


Article

Transformer-Based Semantic Segmentation for Extraction of Building Footprints from Very-High-Resolution Images

Jia Song^{1,3,*} , A-Xing Zhu^{1,2} and Yunqiang Zhu¹

¹ State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China

² Department of Geography, University of Wisconsin, Madison, WI 53706, USA

³ Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

* Correspondence: songj@igsrr.ac.cn

Abstract: Semantic segmentation with deep learning networks has become an important approach to the extraction of objects from very high-resolution remote sensing images. Vision Transformer networks have shown significant improvements in performance compared to traditional convolutional neural networks (CNNs) in semantic segmentation. Vision Transformer networks have different architectures to CNNs. Image patches, linear embedding, and multi-head self-attention (MHSA) are several of the main hyperparameters. How we should configure them for the extraction of objects in VHR images and how they affect the accuracy of networks are topics that have not been sufficiently investigated. This article explores the role of vision Transformer networks in the extraction of building footprints from very-high-resolution (VHR) images. Transformer-based models with different hyperparameter values were designed and compared, and their impact on accuracy was analyzed. The results show that smaller image patches and higher-dimension embeddings result in better accuracy. In addition, the Transformer-based network is shown to be scalable and can be trained with general-scale graphics processing units (GPUs) with comparable model sizes and training times to convolutional neural networks while achieving higher accuracy. The study provides valuable insights into the potential of vision Transformer networks in object extraction using VHR images.



Citation: Song, J.; Zhu, A.-X.; Zhu, Y. Transformer-Based Semantic Segmentation for Extraction of Building Footprints from Very-High-Resolution Images. *Sensors* **2023**, *23*, 5166. <https://doi.org/10.3390/s23115166>

Academic Editor: Gwanggil Jeon

Received: 5 April 2023

Revised: 12 May 2023

Accepted: 24 May 2023

Published: 29 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: vision transformer; hyperparameter; building; self-attention; deep learning

1. Introduction

Semantic segmentation is one of the key image classification tasks in the computer vision (CV) field. It is the process of classifying each pixel in an image belonging to a certain class and can be thought of as a classification problem per pixel [1,2]. In recent years, the success of semantic segmentation using deep convolutional neural networks (CNNs) has rapidly attracted research interest in the remote sensing community, and Object-based Image Analysis (OBIA) [3,4] has been transforming traditional image segmentation methods into semantic segmentation methods using CNNs [5–10]. CNN-based semantic segmentation is an efficient end-to-end learning approach to image classification at the pixel level [11–13]. With a large amount of training data, a CNN is able to automatically extract features from very-high-resolution (VHR) images obtained using aerial or satellite sensors and then apply them to extract natural or artificial objects [14–17] in VHR images. CNNs have been shown to perform better than swallow machine learning methods [18–20] and have become a dominant method in the extraction of objects from VHR images.

With the development of deep learning, a novel neural network architecture, Transformer [21], has garnered significant attention in the Natural Language Processing (NLP) field since 2017 [22–24], and efforts to develop Transformer networks for CV tasks have been promoted in recent years. Vision Transformer (ViT) [25], a vision model based as

closely as possible on the Transformer architecture originally designed for text-based tasks, was proposed at the end of 2020. The notable highlight of Transformer is that it is the first model that relies entirely on a self-attention mechanism to capture the salient parts of input information, and this attention mechanism is one of the most valuable breakthroughs in deep learning in recent years [26,27]. The attention mechanism refers to the ability to dynamically highlight and use the salient parts of information [28], which is similar to the ability of the human brain to dynamically and instinctively select crucial information for decision-making. ViT attains excellent results compared to state-of-the-art CNNs, while it requires substantially fewer computational resources to train [29–31]. Additionally, in comparison with attention-enhanced CNN models, a pure Transformer applied directly to sequences of image patches without a CNN can perform very well in image classification tasks [18,32]; thus, this has been the inspiration for a new wave of vision Transformer networks [33,34], including Pyramid ViT [35], SegFormer [36], Swin Transformer [37], and so on.

Vision Transformer networks show great development potential in the field of computer vision. Nevertheless, investigations into Transformer-based networks for the extraction of geographical objects from VHR remote-sensing images remain scarce [38–41]. Vision Transformer networks have unique implementations such as image patches [42], linear embedding [43], and multi-head self-attention (MHSA) [44]. It remains unclear how we can more effectively configure them for the extraction of objects in VHR images and how they affect the accuracy of networks [45]. Therefore, this article leverages vision Transformer networks for object extraction from VHR remote-sensing images. As building footprints are essential artificial objects on the land surface and there are already several publicly available training datasets of buildings in the remote sensing classification community, we chose building footprints as the research object to investigate the impact on the accuracy of hyperparameters that are often specific to vision Transformer networks [46–49]. In the next section, existing studies on building footprint extraction methods are reviewed, and the foundations of vision Transformer networks are introduced to elucidate how vision Transformers work. In Section 3, a network based on Swin Transformer is presented for the extraction of building footprints. Based on this network, we set up eight different models, and each model had different Transformer-specific hyperparameter values. These models are presented in Section 4, as well as a comparison of the performance of each of them. Section 5 presents the experiment results, followed by a discussion of the results. Section 6 concludes this paper.

2. Related Work

2.1. Building Footprint Extraction Methods

Traditional building footprint extraction methods mainly rely on features designed manually by humans, such as the texture and geometric features of buildings, and the algorithms of building footprint extraction include the gray level co-occurrence matrix [50], Gabor wavelet transform [51], corner detection [52], and contour grouping [53]. However, due to the limited number of features and the model size, the deeper or more abstract features of building footprints are difficult to represent; thus, traditional building extraction methods usually have lower levels of extraction accuracy compared to deep learning methods.

With the advent of deep learning techniques, semantic segmentation methods based on convolutional neural networks (CNNs) have provided new approaches for the extraction of buildings from VHR images. These networks are mainly based on Fully Convolutional Networks (FCNs) [54], SegNet [55], U-Net [56], and DeepLab. For example, CNNs based on ResNet or DenseNet backbone networks combined with Conditional Random Fields (CRFs) [57], the U-Net++ network reconstructed with DenseNet as a backbone network [58], and the SegNet network improved with the Gaussian algorithm and Image Pyramid [59] are all CNN-based building footprint extraction methods. CNN-based methods have dominated the field of building footprint extraction for several years due to their ability to learn and extract complex features from VHR images.

In the last two years, with the great success of Transformer methods in the computer field, Transformer-based semantic segmentation methods have also been utilized for the extraction of building footprints [60–63], such as BuildFormer [64], a ViT-based model with a dual-path structure capable of capturing global context with large windows; MSST-Net [46], a multi-scale adaptive segmentation network model based on Swin Transformer; STT (Sparse Token Transformer) [29], an efficient dual-pathway Transformer structure that learns long-term dependencies in both spatial and channel dimensions; and STEB-UNet [65], a network integrating a Swin-Transformer-based encoding booster in a specially designed U-shaped network to achieve the feature-level fusion of local and large-scale semantics. These novel Transformer-based approaches show great promise for further improvements to the accuracy of building footprint extraction. However, it is important to note that the different hyperparameters of Transformers can also affect the model performance and should be considered. Therefore, this study pays more attention to the impact of the hyperparameters of the Swin Transformer, providing valuable insights into the more effective utilization of vision Transformer networks in VHR images.

2.2. Foundations of Transformers in Vision

Transformers in vision are based on the architecture of the Transformer originally designed for text-based NLP tasks. Instead of a series of word embeddings as the inputs of the Transformer in NLP, image patches, which are generated via image partition, are the inputs of Transformers in vision, and the attention is computed on top of the image patches. Transformers in vision consist of a stack of Transformer blocks, and the Transformer block includes Layer Normalization (LN), multi-head attention (MHA), and Multi-layer Perceptron (MLP), as shown in Figure 1. Residual connections are applied on both MHSA and MLP to resolve the difficulty in the convergence of multi-layer neural networks.

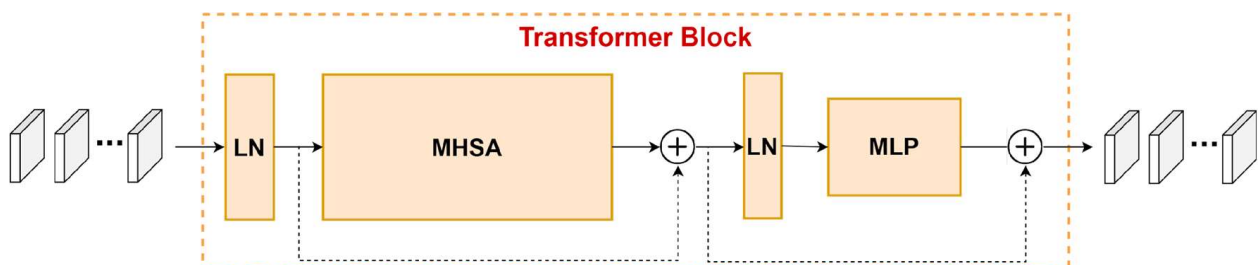


Figure 1. Framework of the Transformer block.

