $\mathcal{L}_{\text{loc}}$ is the binary cross-entropy loss used in the binary building segmentation task.

IV. EXPERIMENTS

A. Dataset and Evaluation Metric

1) Hainan Building Subclass Dataset: As far as we know, few public datasets specifically for remote sensing building subclass segmentation are available. Thus, to facilitate the training of our proposed method, we construct a subclass dataset for buildings in Hainan Province, China. The Hainan dataset we presented compensates for the absence of building subclass segmentation datasets. We will continue to expand this dataset as the research progresses. Four building subclasses exist in this dataset: high-rise zone (HZ), low-rise zone (LZ), single high-rise (SH), single low-rise (SL), which are identified by experts from the Shanxi provincial mapping agency.

The dataset contains 42 images with resolutions ranging from 0.8 to 2 m per pixel, and sizes ranging from 2000×2000 to 5000×6000. We crop the images to size 512×512 patches. This gives a total of 1348 image patches, divided into 70% for training and 30% for test. That is, we got a training set with 944 cropped images and a test set with 404 cropped images. The proportion of each category (ignoring background) is shown in Table I. The data are imbalanced, and the proportion of SL in the dataset is deficient. The reason is that the geographic distribution of images is concentrated in urban areas, where most low-rise buildings are clustered. To solve this imbalance problem, we set the class weight of CELoss [see (5)] of classification network to [1.0, 1.0, 1.0, 1.0, 10.0].

2) xBD Dataset: Since it is challenging to obtain datasets for building subclass segmentation, a building damage assessment dataset xBD [10] is employed to evaluate our proposed method. This dataset is a publicly available, large-scale satellite image dataset for building damage level assessment, which is similar to the task we are working on. While the difference is that the xBD dataset contains images before and after disasters, so the changes brought by disasters also should be concerned in the network.

Although change information should be concerned in the xBD dataset, building damage level is primarily evaluated using images after disasters, whereas images before disasters are inclined to locate buildings. In addition, building damage level can be viewed as a variation of building subclass. These characters are consistent with our proposed work, so we select the xBD dataset to evaluate the effectiveness of our work.

This dataset selects 19 diverse disasters in different locations (such as forest fires, earthquakes, floods, and hurricanes). The dataset contains pre and postdisaster image pairs with 1024×1024. Each image is in the visible spectral band (red, green, and blue) with a spatial resolution of 0.8 m. Four building damage levels exist: no damage, minor damage, major damage, and destroyed. Table II shows the number and distribution in the dataset.

3) Evaluation Metric: To evaluate the performance of our method, we perform qualitative and quantitative analyses in our experiments. We use the $F1$ score ($F1_b$) to evaluate the experiment results of binary building segmentation. And the harmonic mean of the $F1$ score ($F1_c$) of each building class is employed to evaluate the effectiveness of building subclass segmentation. The metrics are defined as follows:

$$F1_b = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{12}$$

$$F1_{c_i} = \frac{2\text{TP}_i}{2\text{TP}_i + \text{FP}_i + \text{FN}_i} \tag{13}$$

$$F1_c = \frac{n}{\sum_n^{i=1} 1/F1_{c_i}} \tag{14}$$

where TP, FP, and FN are the numbers of true-positive, false-positive, and false-negative pixels in segmentation results, respectively. n is the number of classes, and $F1_{c_i}$ is the $F1$ score of class i.

4) Experimental Settings: All the experiments are run on four GeForce GTX 2080Ti GPUs with PyTorch implementation. In training, we crop images to 512×512 patches. We use HRNet-32 as the backbone for our networks with pretrained weights downloaded from the PyTorch library. For pixel contrastive loss, we randomly select 1024 pixels in the same batch as positive and negative embedding sets in all experiments, and loss weight $\lambda_{\text{NCE}} = 0.1$. For boundary loss, we set the kernel size of the Laplacian operator to 3, and loss weight $\lambda_{\text{Dice}} = 1.0$. The model is trained using Adam optimizer with an initial learning rate of 0.0001. The batch size is 4 for 60 000 iterations on the Hainan dataset. The batch size is 8 for 100 000 iterations in xBD dataset. We reduce the learning rate by using the "poly" learning rate policy, in which the initial learning rate is multiplied by $(1 - \frac{\text{iter}}{\max_{\text{iter}}})^{\text{power}}$ and power $= 0.9$. Random crops and horizontal flip are also applied.

