remote sensing

MDPI

Article

Two-Stream Swin Transformer with Differentiable Sobel Operator for Remote Sensing Image Classification

Siyuan Hao [1], Bin Wu [1], Kun Zhao [1,*], Yuanxin Ye [2] and Wei Wang [3]

[1] The College of Communication and Electronic Engineering, Qingdao University of Technology, Qingdao 266520, China; haosiyuan@qut.edu.cn (S.H.); wubin970623@gmail.com (B.W.)
[2] The Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 610031, China; yeyuanxin@home.swjtu.edu.cn
[3] The Department of Information Engineering and Computer Science, University of Trento, Via Sommarive, 38123 Trento, Italy; wei.wang@unitn.it
* Correspondence: zhaokun@qut.edu.cn

Abstract: Remote sensing (RS) image classification has attracted much attention recently and is widely used in various fields. Different to natural images, the RS image scenes consist of complex backgrounds and various stochastically arranged objects, thus making it difficult for networks to focus on the target objects in the scene. However, conventional classification methods do not have any special treatment for remote sensing images. In this paper, we propose a two-stream swin transformer network (TSTNet) to address these issues. TSTNet consists of two streams (i.e., original stream and edge stream) which use both the deep features of the original images and the ones from the edges to make predictions. The swin transformer is used as the backbone of each stream given its good performance. In addition, a differentiable edge Sobel operator module (DESOM) is included in the edge stream which can learn the parameters of Sobel operator adaptively and provide more robust edge information that can suppress background noise. Experimental results on three publicly available remote sensing datasets show that our TSTNet achieves superior performance over the state-of-the-art (SOTA) methods.

Keywords: remote sensing; scene classification; deep learning; swin transformer; feature fusion; edge detection

check for updates

Citation: Hao, S.; Wu, B.; Zhao, K.; Ye, Y.; Wang, W. Two-Stream Swin Transformer with Differentiable Sobel Operator for Remote Sensing Image Classification. *Remote Sens.* **2022**, *14*, 1507. https://doi.org/10.3390/rs14061507

Academic Editors: Xiangtao Zheng, Fulin Luo and Qi Wang

Received: 15 February 2022
Accepted: 16 March 2022
Published: 20 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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/).

1. Introduction

Remote sensing images are valuable sources to help us make Earth observations [1–5]. With the advancement of Earth observation technologies [6,7], the number of high-resolution remote sensing images is increasing dramatically. Therefore, understanding the complex and large amount of remote sensing images becomes very important, and remote sensing image classification has become an important task and has received more and more attention. The scene classification techniques have been widely used in urban planning [8,9], geographic image retrieval [10,11], environment monitoring [12–14], vegetation mapping [15], and geospatial object detection [16–23].

In the last decades, many remote sensing image classification methods have been proposed which are either based on the traditional handcrafted features or the deep learning ones. The deep learning methods include deep belief neural networks (DBNNs), convolutional neural networks (CNNs), and vision transformer (ViT). Cheng et al. [24] first used CNN for feature extraction, and then used SVM for classification. Zhou et al. [25] fine-tuned a pretrained CNN on a remote sensing dataset which can learn in an end-to-end manner. In addition, the attention mechanism has also been introduced into the field of remote sensing imaging by Wang et al. [26] and they proposed a recurrent attention mechanism.

In recent years, the transformer has achieved remarkable performance in the field of natural language processing (NLP) and it has also been introduced into the field of

image classification. Vision transformer (ViT) [27] proposed by Dosoviskiy et al. is the first transformer variant that can be applied into image classification. Recently, the transformer has also been introduced in remote sensing image scene classification, and most of the recent works are based on the classical ViT models. For instance, Bazi et al. [28] transferred the pretrained ViT model with data augmentation and network pruning to boost the classification performance. Deng et al. [29] proposed a joint model named CTNet for remote sensing image classification in which the ViT is used to extract semantic features of HRRS images and CNNs are adopted to compensate for the loss of local structural information. The TRSNet [30] proposed by Zhang et al. optimizes the residual network and adds transformer blocks to achieve better interaction between the transformer and the CNN. Although ViT is currently the dominant vision transformer network, it still has many shortcomings. For example, ViT produces feature maps of a single low-resolution patch and has quadratic computation complexity w.r.t the input image size due to the computation of global self-attention, and ViT focuses too much on long-distance semantic information while ignores local structural features such as those extracted by CNNs. However, the shifted windows (swin) transformer proposed by Liu et al. [31] solves these problems by introducing some CNN features and constructing hierarchical feature maps by merging image patches. Besides, the swin transformer has linear computation complexity w.r.t the input image size. Therefore in this paper, we employ the swin transformer as the original stream in our two-stream architecture to deal with the remote sensing images.