2.3. Layer Normalization (LN)

LN is used before every block and residual connections after every block in a Transformer to scale the features for each sample of a sequence. LN helps to speed up and stabilize the learning process. Additionally, LN [66] is proven to yield significantly better performance than Batch Normalization (BN) in Transformers, and BN is often used in CNNs to scale an entire feature map. For a batch of sentences in Transformers in NLP, BN scales over the words at the same position of each sentence, and LN scales over all the words in each sentence, as shown in Figure 2. Obviously, scaling the words at the same position in different sentences does not follow the design of sequence models, whereas LN satisfies the requirements of Transformers.

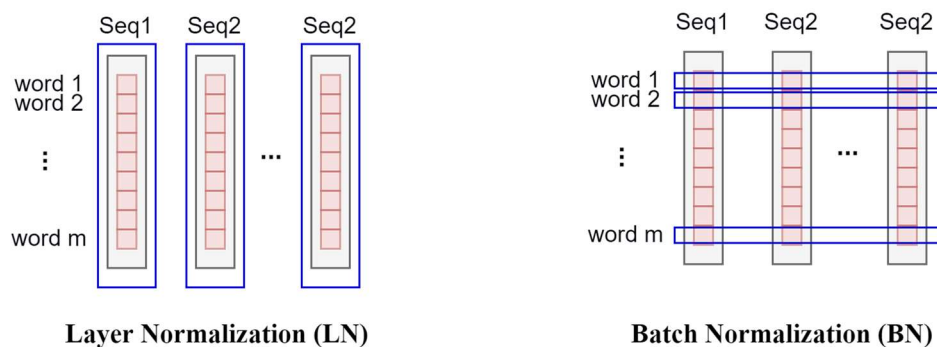


Figure 2. Layer Normalization (LN) and Batch Normalization (BN).

2.4. Multi-Head Self Attention (MSA)

MSA in Transformers is multiple self-attentions in parallel, and each head of self-attention is concatenated and then projected to outputs, as shown in Figure 3. Most Transformers use standard self-attention [21], which is based on scaled-dot products to compute self-attention. Three inputs of Queries (Q), Keys (K), and Values (V) are used to generate self-attention feature maps. Q and K are used to generate weights of features, and the weights work on V , generating self-attention feature maps. The Q , K , and V of standard self-attention are outputs of linear operations with the learnable parameters W^Q , W^K , and W^V , and the standard self-attention is computed as:

$$Self\ Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimension of both Q and K , and the Softmax function scales the weights in the range $[0, 1]$ and makes the weights equal to one. The multi-head self-attention links multiple convolution kernels in CNNs to generate multiple feature maps. The more self-attention feature maps there are, the better the performance models could achieve. The multi-head self-attention is computed as:

$$MSA(Q, K, V) = Concat(head_1, \dots, head_h)W^O \tag{2}$$

where $head_i = Self\ Attention(QW_i^Q, KW_i^K, VW_i^V)$.

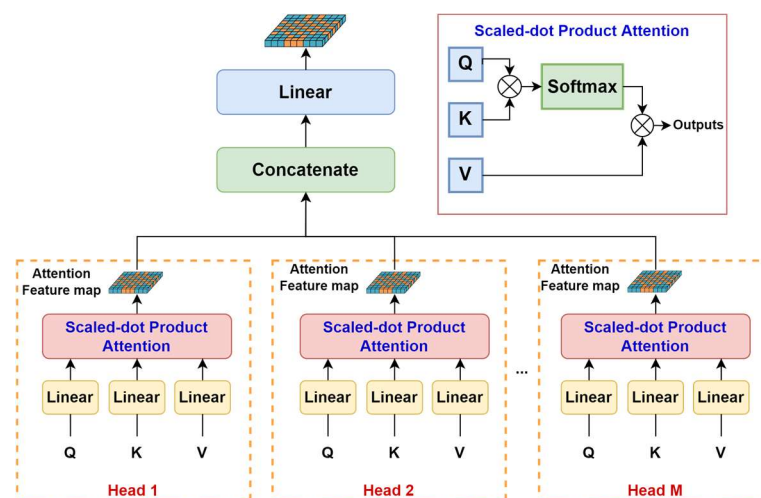


Figure 3. Standard multi-head self-attention in Transformer.

2.5. Multi-Layer Perceptron (MLP)

MLP, also known as the Feed-Forward network (FFN), consists of two linear layers and a GELU nonlinearity in Transformers. The outputs from MLP are added to the inputs (skip connection) to obtain the final output of the Transformer block. The role and purpose of MLP are to process the output from one attention layer in a way that fits the input for the next attention layer better.

3. Transformer-Based Network for Extraction of Build Footprints from VHR Images

3.1. Network Architecture

The proposed Transformer-based network for building extraction has an encoder-decoder architecture, as shown in Figure 4. A novel Swin Transformer is utilized as the encoder to extract the multi-scale-self-attention-based features of the VHR images. Based on the multi-scale features, we further introduce a Pyramid Pooling Module (PPM) [67] in the decoder to add global context to a VHR image; then, we use a Feature Pyramid Network (FPN) [68] in the decoder to fuse the multiple different scales of feature maps. All of these fused feature maps are upsampled into the original resolution of the VHR image via a segmentation head. The segmentation head projects the feature maps onto the pixel space to obtain pixel-by-pixel coverage of the building footprints.

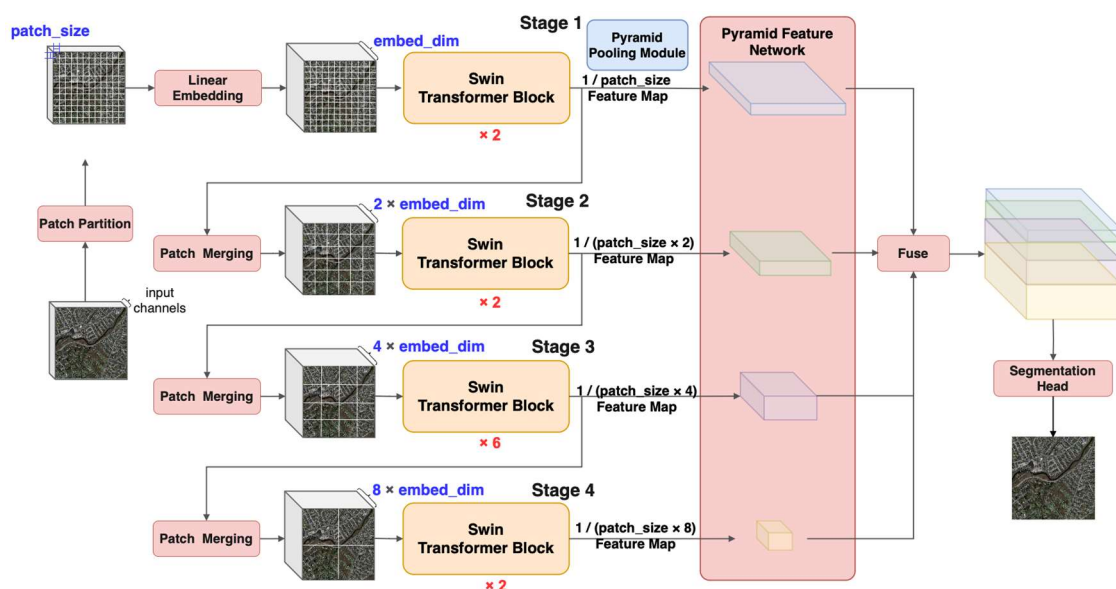


Figure 4. The structure of the Transformer-based network for extraction of building footprints from VHR images.

3.2. Network Modules

The proposed network is composed of a Patch Partition module, Linear Embedding module, Patch Merging module, Swin Transformer block module, Pyramid Pooling Module, and Feature Pyramid Fusion Module. They are described as follows:

3.2.1. Patch Partition

The Patch Partition module is the first layer of the Transformer-based encoder. The Patch Partition Layer splits the raw VHR image into non-overlapping patches for the application of self-attention to the image patches rather than the pixels. The self-attention to image patches can reduce the time complexity of training and thus make the Transformer-based network applicable to a large number of VHR images.

3.2.2. Linear Embedding

“Embedding” means taking some sets of raw inputs and converting them to vectors in machine learning. The Linear Embedding module in Vision Transformers thus takes a sequence of image patches as the input and generates a vector representation of the image patches in another mathematical space using a linear transformation. It can be seen as the abstract representation of the original information at the semantic level. Additionally, with the Linear Embedding module, the arbitrary channel number and arbitrary size of image patches can be transformed into a sequence of one-dimension vectors with the same length, thereby enhancing the model’s ability to adapt to different kinds of images as inputs.