TABLE III
COMPARISON WITH DIFFERENT METHODS ON THE HAINAN BUILDING SUBCLASS DATASET

Method	Backbone	HZ ($F1$)	LZ ($F1$)	SH ($F1$)	SL ($F1$)	Overall $F1$
DeepLabv3+ [49]	ResNet-50	0.684	0.681	0.728	0.348	0.557
DeepLabv3+	ResNet-101	0.673	0.678	0.729	0.352	0.558
FPN [50]	HRNet32	0.666	0.686	0.732	0.360	0.563
FPN	HRNet48	0.680	0.690	0.743	0.366	0.571
OCR [51]	HRNet48	0.672	0.689	0.722	0.329	0.543
MANet [52]	ResNet-101	0.683	0.692	0.736	0.354	0.564
MCFINet [53]	ResNet-101	0.682	0.699	0.739	0.352	0.564
Ours(vanilla network)	HRNet32	0.693	0.709	0.755	0.349	0.568
Ours(single network)	HRNet32	0.711	0.719	0.758	0.371	0.587
Ours	HRNet32	**0.714**	**0.721**	**0.759**	**0.392**	**0.601**

B. Main Results

1) Hainan Dataset: To demonstrate the effectiveness of our proposed BSSNet, we first compared our method with several SOTA segmentation methods on the Hainan building subclass dataset.

1) Some popular semantic segmentation networks, including FPN [50] and DeepLabv3 [49], are employed for comparison.

2) OCR [51] uses HRNet48 as the backbone network. Features in different levels are concatenated, and the feature before using the OCR module is also used to generate auxiliary prediction.

3) MANet [52] and MCFINet [53] use ResNet101 as the backbone network. The last layer of features is directly used to predict segmentation results.

In addition to our full framework, the two subnetworks clean baseline without FFM, SGF module, and pixel contrastive loss, called the vanilla network, is also compared. A single network with two heads is also compared. The functions of the two heads are similar to those of the two subnetworks, which locate and classify, respectively, and SGF and pixel contrastive loss are also added. According to Table III, the proposed framework using the vanilla network alone can be competitive to existing methods in terms of all metrics. Bold entities emphasize that the current method achieves the best results on the corresponding metrics.

We also evaluate our method with popular and SOTA methods in the natural image and remote sensing image segmentation area. As shown in Table III, our proposed method outperforms these methods by an impressive $F1$ score. On the overall $F1$ metric, BSSNet produces a 4.0% improvement over the previous best results. Our vanilla network is only 0.2% below FPN on the overall $F1$ metric because our two-subnetwork framework divides and conquers the task and gives a better localization result. We also observe a 1.4% increase in overall $F1$ score when splitting the single network into two subnetworks framework, which again proves that our two-subnetwork framework is effective. Furthermore, with the help of other modules, including FFM, SGF, and contrastive loss, the performance of our method is significantly improved over the vanilla network. The SH $F1$ score of our BSSNet is 2.8% higher than that of FPN, which is due to the contrastive loss making features more robust, and the difference between features of the LZ and that of SL is more distinct.

Fig. 5 shows a visual comparison of the building subclass segmentation results of different networks. The predicted masks of our proposed method are more precise and highly coincident with their boundaries. Our method also better predicts small, isolated objects, such as SL buildings in rural areas. Additionally, in comparison with FPN and MCFINet, our method can better separate buildings close to each other.

TABLE IV
COMPARISON WITH DIFFERENT METHODS ON THE XBD DATASET

Method	No damage	Minor	Major	Destroyed	Overall $F1$	Binary $F1$
xBD baseline [10]	0.663	0.144	0.090	0.466	0.184	0.852
BDANet [56]	0.800	0.440	0.633	0.722	0.616	0.821
Weber et al. [55]	**0.906**	0.493	0.722	0.837	0.700	0.835
RescueNet [54]	0.885	**0.563**	0.771	0.808	**0.735**	0.840
Ours	0.904	0.533	**0.772**	**0.846**	0.733	**0.863**

2) xBD Dataset: Moreover, we present quantitative and qualitative comparisons of building disaster damage assessment on the xBD dataset. We compare our method with the xBD baseline [10], BDANet [56], RescueNet [54] and the method of Weber et al. [55], which are popular and typical methods in building disaster damage assessment. All results indicate that our method can be competitive in building disaster damage assessment.

In Fig. 6, we give qualitative results on a small but diverse sample of the dataset. From these results, our method appears to be remarkably better than the baseline model. The baseline model produces quite a few false positive and false negative errors eliminated by our model because the contrastive loss in our method makes features more consistent. In addition, the SGF module makes our prediction results more apparent and accurate at the boundary.