In addition, the traditional methods mentioned above, whether based on CNN, ViT, or swin transformer, are designed for natural images and they do not take into account the characteristics of remote sensing images. Remote sensing images are usually large in size, high in resolution, and cover a large amount of ground objects. More importantly, since the RS scene images are composed of various ground objects which are stochastically distributed, they are difficult to recognize effectively. The edge curves are more representative as they can link the stochastically distributed objects together. Figure 1 shows six examples, namely "center", "river", "square", "port", "baseball field", and "park". The edges of the main object in the scene are depicted with red curves. As can be seen in Figure 1, in the scene of "center", the circular building of center is composed of several different parts which are surrounded by edge curves. In addition, it is well known that compared with the natural images, remote sensing images have relatively large intra-class differences and small inter-class differences. However, the same scenes have similar edge shapes even though they are observed from different views by different satellites. Therefore, we rely on edges, which are more representative and robust to help the networks to accurately recognize the scenes. As a result, we add an auxiliary stream, named edge stream, to deal with the edge information.

In order to effectively extract the edge information, we propose a differentiable edge Sobel operator module (DESOM). Different to other approaches using the default values of Sobel operator directly for edge extraction, we use a differentiable Sobel operator to extract the edges. In particular, the Sobel operator is set as a convolution kernel to extract the edges in the x-direction and y-direction, and the default values are only used for the initialization purpose. The underlying reason to make it differentiable is that the Sobel operator can be more flexible and it can learn to adapt its parameters automatically to better serve the classification task. However, a single edge image that serves as the input of the auxiliary edge stream completely ignores the content, leading to a poor accuracy. Therefore, we stack the grayscale image containing rich content information with the double channel edge image together, forming a novel three-channel image as the input to the edge stream.

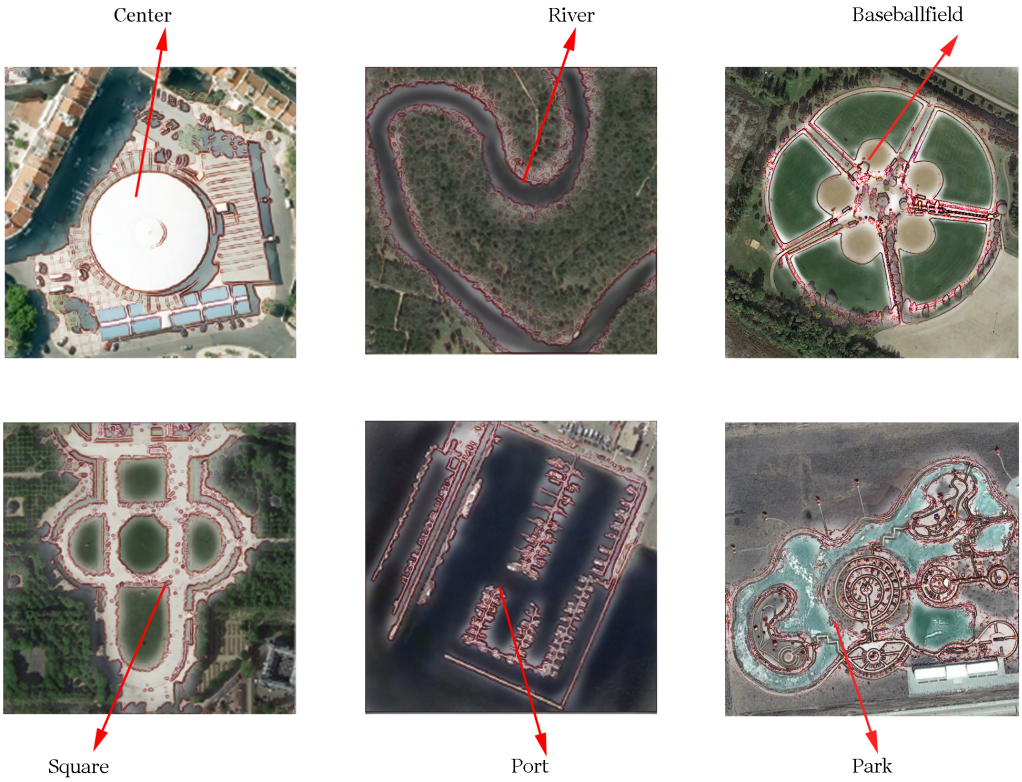


Figure 1. Edge information of the six scenes; the red curves represent the edge of the scene object.