3.2.3. Swin Transformer Block

Swin Transformer blocks [37] are kernels in the Transformer-based building extraction network which implement the self-attention mechanism in an efficient way. Swin Transformer blocks are often stacked to capture deeper and more advanced features, as CNN blocks do. Inside a Swin Transformer block, a shifted window is introduced to compute both local and global self-attention. The shifted windows are non-overlapping windows that partition the VHR images on the top of image patches. To reduce quadratic complexity in computing self-attention, two successive Swin Transformer blocks can achieve self-attention computation with less complexity, as shown in Figure 5. The first Swin Transformer block contains a window-based multi-head attention (W-MSA) module which computes the self-attention within the window, and the second Swin Transformer block contains a shifted-window-based multi-head attention (SW-MSA) module, which computes self-attention across the windows by alternating between two partitioning configurations W in consecutive Swin Transformer blocks. Therefore, two successive Swin Transformer blocks can compute the self-attention computation over the whole VHR image, and the computation takes less time.

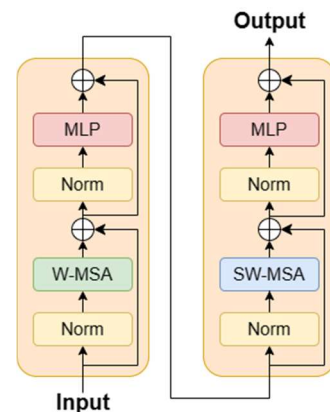


Figure 5. Swin Transformer block.

3.2.4. Pyramid Pooling

To make the model learn not only the detailed features but also the global features of VHR images, we introduce the Pyramid Pooling module in the decoder to capture the global context of the feature map learned by the encoder. The Pyramid Pooling module is an effective global prior representation and captures the global context using a CNN-based multi-level pyramid. Each level of the multi-level pyramid is a pooling layer with a different pooling rate. A multi-level pyramid of pooling layers can learn different granularities of global features, which enables the model to more comprehensively grasp information regarding the global scene of VHR images.

3.2.5. Feature Pyramid Fusion

To more effectively utilize the multi-scale feature maps generated by the encoder, the Feature Pyramid Fusion (FPN) module is applied in the decoder to fuse the feature maps

from the Pyramid Pooling and the Swin Transformer block. With the FPN, feature maps with different sizes and channel numbers are fused to a single feature map. The fused feature map integrates all the features at different levels and thus may help to further improve the classification accuracy.

3.3. Transformer-Specific Hyperparameters

The main Transformer-specific hyperparameters in the network are patch size, embedding dimension, and window size. They are described as follows:

(1) Patch size

The patch size refers to the size of image patches and determines how many pixels are in a unit to generate feature maps based on the self-attention calculation method. The patch size is related to the resolution of the feature maps. When a VHR image is represented as

$$Image(X) \in R^{H \times W \times C} \quad (3)$$

where H , W is the height and width of the VHR image, and C is the channel of the VHR image, the VHR image patches can be represented as

$$ImagePatches(X) \in R^{(P \times P \times C) \times N} \quad (4)$$

where P is the patch size, and N is the length of the sequence of image patches ($N = H \times W / P^2$). Each patch is flattened to a vector with a length of $P \times P \times C$ before it is passed into the Linear Embedding module.

(2) Dimension of embeddings

The dimension of embeddings refers to the length of a vector that represents an embedded image patch. The embedded image patches are generated by the Linear Embedding module, represented as

$$PatchEmbeddings(X) \in R^{D \times N} \quad (5)$$

where D is the dimension of embeddings, and N is the length of the sequence of embeddings, which is the same as the length of image patches.

(3) Window size

The window size refers to how many image patches are grouped to directly calculate self-attention within windows; thus, a larger window size means more image patches are used to directly calculate the window-level self-attention. Supposing each window contains $M \times M$ patches, the feature map generated by the Swin Transformer block in Stage 1 is represented as

$$FeatureMap_{stage1}(X) \in R^{H/P \times W/P \times D} \quad (6)$$

After merging image patches, the feature maps generated by the Swin Transformer block in Stage 2, Stage 3, and Stage 4 are represented as

$$FeatureMap_{stage2}(X) \in R^{H/2P \times W/2P \times 2D} \quad (7)$$

$$FeatureMap_{stage3}(X) \in R^{H/4P \times W/4P \times 4D} \quad (8)$$

$$FeatureMap_{stage4}(X) \in R^{H/8P \times W/8P \times 8D} \quad (9)$$

4. Experimental Section

4.1. Datasets

We chose the publicly available Massachusetts Buildings Dataset (<https://www.cs.toronto.edu/~vmnih/data/> accessed on 1 September 2022) as the experiment data. The Massachusetts Buildings Dataset consists of 151 aerial images in the Boston area of the U.S. Each image is 1500 pixels \times 1500 pixels with red, green, and blue bands, and the spatial resolution is 1 m. The original 151 images were split into a training dataset of 137 images, a validation dataset of 4 images, and a test dataset of 10 images.

Due to the limitation of GPU memory, the original 1500 pixel \times 1500 pixel images needed to be divided into smaller images in the experiment. Due to the hierarchical structure of the Swin Transformer, the arbitrary sizing of image samples is not recommended. The downscaling was performed during the generation of the multi-scale attention-based feature maps, and the upscaling was performed when merging them. Hence, inappropriate image sizes will lead to merging failure in the Swin Transformer. By analyzing the structure of this network, the appropriate image size was determined to be $patch_size \times 2^{merge_times} \times window_size$ or the integer multiples of it, and Table 1 lists the appropriate image sizes between approximately 200 and 400 pixels. The patch number in the Nth stage was calculated using $H/P/N \times W/P/N$, where H and W are the height and width of images, P is the size of the image patches, and N is the sequential number of the stages (i.e., 1, 2, 3, 4). Considering maximization by utilizing the original 1500 pixels \times 1500 pixels images, the image size selected in this experiment was 288 \times 288 pixels. Therefore, we finally obtained 3000 pieces of 288 \times 288-pixel samples for training and 98 pieces of the same-sized samples for validation.

Table 1. Appropriate image sizes between approximately 200 and 400 pixels.

| Input Image Size (Pixels) | Utilization for the Original 1500 \times 1500 Pixel Image | Patch Size (Pixels) | Patch Numbers of the Four Stages | Window Size (Patches) |
|---------------------------|---|---------------------|---|-----------------------|
| 224 | 89.6% | 2 | [112 ² , 56 ² , 28 ² , 14 ²] | 7 or 14 |
| | | 4 | [56 ² , 28 ² , 14 ² , 7 ²] | 7 |
| 256 | 85.3% | 2 | [128 ² , 64 ² , 32 ² , 16 ²] | 8 or 16 |
| | | 4 | [64 ² , 32 ² , 16 ² , 8 ²] | 8 |
| 288 | 96.0% | 2 | [144 ² , 72 ² , 36 ² , 18 ²] | 9 or 18 |
| | | 4 | [72 ² , 36 ² , 18 ² , 9 ²] | 9 |
| 320 | 93.9% | 2 | [160 ² , 80 ² , 40 ² , 20 ²] | 10 or 20 |
| | | 4 | [80 ² , 40 ² , 20 ² , 10 ²] | 10 |
| 352 | 93.9% | 2 | [176 ² , 88 ² , 44 ² , 22 ²] | 11 or 22 |
| | | 4 | [88 ² , 44 ² , 22 ² , 11 ²] | 11 |
| 384 | 76.8% | 2 | [192 ² , 96 ² , 48 ² , 24 ²] | 12 or 24 |
| | | 4 | [96 ² , 48 ² , 24 ² , 12 ²] | 12 |

Note: the largest utilization is highlighted in bold.

4.2. Hyperparameter Settings

In order to explore how the Transformer-specific hyperparameters affect the accuracy of the extraction of building footprints, we set up eight experiment groups, and each group had different Transformer-specific hyperparameter values, as shown in Table 2. They all were trained by the same training samples described in Section 4.1.

Table 2. Experiment group for the Transformer-specific hyperparameter.

| Experiment Group | Patch Size (Pixels) | Embedding Dimension | Window Size (Patches) |
|-------------------|---------------------|---------------------|-----------------------|
| patch2_em24_win09 | 2 | 24 | 9 |
| patch2_em96_win09 | 2 | 96 | 9 |
| patch2_em24_win18 | 2 | 24 | 18 |
| patch2_em96_win18 | 2 | 96 | 18 |
| patch4_em24_win09 | 4 | 24 | 9 |
| patch4_em96_win09 | 4 | 96 | 9 |
| patch4_em24_win18 | 4 | 24 | 18 |
| patch4_em96_win18 | 4 | 96 | 18 |

4.3. Training Settings

The eight building footprint extraction networks with the different hyperparameter values were trained on the same NVIDIA GeForce RTX 3080 Ti GPU with 12 GB memory for 200 epochs. The batch size of the training samples was set to four due to the capacity limitation of GPU memory. The optimizer employed in the experiment was AdamW, with an initial learning rate of 6×10^{-5} and a weight decay of 0.01. In addition, a scheduler of linear learning rate was used to train the models with a warmup of 10 iterations. The building footprint extraction networks were not pre-trained on any other datasets, and no data augmentation methods were applied.