According to Table IV, our method produces 0.1% and 3.8% improvement in major and destroyed damage levels, respectively, but the overall $F1$ metric is 0.3% lower than the maximum value. Although the overall $F1$ is slightly lower than that of the method RescueNet, our method shows a significant improvement in binary $F1$(2.3%, 84.0%→86.3%), which is also reflected in Fig. 6. The reason is that we use a localization network to focus on binary segmentation, which is a relatively simple task, and an SGF module to improve the boundary of binary segmentation results. The overall $F1$ score is slightly lower than that of RescueNet because the network is not designed to take advantage of the differences between pre and postdisaster images.

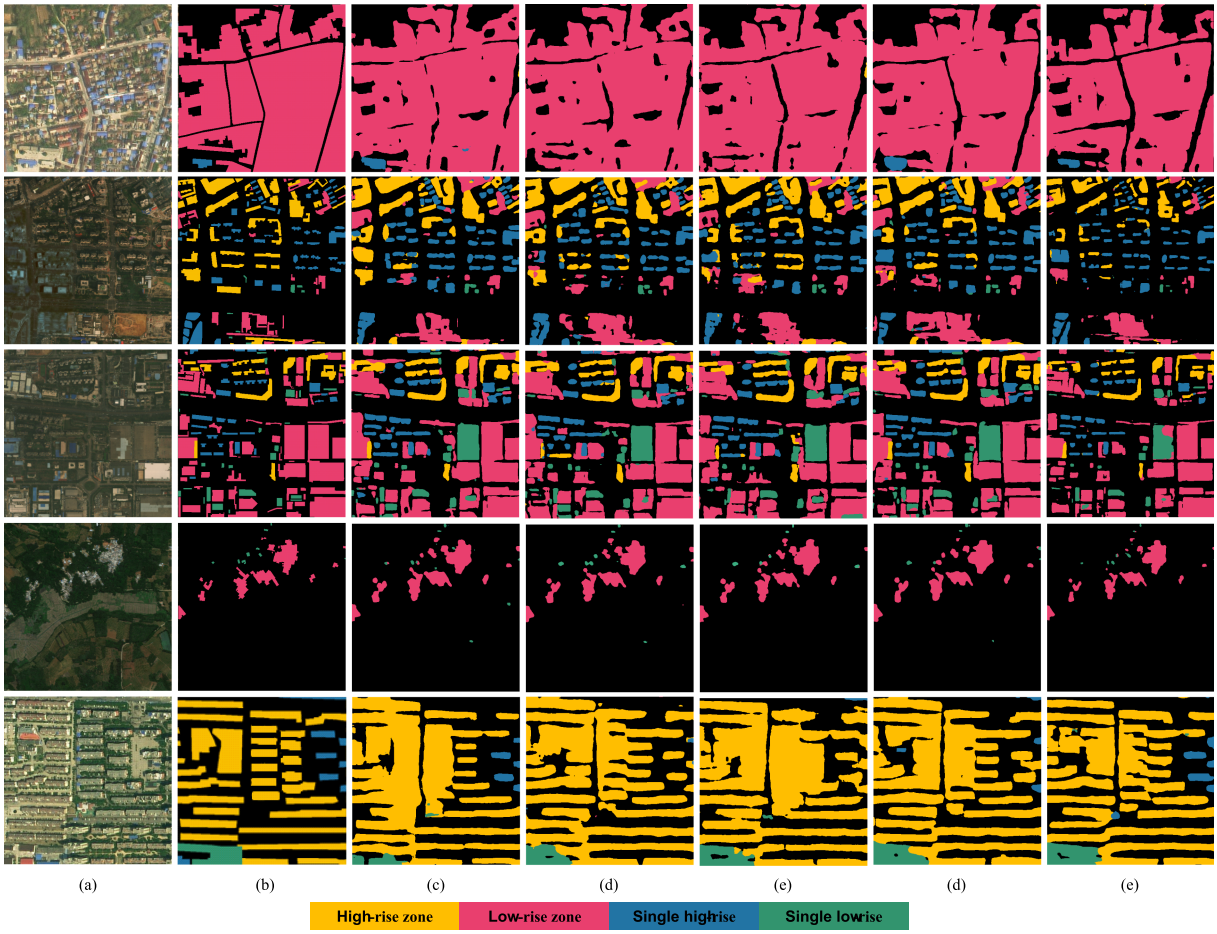

Fig. 5. Visual comparisons of building subclass segmentation results. From left to right: (a) image (b) ground-truth, (c) our proposed network, (d) FPN (HRNet48 backbone), and (e) MCFINet.

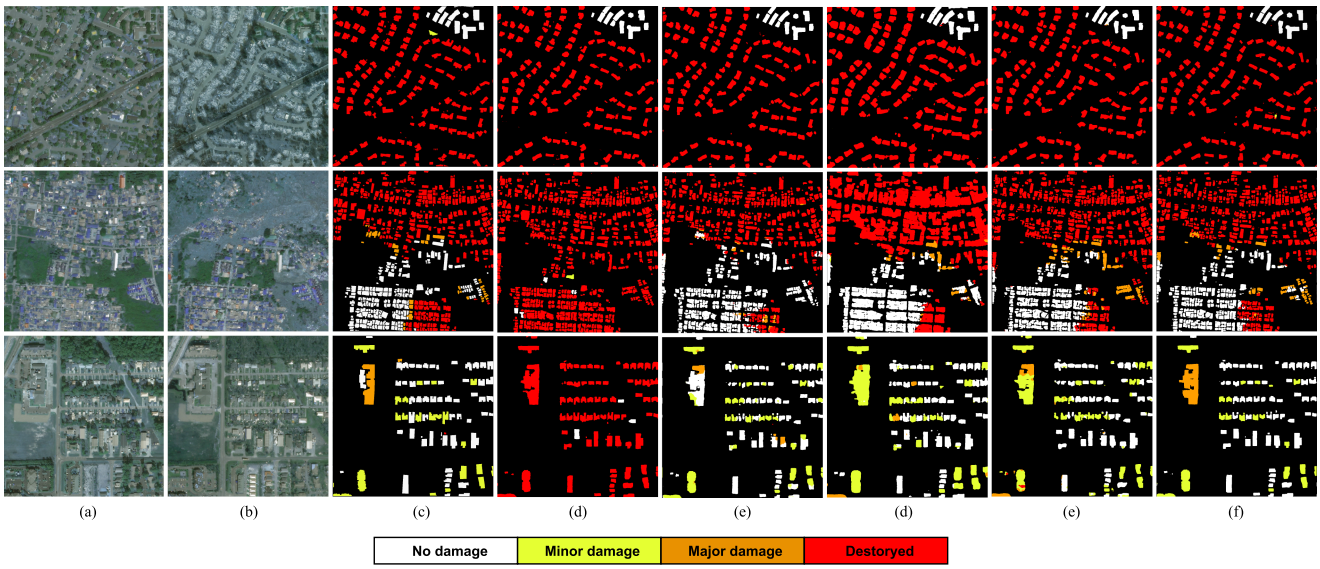

Fig. 6. Visual comparisons of building disaster damage assessment results. From left to right: (a) pre-image (b) post-image, (c) the ground-truth, (d) RescueNet [54], (e) Weber et al. [55], and (f) our proposed network.

TABLE V
ABLATION STUDY OF DIFFERENT COMPONENTS: FFM, SGF MODULE, CONTRASTIVE LOSS (CON) IN THE PROPOSED FRAMEWORK

	HZ ($F1$)	LZ ($F1$)	SH ($F1$)	SL ($F1$)	Overall ($F1$)
Vanilla	0.693	0.709	0.756	0.349	0.568
Vanilla + FFM	0.710	0.718	0.767	0.356	0.579
Vanilla + SGF	0.692	0.704	0.759	0.372	0.582
Vanilla + Con	0.711	0.720	0.756	**0.386**	0.596
Vanilla + FFM + SGF	0.706	0.720	**0.770**	0.348	0.573
Vanilla + FFM + Con	**0.712**	0.733	0.758	0.382	**0.597**
Vanilla + SGF + Con	0.709	**0.721**	0.766	0.379	0.593

TABLE VI
ABLATION STUDY OF ADDING FFM AFTER DIFFERENT STAGES (SHORTENED TO S) IN THE PROPOSED FRAMEWORK

	HZ ($F1$)	LZ ($F1$)	SH ($F1$)	SL ($F1$)	Overall ($F1$)
w/o FFM	0.693	0.709	0.755	0.349	0.568
$S1$	0.689	0.718	0.749	0.355	0.572
$S1 + S2$	0.707	**0.720**	0.760	0.352	0.575
$S1 + S2 + S3$	**0.710**	0.718	**0.767**	**0.356**	**0.579**

TABLE VII
ABLATION STUDY OF SAMPLING DIFFERENT NUMBERS OF PIXEL FEATURES IN CONTRASTIVE LOSS

sample	view	FLOP(G)	$F1$
512	100	17.560	0.592
1024	200	17.623	0.596
2048	400	18.005	0.591
4096	800	18.978	**0.603**

$F1$ denotes the overall $F1$ score.