4.4. Evaluation Metrics

Four evaluation metrics were used in this study to evaluate the inference results. They are listed as follows:

(1) Overall accuracy (OA)

Accuracy is the metric calculated in the simplest way. It is the ratio of the correct predictions to the total number of predictions, represented as

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

where TP , FP , TN , and FN are the number of true positives, false positives, true negatives, and false negatives, respectively, in the confusion matrix;

(2) Intersection over union (IoU)

The mIoU is the average IoU. The IoU, also known as the intersection over union, is often used in object detection and semantic segmentation. It is the ratio of the overlap and union areas of prediction and ground truth. The mIoU can also be represented as

$$mIoU = \frac{1}{n+1} \sum_{i=1}^n \frac{TP}{TP + FP + FN} \quad (11)$$

(3) F1-score

The F1-score is a metric that combines the precision and recall metrics, and it is more suitable for imbalanced data. The F1 score is defined as the harmonic mean of precision and recall, represented as

$$F1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

(4) Kappa

Kappa, also known as Cohen's Kappa [69], is a metric used to assess the agreement between two raters. Kappa is also a useful evaluation metric when dealing with imbalanced data. It is represented as

$$Kappa = \frac{p_0 - p_e}{1 - p_e} \quad (13)$$

where p_0 is the overall accuracy of a model and p_e is the measure of the agreement between the model predictions and the ground truth values.

5. Results and Discussion

5.1. Accuracy Evaluation

Accuracy evaluation was performed when training the models. After every epoch, the evaluation was performed using the validation samples described in Section 4.1.

Figure 6 shows the accuracy variation curve on the validation samples during training. It demonstrates that the networks with the 2×2 pixels image patches and 96-dimensional embeddings (i.e., 'patch2_em96_win09' and 'patch2_em96_win18') achieved the highest score for all metrics. Figure 5 also demonstrates that when the patch size and the dimension of the embeddings of the build footprint extraction networks were the same, the varying curves of OA, mIoU, F1-score, and Kappa were very similar, which indicates that the window size of the network has little impact on the accuracy of the building footprint extraction.

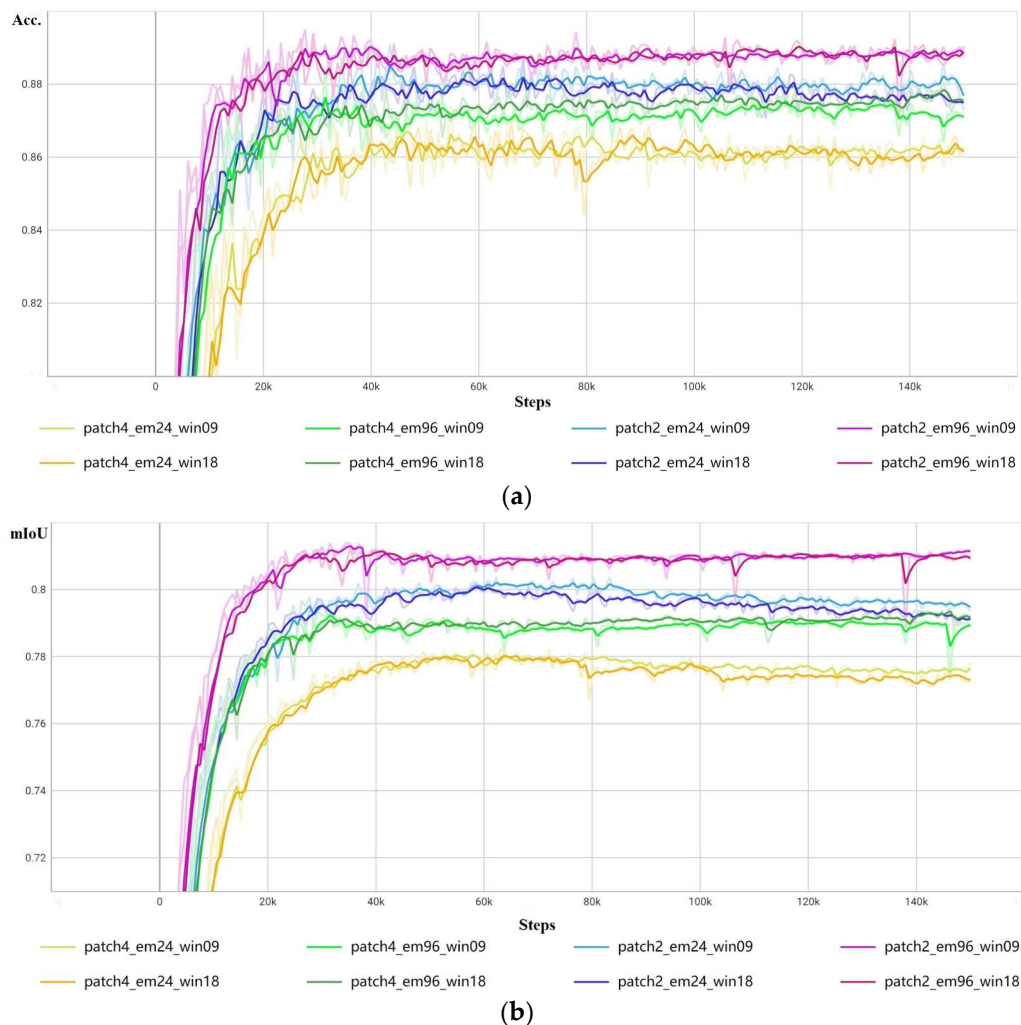


Figure 6. Cont.

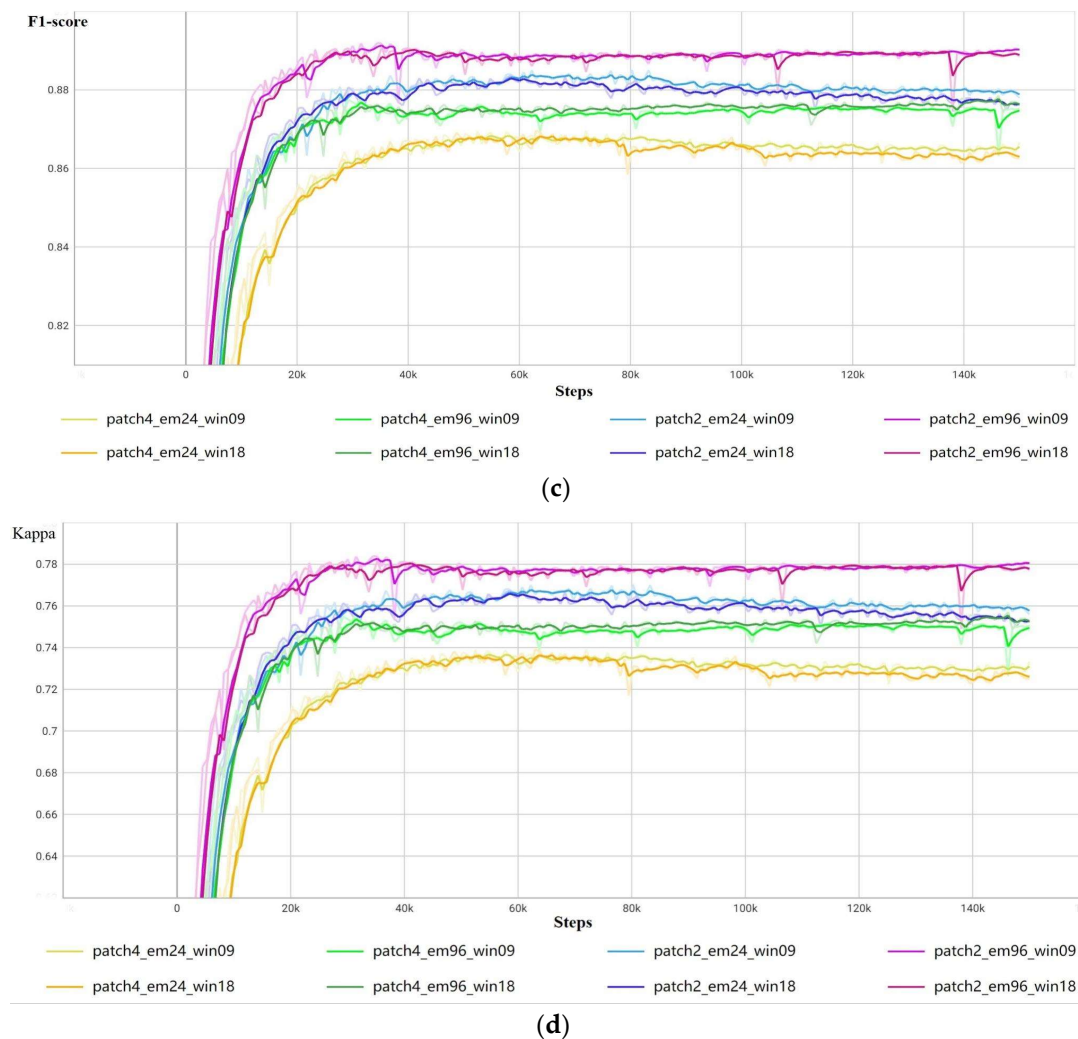


Figure 6. The accuracy evaluation curves on the validation samples. (a) OA change curve on the validation dataset. (b) mIoU change curve on the validation dataset. (c) F1-score change curve on the validation dataset. (d) Kappa change curve on the validation dataset.