TABLE VIII
ABLATION STUDY OF USING DIFFERENT PROJECT DIMENSIONS IN CONTRASTIVE LOSS

Dims	Params(M)	FLOP(G)	$F1$
128	0.293	14.511	0.588
256	0.354	17.623	**0.596**
512	0.478	23.881	0.583
1024	0.724	36.015	0.584

$F1$ denotes the overall $F1$ score.

C. Ablation Study

To understand how our proposed method works, we perform complete experiments to study its components. Table V shows the results of the vanilla network combined with each component individually. The FFM and SGF modules achieve 0.9% and 1.4% improvement in $F1_c$, respectively. The contrastive loss promotes overall performance by 2.8%. The addition of contrastive loss enhances the network's ability to distinguish categories with similar characteristics, thus increasing the $F1$ scores of HZ, LZ, and SL by 1.8%, 1.1%, and 3.7%, respectively. The influence of adding multiple modules is also explored. The FFM module along with the contrastive loss can bring an improvement in overall $F1$ score by 2.9% to the vanilla network. Each component will be analyzed more detailedly in the subsequent sections.

1) FFM in Different Stages: To validate the effect of FFM, we add FFM stage by stage, as shown in Table VI. With the addition of FFMs, the $F1$ score of a single category or the overall $F1$ score maintains an upward trend. The improvement of the overall $F1$ score is 0.4% when FFM is added to the first stage and 1.1% when FFM is added to all three stages. Through fusing classification and localization features, our network gives considerable attention to the location recognized as buildings. The $F1$ score of LZ fluctuates after FFM is added in different stages because manual labeling in the Hainan dataset tends to label LZ into a whole piece. Hence, accurate localization information may be of little help in this category.

2) Pixel Contrastive Loss: In (7), we use positive embedding set $\mathcal{P}_i$ and negative embedding set $\mathcal{N}_i$ to compute contrastive loss $\mathcal{L}_{\text{NCE}}$. The way of obtaining these two sets will greatly impact the network's performance. Simply using all pixels in the same batch may be computationally expensive. Accordingly, first, we randomly sample a specified number of pixel embeddings. Then, to make the embedding number of each class even, we set a hyperparameter named view number limiting the max embedding number of each class.

Table VII shows the result of using various sample numbers and view numbers. The network's computational cost and performance grow as the sample and view numbers increase. When sample number $= 4096$ and view number $= 800$, the network can achieve the best overall $F1$ score of 60.3%, but the efficiency is low at this time. Therefore, considering effect and efficiency, we choose sample number=1024 and view number=200 as our default setting.

Moreover, the number of dimensions of the projection head in our network is crucial. Thus, we study the effects of the number of embedding dimensions, as shown in Table VIII. The larger the number of dimensions is, the richer the embedded information is; otherwise, the less efficient the computation is. At dimension=256, the overall $F1$ score reaches the highest value of 59.6%. However, as the dimensions continue to increase, the network's performance decrease. At dimension=1024, the overall $F1$ score is lower at 58.4%. The reason is the excessive dimension of the embeddings, which are mixed with redundant information and thus affect the network's performance.

To study the effect of the contrastive loss, we perform experiments using different contrastive loss weights λ_{NCE}. As the result shown in Fig. 10(b), we can get the best overall $F1$ score when $\lambda_{\text{NCE}} = 0.1$. Interestingly, with the growth of weight, the overall $F1$ score remains constant at around 60.0%. This shows that, the contrastive loss can only bring limited help to learning after it has learned to a certain extent. This is because the features of

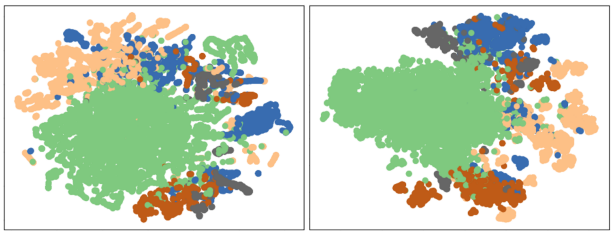

Fig. 7. Visualization of features learned with pixel-wise entropy loss (5) (left) and our pixel-wise contrastive loss (8) (right) on the Hainan dataset. Features' color is in accordance with class labels. As presented, features of contrastive loss have a result of clustering.

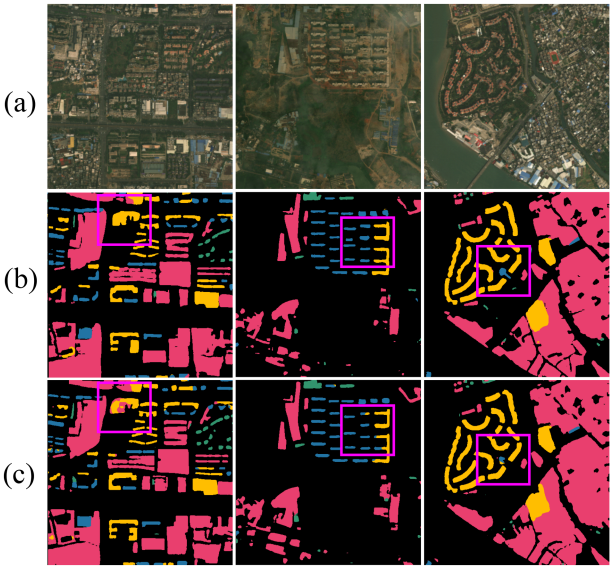

Fig. 8. Visualization results for analyzing the impacts of pixel contrastive loss on segmentation prediction. From top to down: (a) image, (b) contrastive loss, and (c) vanilla network.

different classes of buildings are similar, which makes it difficult for the network to learn new knowledge.

In addition, to understand the improvement from pixel contrastive loss, we use t-SNE [57] to visualize the embedding space before and after contrastive loss is added. Fig. 7 exhibits that, after the addition of contrastive loss, the boundary between features of different categories is more apparent, or the clustering of embeddings of the same category is more compact. As a result, the network can better distinguish different categories. From the predicted masks in Fig. 8, with pixel contrastive loss, our method is capable of producing a more accurate segment.

3) Boundary Loss: Although additional constraints on the boundary can improve the segmentation effect, learning the boundary with different loss functions will have a great impact on the result. Table IX reports the influence of boundary constraints with different loss functions on the $F1$ score of binary segmentation. BCE, weighted BCE and Dice loss yield about 0.3%, 0.4%, and 0.2% raise in binary $F1$ score, respectively. The biggest gains of 0.6% were made by utilizing BCE and Dice loss together in the network.

To find out how the combined Dice loss and BCE loss lead to such competitive advantages, we analyze the visualization

TABLE IX
ABLATION STUDY OF USING DIFFERENT BOUNDARY LOSSES IN THE LOCALIZATION NETWORK

Loss function	Vanilla	BCE	W-BCE	Dice	Dice-BCE
Binary $F1$	0.898	0.901	0.902	0.900	**0.904**

Vanilla and W-BCE denote our vanilla network and weighted BCE, respectively.

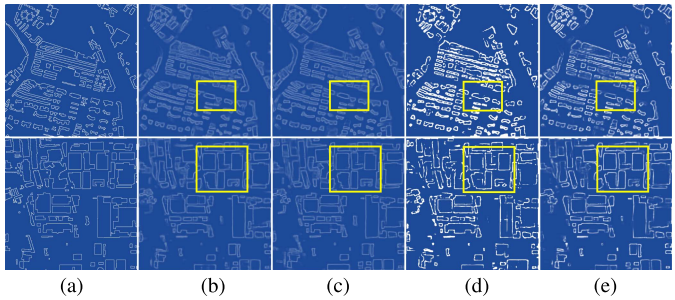

Fig. 9. Visualization results for analyzing the impacts of different loss functions to boundary prediction. From left to right: (a) ground truth, (b) BCE loss, (c) weighted BCE loss, (d) Dice loss, and (e) Dice-BCE loss.

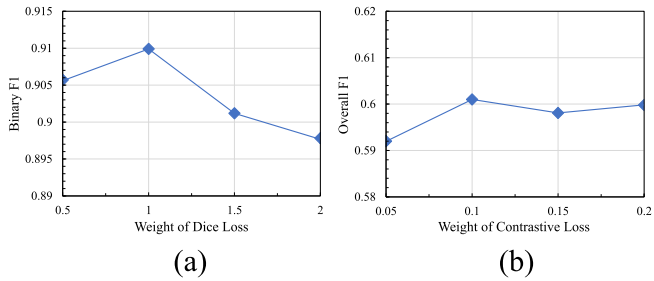

Fig. 10. Ablation Studies on the Hainan dataset with different Dice loss weights λ_{Dice} and contrastive loss weights λ_{NCE}. From left to right. (a) Binary $F1$ score with different Dice loss weights λ_{Dice}. (b) Overall $F1$ score with different contrastive loss weights λ_{NCE}.