Table 3 further lists the top-three accuracy evaluation results. The ‘patch2_em96_win09’ experiment group achieved the best performance, which comprised values of 0.8913 for OA, 0.8138 for mIoU, 0.8919 for F1-score, and 0.7838 for Kappa, and the ‘patch2_em96_win18’ experiment group had very similar evaluation results to ‘patch2_em96_win09’. The ‘patch2_em96_win09’ and ‘patch2_em96_win18’ experiment groups had the same 96-dimension embeddings and two-pixel-sized image patches, and only their window sizes were different. Table 3 also demonstrates that the other experiment groups, which had the same patch sizes and embedding dimensions but different window sizes, had similar evaluation results. For example, ‘patch4_em24_win09’ and ‘patch4_em24_win18’ had the same four-pixel-sized image patches and 24-dim embeddings, and their results were similar.

Table 3 also shows that the vision Transformer networks with 96-dim embeddings had higher levels of evaluation accuracy than those with 24-dim embeddings. Higher-dimensional embeddings can represent richer features of buildings on remote sensing images. With the representation of richer features, the network can more effectively distinguish buildings from other objects, thereby obtaining a higher level of accuracy. Additionally, as the dimension of embeddings reflects the level of feature representation, we suggest higher-dimensional embeddings are suitable for extracting features of complex objects such as crops and wetlands, while lower dimensions can be relatively simple objects such as water and ice. However, it should be noted that using higher-dimensional

embeddings increases the size of the model, resulting in higher CPU and GPU memory usage. Hence, given high-dimensional embeddings, it is necessary to pay attention to the size of the model so as not to exceed GPU memory limitations.

Table 3. Evaluation metrics scores for the different models.

| Experiment Group | Top-3 | OA | mIoU | F1-Score | Kappa | Epoch |
|-------------------|------------|---------------|---------------|---------------|---------------|-------|
| patch4_em24_win09 | 1 | 0.8663 | 0.7822 | 0.8696 | 0.7393 | 96 |
| | 2 | 0.8675 | 0.7821 | 0.8695 | 0.7391 | 85 |
| | 3 | 0.8627 | 0.7820 | 0.8694 | 0.7388 | 77 |
| | Avg | 0.8655 | 0.7821 | 0.8695 | 0.7391 | - |
| patch4_em24_win18 | 1 | 0.8673 | 0.7822 | 0.8697 | 0.7393 | 84 |
| | 2 | 0.8642 | 0.7815 | 0.8691 | 0.7382 | 85 |
| | 3 | 0.8682 | 0.7814 | 0.8691 | 0.7382 | 65 |
| | Avg | 0.8666 | 0.7817 | 0.8693 | 0.7386 | - |
| patch4_em96_win09 | 1 | 0.8838 | 0.7945 | 0.8785 | 0.7569 | 42 |
| | 2 | 0.8743 | 0.7933 | 0.8775 | 0.7550 | 39 |
| | 3 | 0.8730 | 0.7925 | 0.8769 | 0.7538 | 40 |
| | Avg | 0.8770 | 0.7934 | 0.8776 | 0.7552 | - |
| patch4_em96_win18 | 1 | 0.8804 | 0.7947 | 0.8786 | 0.7571 | 195 |
| | 2 | 0.8775 | 0.7937 | 0.8778 | 0.7557 | 186 |
| | 3 | 0.8782 | 0.7936 | 0.8778 | 0.7555 | 189 |
| | Avg | 0.8787 | 0.7940 | 0.8781 | 0.7561 | - |
| patch2_em24_win09 | 1 | 0.8864 | 0.8038 | 0.8851 | 0.7703 | 107 |
| | 2 | 0.8818 | 0.8038 | 0.8850 | 0.7701 | 111 |
| | 3 | 0.8855 | 0.8037 | 0.8850 | 0.7700 | 83 |
| | Avg | 0.8846 | 0.8038 | 0.8850 | 0.7701 | - |
| patch2_em24_win18 | 1 | 0.8802 | 0.8019 | 0.8837 | 0.7674 | 78 |
| | 2 | 0.8827 | 0.8013 | 0.8834 | 0.7668 | 81 |
| | 3 | 0.8872 | 0.8011 | 0.8833 | 0.7666 | 91 |
| | Avg | 0.8834 | 0.8014 | 0.8835 | 0.7669 | - |
| patch2_em96_win09 | 1 | 0.8931 | 0.8139 | 0.8920 | 0.7839 | 46 |
| | 2 | 0.8909 | 0.8138 | 0.8919 | 0.7837 | 42 |
| | 3 | 0.8898 | 0.8138 | 0.8918 | 0.7837 | 47 |
| | Avg | 0.8913 | 0.8138 | 0.8919 | 0.7838 | - |
| patch2_em96_win18 | 1 | 0.8900 | 0.8124 | 0.8909 | 0.7819 | 47 |
| | 2 | 0.8898 | 0.8121 | 0.8907 | 0.7814 | 53 |
| | 3 | 0.8940 | 0.8119 | 0.8906 | 0.7813 | 39 |
| | Avg | 0.8913 | 0.8121 | 0.8907 | 0.7815 | - |

Note: underline denotes the highest score.

Table 3 also shows that the vision Transformer networks with two-pixel-sized image patches outperformed those with four-pixel-sized image patches, as image patches, rather than pixels, are used to calculate self-attention in vision Transformer networks. Smaller image patches generate higher-resolution features that are fed into the model and calculated to output attention feature maps. A finer attention feature map obviously reduces the number of errors raised by upsampling to the original size of images. As a result, using

smaller image patches improves the accuracy of building footprint extraction. Also, unlike the common use of four- or six-pixel-sized patches in natural images, our findings indicate that two-pixel-sized patches are preferred in the context of VHR image analysis. Therefore, we recommend using two-pixel-sized patches for building footprint extraction tasks to maximize accuracy and reduce errors related to upsampling.

In addition, we compared Transformer-based methods to the CNN-based methods in the extraction of building footprints, as shown in Table 4. U-Net and DeepLab V3 are the most commonly used networks in the extraction of building footprints; thus, they were selected for comparison. Table 4 shows that the Transformer-based network outperformed the CNN-based U-Net and DeepLab V3 networks in all of the evaluated metrics. This result is consistent with the result in the CV field.

Table 4. Quantitative comparison with CNN-based methods.

| Methods | Parameters (Million) | OA | mIoU | F1-Score | Kappa |
|--|----------------------|--------|--------|----------|--------|
| U-Net | 7.7 | 0.8271 | 0.7412 | 0.8390 | 0.6780 |
| DeepLabV3 | 39.6 | 0.8339 | 0.7471 | 0.8440 | 0.6881 |
| Swin Transformer+PFN (patch2_em96_win09) | 62.3 | 0.8913 | 0.8138 | 0.8919 | 0.7838 |

In general, the accuracy evaluation results confirm that the size of image patches and the dimension of embeddings has significant impacts on the accuracy of the extraction of building footprints using vision Transformer networks. Smaller-sized image patches or higher-dimension embeddings can achieve a higher level of accuracy in building footprint extraction, whereas the parameter of window size has little impact on the accuracy.

5.2. Model Size and Training Time

With the same GPU, the training time is mainly determined by the size of the model and training samples. In this study, the number of training samples was 3000, and the size of each sample was 288×288 pixels. The training times for the eight experiment groups are listed in Table 5. It can be seen that, in general, the training time of our Transformer-based building footprint extraction network was approximately between 9 and 12 h. The exact training time for each experimental group was slightly different due to the different parameter settings. We can see that the higher the embedding dimension was, the longer the training took since higher-dimensional embeddings lead to larger models.