results using different boundary loss functions in detail. As shown in Fig. 9, weighted BCE tries to solve the data imbalance problem by applying balancing weights. Still, this hard balancing carries few promotions because the proportion of boundary pixels of different images fluctuates over a wide range, and fixed weights cannot cope with this problem. Dice loss makes boundaries clearer, but buildings are incomplete because it regards loss function as the $F1$ score, thus ignoring the influence of a single pixel. Consequently, combining Dice loss and BCE loss can generate precise, clear, and complete building boundaries by combining the advantages of both to complement each other. It is worth noting that Dice loss is much brighter than the boundary lines in the other comparisons. According to (10), we can get the gradient of boundary prediction: $\hat{\boldsymbol{b}}(\boldsymbol{b}^2 - \hat{\boldsymbol{b}}^2)/(\boldsymbol{b}^2 + \hat{\boldsymbol{b}}^2)^2$, which is much stricter to wrong predictions than that of BCE loss. Therefore, compared with BCE loss, the correct prediction of Dice loss tends to be 1, while the wrong prediction tends to be 0, resulting in the brighter prediction.

We conduct experiments in Fig. 10(a) to study the impact of different Dice loss weights on the boundary loss function.

This weight parameter λ_{Dice} has a certain impact on the performance. We find that $\lambda_{\text{Dice}} = 1.0$ achieves the best binary building segmentation performance. Large λ_{Dice} will cause the network to focus too much on the boundary, leaving the incomplete prediction.

V. Conclusion

This article proposes a CNN-based learning framework with two subnetworks named BSSNet for building subclass segmentation from satellite images. The first network is used for binary building segmentation and guides the building locations in building subclass segmentation. An SGF module is added to the first network, and it improves the binary segmentation result by supervising the spatial gradient map of prediction. In the second network, building subclasses (HZ, LZ, SH, and SL) are predicted. Intermediate features of the second network are supervised using contrastive learning loss to improve feature consistency. Finally, predictions of the two networks are combined to generate the final result. Experimental results demonstrate that significant improvements can be obtained using our proposed framework. Adequate experiments are performed on the Hainan and xBD datasets to prove our method's effectiveness.

For future works, it would be interesting to divide the building into more fine-grained subclasses. And another possible direction of subclass segmentation is extending other classes, such as vegetation and road. These classes are more challenging than the building. For vegetation, the concept of object ceases to exist, and there is less difference in features between subclasses. The scale of roads is more flexible, and its subclasses may need to be determined by features that are far apart on the same road.

References

[1] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019.

[2] J. Yuan, "Learning building extraction in aerial scenes with convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2793–2798, Nov. 2017.

[3] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[4] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1480–1484.

[5] T.-H. K. Chen, C. Qiu, M. Schmitt, X. X. Zhu, C. E. Sabel, and A. V. Prishchepov, "Mapping horizontal and vertical urban densification in Denmark with landsat time-series from 1985 to 2018: A semantic segmentation solution," *Remote Sens. Environ.*, vol. 251, 2020, Art. no. 112096.

[6] C. Haaland and C. K. van Den Bosch, "Challenges and strategies for Urban green-space planning in cities undergoing densification: A review," *Urban Forestry Urban Greening*, vol. 14, no. 4, pp. 760–771, 2015.

[7] A. A. Sheeba and R. Jayaparvathy, "Performance modeling of an intelligent emergency evacuation system in buildings on accidental fire occurrence," *Saf. Sci.*, vol. 112, pp. 196–205, 2019.

[8] F. Peng, J. Gong, L. Wang, H. Wu, and P. Liu, "A new stereo pair disparity index (SPDI) for detecting built-up areas from high-resolution stereo imagery," *Remote Sens.*, vol. 9, no. 6, p. 633, 2017.

[9] Y. Shao, G. N. Taff, and S. J. Walsh, "Shadow detection and building-height estimation using IKONOS data," *Int. J. Remote Sens.*, vol. 32, no. 22, pp. 6929–6944, 2011.

[10] R. Gupta et al., "Creating xbd: A dataset for assessing building damage from satellite imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 10–17.

[11] F. Nex, D. Duarte, F. G. Tonolo, and N. Kerle, "Structural building damage detection with deep learning: Assessment of a state-of-the-art CNN in operational conditions," *Remote Sens.*, vol. 11, no. 23, 2019, Art. no. 2765.

[12] M. Presa-Reyes and S.-C. Chen, "Assessing building damage by learning the deep feature correspondence of before and after aerial images," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, 2020, pp. 43–48.

[13] B. Sirmacek and C. Unsalan, "Building detection from aerial images using invariant color features and shadow information," in *Proc. 23rd Int. Symp. Comput. Inf. Sci.*, 2008, pp. 1–5.

[14] Y. Zhang, "Optimisation of building detection in satellite images by combining multispectral classification and texture filtering," *ISPRS J. Photogrammetry Remote Sens.*, vol. 54, no. 1, pp. 50–60, 1999.

[15] Y. Li and H. Wu, "Adaptive building edge detection by combining Lidar data and aerial images," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 37, no. Part B1, pp. 197–202, 2008.

[16] G. Ferraioli, "Multichannel InSar building Edge detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1224–1231, Mar. 2009.

[17] S.-H. Zhong, J.-J. Huang, and W.-X. Xie, "A new method of building detection from a single aerial photograph," in *Proc. 9th Int. Conf. Signal Process.*, 2008, pp. 1219–1222.

[18] S. Taoufiq, B. Nagy, and C. Benedek, "Hierarchynet: Hierarchical CNN-based urban building classification," *Remote Sens.*, vol. 12, no. 22, 2020, Art. no. 3794. [Online]. Available: https://www.mdpi.com/2072-4292/12/22/3794

[19] Y. Huang, L. Zhuo, H. Tao, Q. Shi, and K. Liu, "A novel building type classification scheme based on integrated Lidar and high-resolution images," *Remote Sens.*, vol. 9, no. 7, p. 679, 2017.

[20] M. Zhen et al., "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13663–13672.

[21] T. Cheng, X. Wang, L. Huang, and W. Liu, "Boundary-preserving mask R-CNN," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 660–676.

[22] H. Jung, H.-S. Choi, and M. Kang, "Boundary enhancement semantic segmentation for building extraction from remote sensed image," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5215512.

[23] S. Wei, S. Ji, and M. Lu, "Toward automatic building footprint delineation from aerial images using CNN and regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2178–2189, Mar. 2020.

[24] S. He and W. Jiang, "Boundary-assisted learning for building extraction from optical remote sensing imagery," *Remote Sens.*, vol. 13, no. 4, p. 760, 2021.

[25] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5891–5900.

[26] S. Xie and Z. Tu, "Holistically-nested Edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[27] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "CaseNet: Deep category-aware semantic Edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5964–5973.

[28] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2021.3090866.

[29] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[30] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," 2014, *arXiv:1410.8516*.

[31] R. Lopez, J. Regier, M. I. Jordon, and N. Yosef, "Information constraints on auto-encoding variational bayes," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[32] A. V. D. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 4797–4805.

[33] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statis., JMLR Workshop Conf. Proc.*, 2010, pp. 297–304.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.

[35] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1735–1742.

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[37] T.-W. Ke, J.-J. Hwang, and S. X. Yu, "Universal weakly supervised segmentation by pixel-to-segment contrastive learning," in *Proc. Int. Conf. Learn. Representations*, 2021.
[38] P. Khosla et al., "Supervised contrastive learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 18 661–18 673, 2020.
[39] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7303–7313.
[40] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16679–16688.
[41] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, "Contrastive learning of global and local features for medical image segmentation with limited annotations," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12 546–12 558, 2020.
[42] M. Zhu, J. Fan, Q. Yang, and T. Chen, "SC-EADNet: A self-supervised contrastive efficient asymmetric dilated network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5519517.
[43] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.
[44] Y. Qin, L. Bruzzone, and B. Li, "Learning discriminative embedding for hyperspectral image clustering based on set-to-set and sample-to-sample distances," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 473–485, Jan. 2020.
[45] L. Zhang, S. Zhang, B. Zou, and H. Dong, "Unsupervised deep representation learning and few-shot classification of PolSAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5100316.
[46] Y. Zhai et al., "Weakly contrastive learning via batch instance discrimination and feature clustering for small sample SAR ATR," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5204317.
[47] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.
[48] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.
[49] 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.*, 2018, pp. 833–851.
[50] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 272–2723.
[51] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 173–190.
[52] R. Li et al., "Multiattention network for semantic segmentation of fine-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5607713.
[53] D. Wang and Q. Dong, "MCFINet: Multidepth convolution network with shallow-deep feature integration for semantic labeling in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8009305.
[54] R. Gupta and M. Shah, "RescueNet: Joint building segmentation and damage assessment from satellite imagery," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 4405–4411.
[55] E. Weber and H. Kané, "Building disaster damage assessment in satellite imagery with multi-temporal fusion," 2020, *arXiv:2004.05525*.
[56] Y. Shen et al., "BdaNet: Multiscale convolutional neural network with cross-directional attention for building damage assessment from satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5402114.
[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.