Table 5. Training time, model parameters, and sizes.

| Experiment Group | Training Time (Hours) | Model Parameters (Million) | Model Size (MB) |
|-------------------|-----------------------|----------------------------|-----------------|
| patch4_em24_win09 | 8.8 | 8.9 | 35 |
| patch2_em24_win09 | 9.0 | 8.9 | 35.6 |
| patch4_em24_win18 | 9.4 | 8.9 | 35.6 |
| patch2_em24_win18 | 9.8 | 8.9 | 35.6 |
| patch4_em96_win09 | 10.4 | 62.3 | 249.2 |
| patch2_em96_win09 | 10.8 | 62.3 | 249.2 |
| patch4_em96_win18 | 12.4 | 62.3 | 249.2 |
| patch2_em96_win18 | 12.9 | 62.3 | 249.2 |

5.3. Prediction Results

Since buildings in remote sensing images have different sizes and non-buildings could be misclassified as buildings, we show the prediction results in terms of large buildings, small buildings, and non-building misclassification.

Large buildings. In this study, buildings with areas larger than 1000 sq. meters were classified as large buildings, such as shopping malls, big libraries, and museums. Figure 7 shows the results of the large building footprints predicted using the Transformer-based building footprint extraction network. It demonstrates that the models with 96-dim embeddings (i.e., Figure 7e–h) generally outperformed the ones with 24-dim embeddings (i.e., Figure 7a–d), and the integrity of the large building footprint boundaries extracted using the 96-dim embeddings was better than that of those extracted using the 24-dim embeddings. We believe that higher-dimensional embeddings have more parameters, which helps to more accurately represent the overall characteristics of large buildings, resulting in better integrity when extracting them. Regarding the patch size, the results show its value was less sensitive than the embedding dimensions to the large buildings. This demonstrates that patch size is related to spatial resolution, and spatial resolution has a small impact on the extraction of large buildings from VHR images.

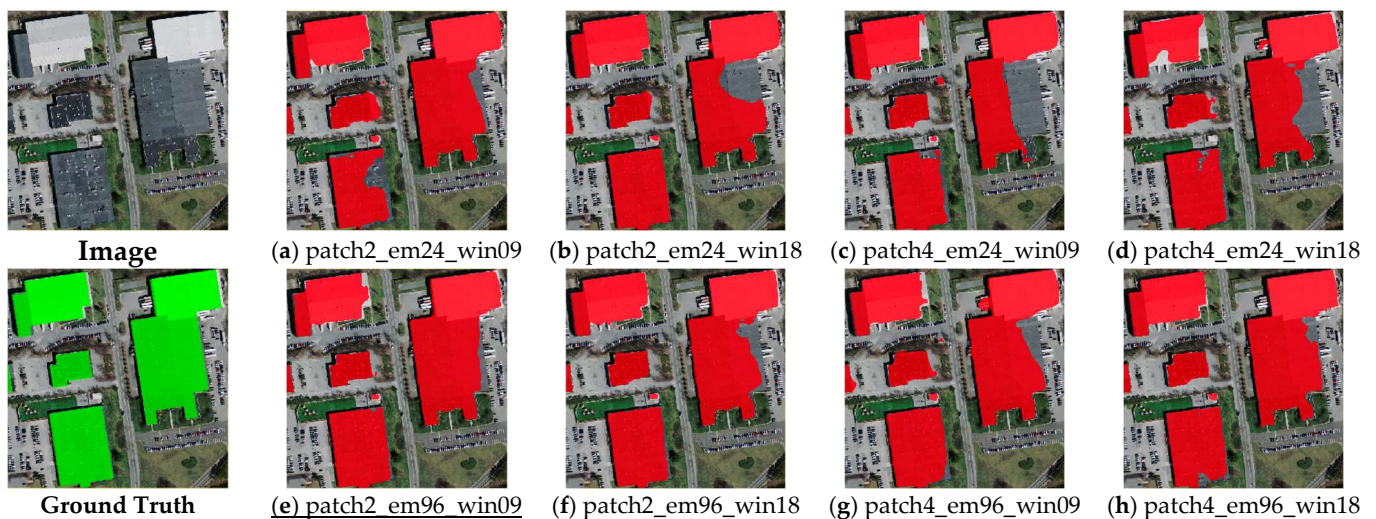


Figure 7. Example of prediction results for large buildings (underline denotes the best result among (a–h)).

Small buildings. In this study, buildings with areas smaller than 300 sq. meters were classified as small buildings, such as houses and small commercial buildings. Figure 8 shows the results of the small building footprints predicted using the Transformer-based building footprint extraction network. The results demonstrate that the models with 2×2 -pixel image patches (i.e., Figure 8a,b,e,f) generally outperformed those with 4×4 -pixel image patches (i.e., Figure 8c,d,g,h). This suggests that smaller image patches are more effective for the prediction of the footprints of small buildings. These results could be explained by the fact that using smaller image patches helps the network capture finer details and edges, which can be important for the accurate prediction of small buildings' footprints. In contrast, using larger image patches may result in the loss of some finer details, as well as the overlapping of the extracted building footprints.

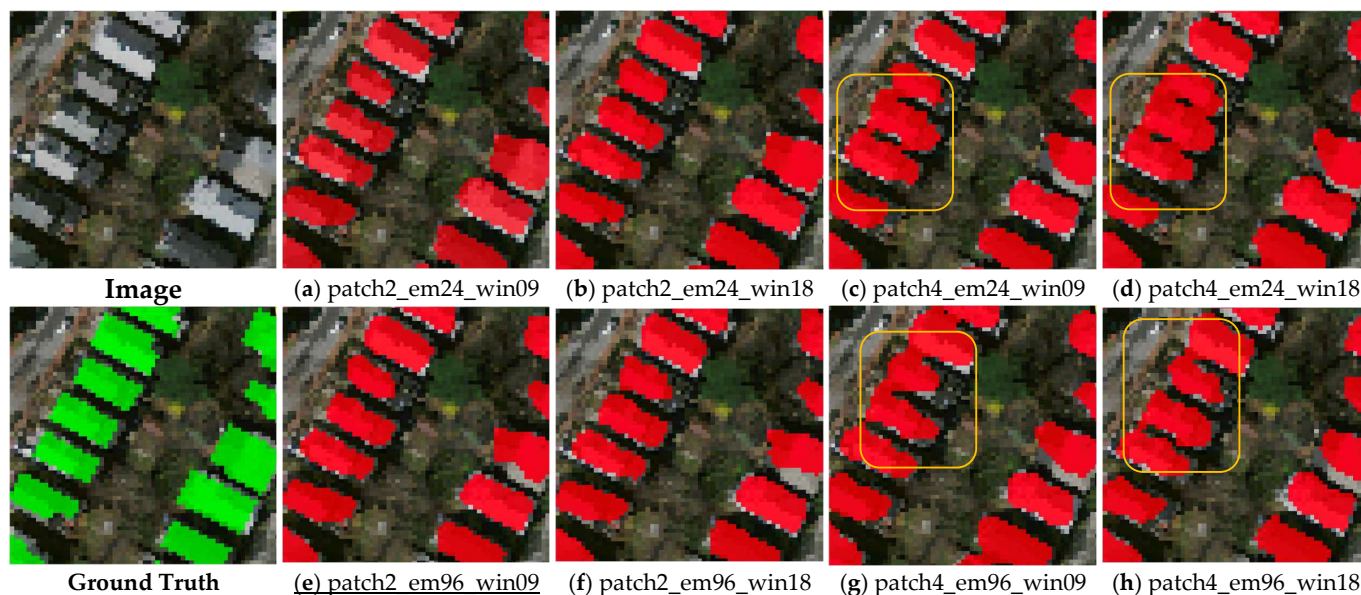


Figure 8. Example of prediction results for small buildings (underline denotes the best result among (a–h), and the overlapping area is circled in orange).

Non-building misclassification. In this study, the main non-building objects misclassified as building footprints are roads. Figure 9 shows an example of roads being misclassified as building footprints. It can be seen that the ‘patch2_em96_win09’ experiment group (i.e., Figure 9a) achieved the best performance, and a few pixels of roads were misclassified as building footprints. The ‘patch2_em96_win09’ experiment group, which was only different in terms of window size, only misclassified a few pixels of roads. Figure 6 also demonstrates the models with 24-dim embeddings (i.e., Figure 9a–d) misclassified roads more seriously than the models with 96-dim embeddings (i.e., Figure 9e–h), especially for the models with patch sizes of 4×4 pixels (i.e., Figure 9c,d).

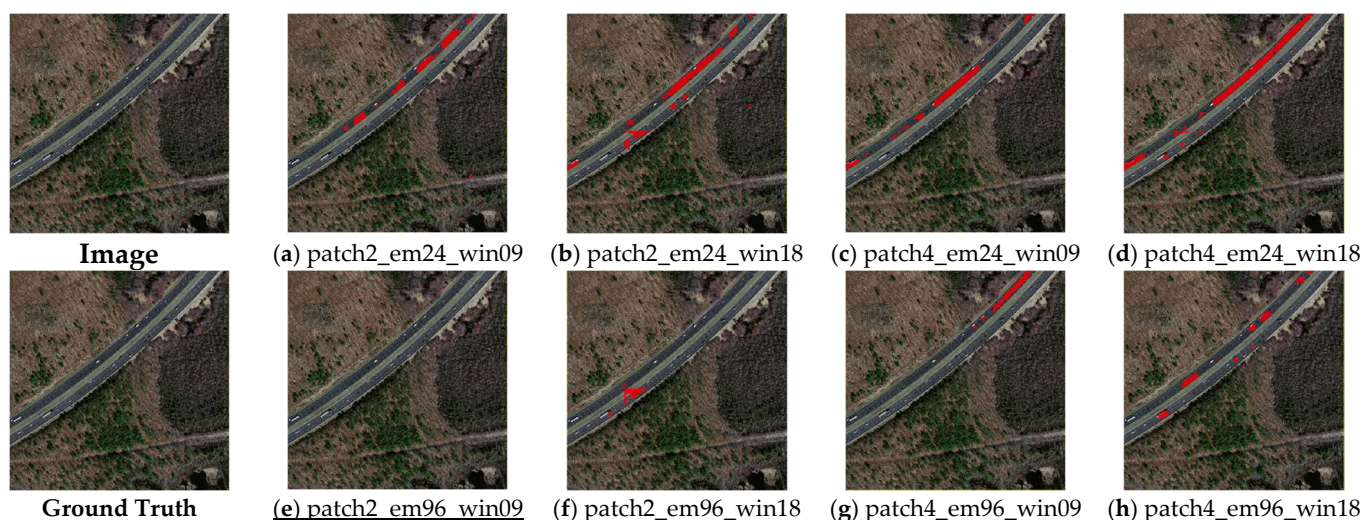


Figure 9. Example of the misclassified result of roads (underline denotes the best result among (a–h)).

6. Conclusions

Vision Transformer networks have been developed as an alternative to CNNs and have shown significant improvements in performance over traditional CNNs in multiple tasks such as image classification, object detection, and semantic segmentation. This study explored the potential of vision Transformer networks in extracting geographical

objects from VHR images, with a focus on building footprints. Moreover, we analyzed the particular hyperparameters of Swin Transformer networks, such as image patches, linear embedding, and window size, and investigated how they affect the accuracy of the extraction of building footprints. We found the hyperparameters of image patches and linear embedding had significant impacts on the accuracy. Smaller image patches resulted in higher accuracy in building footprint extraction. High-dimensional embeddings also resulted in higher accuracy in building footprint extraction. The window size had a smaller impact on the accuracy, but it impacted the size of the model, thereby affecting the training time. With the same image patches and embeddings, we recommend a smaller window size for the Swin Transformer network. These results provide an essential reference in Transformer-based network hyperparameter configuration to improve the accuracy of land cover classification with VHR images. In our experiment, when the size of the image patches was 2×2 pixels, the dimension of the embeddings was 96, and the window size was nine, the network achieved the highest accuracy in building footprint extraction. The values were 0.8913 for OA, 0.8138 for mIoU, 0.8919 for F1-score, and 0.7838 for Kappa, and the accuracy evaluation was based on the Massachusetts Buildings Dataset (<https://www.cs.toronto.edu/~vmnih/data/> accessed on 16 November 2022). In addition, the experiment also showed that the Swin Transformer network could be trained with general-scale GPUs when applying VHR remote sensing images, and the model size and training time are acceptable compared to traditional CNNs while achieving better accuracy. This further demonstrates that Transformer networks are highly scalable and have broad potential applications in the field of remote sensing.

Author Contributions: Conceptualization, J.S. and A.-X.Z.; methodology, J.S.; software, J.S.; validation, J.S.; formal analysis, J.S.; writing—original draft preparation, J.S.; writing—review and editing, A.-X.Z. and Y.Z.; visualization, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2022YFF0711602), the 14th Five-year Informatization Plan of Chinese Academy of Sciences (CAS-WX2021SF-0106), and the National Data Sharing Infrastructure of Earth System Science (<http://www.geodata.cn/> accessed on 16 November 2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original Massachusetts Buildings Dataset is available at (<https://www.cs.toronto.edu/~vmnih/data/> accessed on 1 September 2022). The data generated and analyzed during this study are available from the corresponding author by request.

Acknowledgments: We appreciate the detailed comments from the editor and the anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yuan, X.; Shi, J.; Gu, L. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Syst. Appl.* **2021**, *169*, 114417. [[CrossRef](#)]
2. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimedia Inf. Retr.* **2018**, *7*, 87–93. [[CrossRef](#)]
3. Baatz, M.; Hoffmann, C.; Willhauck, G. Progressing from object-based to object-oriented image analysis. In *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*; Blaschke, T., Lang, S., Hay, G.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 29–42.
4. Lang, S. Object-based image analysis for remote sensing applications: Modeling reality—Dealing with complexity. In *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*; Blaschke, T., Lang, S.H.G.J., Eds.; Springer: Heidelberg/Berlin, Germany; New York, NY, USA, 2008.
5. Dong, L.; Du, H.; Mao, F.; Han, N.; Li, X.; Zhou, G.; Zhu, D.; Zheng, J.; Zhang, M.; Xing, L.; et al. Very High Resolution Remote Sensing Imagery Classification Using a Fusion of Random Forest and Deep Learning Technique—Subtropical Area for Example. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 113–128. [[CrossRef](#)]

6. Guo, H.; Shi, Q.; Marinoni, A.; Du, B.; Zhang, L. Deep building footprint update network: A semi-supervised method for updating existing building footprint from bi-temporal remote sensing images. *Remote Sens. Environ.* **2021**, *264*, 112589. [[CrossRef](#)]
7. Zhu, Q.; Zhang, Y.; Wang, L.; Zhong, Y.; Guan, Q.; Lu, X.; Zhang, L.; Li, D. A Global Context-aware and Batch-independent Network for road extraction from VHR satellite imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 353–365. [[CrossRef](#)]
8. Guo, H.; Du, B.; Zhang, L.; Su, X. A coarse-to-fine boundary refinement network for building footprint extraction from remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *183*, 240–252. [[CrossRef](#)]
9. Hosseinpour, H.; Samadzadegan, F.; Javan, F.D. CMGFNet: A deep cross-modal gated fusion network for building extraction from very high-resolution remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 96–115. [[CrossRef](#)]
10. Alam, M.; Wang, J.-F.; Guangpei, C.; Yunrong, L.; Chen, Y. Convolutional Neural Network for the Semantic Segmentation of Remote Sensing Images. *Mob. Netw. Appl.* **2021**, *26*, 200–215. [[CrossRef](#)]
11. Dong, S.; Quan, Y.; Feng, W.; Dauphin, G.; Gao, L.; Xing, M. A Pixel Cluster CNN and Spectral-Spatial Fusion Algorithm for Hyperspectral Image Classification with Small-Size Training Samples. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4101–4114. [[CrossRef](#)]
12. Pan, X.; Zhao, J. High-Resolution Remote Sensing Image Classification Method Based on Convolutional Neural Network and Restricted Conditional Random Field. *Remote Sens.* **2018**, *10*, 920. [[CrossRef](#)]
13. Jia, Z.; Lu, W. An End-to-End Hyperspectral Image Classification Method Using Deep Convolutional Neural Network with Spatial Constraint. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1786–1790. [[CrossRef](#)]
14. Tong, X.-Y.; Xia, G.-S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [[CrossRef](#)]
15. Långkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 329. [[CrossRef](#)]
16. Liu, J.; Wang, S.; Hou, X.; Song, W. A deep residual learning serial segmentation network for extracting buildings from remote sensing imagery. *Int. J. Remote Sens.* **2020**, *41*, 5573–5587. [[CrossRef](#)]
17. Huang, H.; Chen, P.; Xu, X.; Liu, C.; Wang, J.; Liu, C.; Clinton, N.; Gong, P. Estimating building height in China from ALOS AW3D30. *ISPRS-J. Photogramm. Remote Sens.* **2022**, *185*, 146–157. [[CrossRef](#)]
18. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: An applied review. *Int. J. Remote Sens.* **2018**, *39*, 2784–2817. [[CrossRef](#)]
19. Norman, M.; Shahar, H.M.; Mohamad, Z.; Rahim, A.; Mohd, F.A.; Shafri, H.Z.M. Urban building detection using object-based image analysis (OBIA) and machine learning (ML) algorithms. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *620*, 012010. [[CrossRef](#)]
20. Qian, Y.; Zhou, W.; Yan, J.; Li, W.; Han, L. Comparing Machine Learning Classifiers for Object-Based Land Cover Classification Using Very High Resolution Imagery. *Remote Sens.* **2015**, *7*, 153–168. [[CrossRef](#)]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
22. Conneau, A.; Lample, G. Cross-Lingual Language Model Pretraining. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 7059–7069.
23. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MI, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
24. Yu, P.; Fei, H.; Li, P. Cross-lingual Language Model Pretraining for Retrieval. In Proceedings of the Web Conference, Ljubljana, Slovenia, 12–16 April 2021; pp. 1029–1039. [[CrossRef](#)]
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021; p. 11929. [[CrossRef](#)]
26. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [[CrossRef](#)]
27. Ghaffarian, S.; Valente, J.; van der Voort, M.; Tekinerdogan, B. Effect of Attention Mechanism in Deep Learning-Based Remote Sensing Image Processing: A Systematic Literature Review. *Remote Sens.* **2021**, *13*, 2965. [[CrossRef](#)]
28. Lindsay, G.W. Attention in Psychology, Neuroscience, and Machine Learning. *Front. Comput. Neurosci.* **2020**, *14*, 29. [[CrossRef](#)] [[PubMed](#)]
29. Chen, K.; Zou, Z.; Shi, Z. Building Extraction from Remote Sensing Images with Sparse Token Transformers. *Remote Sens.* **2021**, *13*, 4441. [[CrossRef](#)]
30. Chen, C.F.R.; Fan, Q.; Panda, R. Crossvit: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 357–366.
31. Arkin, E.; Yadikar, N.; Xu, X.; Aysa, A.; Ubul, K. A survey: Object detection methods from CNN to transformer. *Multimedia Tools Appl.* **2022**, *27*, 1–31. [[CrossRef](#)]
32. Cao, F.; Lu, X. Self-Attention Technology in Image Segmentation. In Proceedings of the International Conference on Intelligent Traffic Systems and Smart City, Zhengzhou, China, 19–21 November 2021; p. 1216511. [[CrossRef](#)]
33. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2021**, *54*, 200. [[CrossRef](#)]

34. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [[CrossRef](#)]
35. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 568–578. [[CrossRef](#)]
36. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Montreal, QC, Canada, 6–14 December 2021; Volume 34, pp. 12077–12090. [[CrossRef](#)]
37. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021. Available online: <https://arxiv.org/abs/2103.14030> (accessed on 15 July 2022).
38. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. *arXiv* **2022**, arXiv:2107.00652. [[CrossRef](#)]
39. Bazi, Y.; Bashmal, L.; Al Rahhal, M.M.; Al Dayil, R.; Al Ajlan, N. Vision Transformers for Remote Sensing Image Classification. *Remote Sens.* **2021**, *13*, 516. [[CrossRef](#)]
40. Reedha, R.; Dericquebourg, E.; Canals, R.; Hafiane, A. Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images. *Remote Sens.* **2022**, *14*, 592. [[CrossRef](#)]
41. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training Data-Efficient Image Transformers & Distillation through Attention. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Volume 139, pp. 10347–10357. [[CrossRef](#)]
42. Gu, J.; Tresp, V.; Qin, Y. Are Vision Transformers Robust to Patch Perturbations? In *Computer Vision—ECCV 2022*; Lecture Notes in Computer Science; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer: Cham, Germany, 2022; p. 13672. [[CrossRef](#)]
43. Chen, J.; Liu, Y. Locally linear embedding: A survey. *Artif. Intell. Rev.* **2011**, *36*, 29–48. [[CrossRef](#)]
44. Zhou, Y.; Wang, F.; Zhao, J.; Yao, R.; Chen, S.; Ma, H. Spatial-Temporal Based Multihead Self-Attention for Remote Sensing Image Change Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6615–6626. [[CrossRef](#)]
45. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.; Tay, F.E.H.; Feng, J.; Yan, S. Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 538–547. [[CrossRef](#)]
46. Yuan, W.; Xu, W. MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer. *Remote Sens.* **2021**, *13*, 4743. [[CrossRef](#)]
47. Zhu, Q.; Li, Z.; Zhang, Y.; Li, J.; Du, Y.; Guan, Q.; Li, D. Global-Local-Aware conditional random fields based building extraction for high spatial resolution remote sensing images. *Natl. Remote Sens. Bull.* **2020**, *25*, 1422–1433. [[CrossRef](#)]
48. Liu, T.; Yao, L.; Qin, J.; Lu, N.; Jiang, H.; Zhang, F.; Zhou, C. Multi-scale attention integrated hierarchical networks for high-resolution building footprint extraction. *Int. J. Appl. Earth Obs.* **2022**, *109*, 102768. [[CrossRef](#)]
49. Ji, S.; Wei, S.; Lu, M. A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery. *Int. J. Remote Sens.* **2018**, *40*, 3308–3322. [[CrossRef](#)]
50. Pesaresi, M.; Gerhardinger, A. Improved Textural Built-Up Presence Index for Automatic Recognition of Human Settlements in Arid Regions with Scattered Vegetation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2011**, *4*, 16–26. [[CrossRef](#)]
51. Firmacek, B.; Unsalan, C. Urban Area Detection Using Local Feature Points and Spatial Voting. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 146–150. [[CrossRef](#)]
52. Li, Y.; Tan, Y.; Deng, J.; Wen, Q.; Tian, J. Cauchy Graph Embedding Optimization for Built-Up Areas Detection From High-Resolution Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2078–2096. [[CrossRef](#)]
53. Wang, J.; Yang, X.; Qin, X.; Ye, X.; Qin, Q. An Efficient Approach for Automatic Rectangular Building Extraction From Very High Resolution Optical Satellite Imagery. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 487–491. [[CrossRef](#)]
54. Du, S.; Du, S.; Liu, B.; Zhang, X. Incorporating DeepLabv3+ and object-based image analysis for semantic segmentation of very high resolution remote sensing images. *Int. J. Digit. Earth* **2021**, *14*, 357–378. [[CrossRef](#)]
55. Chen, H.; Lu, S. Building Extraction from Remote Sensing Images Using SegNet. In Proceedings of the 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), Xiamen, China, 5–7 July 2019; pp. 227–230. [[CrossRef](#)]
56. Chen, D.-Y.; Peng, L.; Li, W.-C.; Wang, Y.-D. Building Extraction and Number Statistics in WUI Areas Based on UNet Structure and Ensemble Learning. *Remote Sens.* **2021**, *13*, 1172. [[CrossRef](#)]
57. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 645–657. [[CrossRef](#)]
58. Tong, Z.; Li, Y.; Li, Y.; Fan, K.; Si, Y.; He, L. New Network Based on Unet++ and Densenet for Building Extraction from High Resolution Satellite Imagery. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 2268–2271. [[CrossRef](#)]
59. Yu, S.; Xie, Y.; Liu, C. Building extraction from remote sensing image based on improved segnet neural network and image pyramid. *J. Phys. Conf. Ser.* **2020**, *1651*, 012145. [[CrossRef](#)]

60. Angelis, G.-E.; Domi, A.; Zamichos, A.; Tsourma, M.; Drosou, A.; Tzovaras, D. On The Exploration of Vision Transformers in Remote Sensing Building Extraction. In Proceedings of the 2022 IEEE International Symposium on Multimedia (ISM), Naples, Italy, 5 December 2022; pp. 208–215. [\[CrossRef\]](#)
61. Cui, L.; Jing, X.; Wang, Y.; Huan, Y.; Xu, Y.; Zhang, Q. Improved Swin Transformer-Based Semantic Segmentation of Postearthquake Dense Buildings in Urban Areas Using Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 369–385. [\[CrossRef\]](#)
62. Yuan, W.; Zhang, X.; Shi, J.; Wang, J. LiteST-Net: A Hybrid Model of Lite Swin Transformer and Convolution for Building Extraction from Remote Sensing Image. *Remote Sens.* **2023**, *15*, 1996. [\[CrossRef\]](#)
63. Sun, Z.; Zhou, W.; Ding, C.; Xia, M. Multi-Resolution Transformer Network for Building and Road Segmentation of Remote Sensing Image. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 165. [\[CrossRef\]](#)
64. Wang, L.; Fang, S.; Meng, X.; Li, R. Building Extraction with Vision Transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [\[CrossRef\]](#)
65. Xiao, X.; Guo, W.; Chen, R.; Hui, Y.; Wang, J.; Zhao, H. A Swin Transformer-Based Encoding Booster Integrated in U-Shaped Network for Building Extraction. *Remote Sens.* **2022**, *14*, 2611. [\[CrossRef\]](#)
66. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450. Available online: <http://arxiv.org/abs/1607.06450> (accessed on 30 July 2022).
67. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified Perceptual Parsing for Scene Understanding. In Proceedings of the Lecture Notes in Computer Science, Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 432–448. [\[CrossRef\]](#)
68. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [\[CrossRef\]](#)
69. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.