The features extracted from two streams are concatenated and fused by a fully connected layer which is followed by a weighted cross-entropy loss. To emphasize the importance of the first stream, an auxiliary loss is added to enlarge the gradients backpropagated to the first stream. The details of the loss are available in later section.

In summary, the main contributions of this paper are as follows:

(1) A two-stream swin transformer network (TSTNet) is designed for remote sensing image classification. The first stream is in charge of extracting the original image features while the second stream is in charge of extracting the edge information.
(2) To extract more powerful edge features for the remote sensing images, a novel differentiable edge Sobel operator module (DESOM) is proposed in the second stream.
(3) We have designed a weighted feature fusion module to enable the edge information and the original image information to be fused effectively to boost the classification performance.

The rest of the paper is organized as follows. Section 1 gives a brief introduction of the background and motivation of this paper. Section 2 introduces the related works. Section 3 introduces the details of the two-stream swin transformer. In Section 4, the experimental results and analysis are reported. Finally, the conclusiond is given in Section 6.

2. Related Works

In this section, we briefly review the remote sensing image classification approaches including hand-crafted feature-based, CNN-based, and transformer-based methods.

2.1. Hand-Crafted Features

The early scene classification methods are mainly based on hand-crafted features. These features include texture, color, shape, spectrum, etc. There are many classical methods to extract handcrafted features, such as color histograms [32], texture descriptors [33], GIST [34], scale-invariant feature transform (SIFT) [35], and histogram of oriented gradients (HOG) [36]. However, as remote sensing images become higher in resolution and larger

in size, and contain richer and more obscure information, it is very difficult for the hand-crafted features to deliver good performance.

2.2. CNNs in Remote Sensing Image Classification

Deep learning approaches have also been introduced into the field of remote sensing. Some popular CNN frameworks are widely used, such as AlexNet [37], VGGNet [38], ResNet [39], and DenseNet [40], all of which have become the common backbone to process remote sensing images. Inspired by the attention mechanism, Fan et al. [41] proposed a CNN-based network that combines residual units and attention mechanism. It automatically assigns large weights to key areas of the image and thus the network has the ability to ignore redundant information. CNNs are suitable for common images, but sometimes they do not consider the characteristics of the remote sensing images, and often lose spatial information. Therefore, Zhang et al. [42] proposed CapsNet, a novel network architecture that uses a group of neurons as a capsule or vector to replace the neuron in the traditional neural network to better extract the spatial information. In addition, features at different scales of the CNN contain different information, and aggregating these features can improve the classification accuracy. For instance, the GBNet proposed by Wang et al. [43] aggregates the features at different scales while eliminating the interference information in the aggregation process. As CNN models become more and more complex in structure, the learned low-level features become less representative and interpretable. Recently, Wang et al. [44] introduced lie group machine learning into the CNN model, and a combination of the two to improve the interpretability of the model.

2.3. Transformer in Vision

Recently, vision transformer (ViT) has been introduced to scene classification and has achieved good performance. ViT is the first to beat the classical CNNs (e.g., GoogleNet, ResNet, and DenseNet) on most image recognition datasets. Thus, ViT shows the effectiveness of the transformer in the field of computer vision. However, ViT also has its limitations as it needs more training data to converge well. To address this issue, Touvron et al. proposed the data-efficient image transformer (DeiT) [45] to solve the problem. DeiT uses a good CNN as a teacher to distill knowledge to teach ViT networks as students. This enables DeiT to achieve good performance even when trained on the small Image-1k dataset. Although ViT is good at long-range semantic understanding, the local structure also needs to be well understood, which is ignored by ViT. To solve this problem, Liu et al. proposed a shifted windows (swin) [31] transformer to extract both global and local features using a sliding window to shift in the spatial dimension. In this way, both long-distance information and local structure information are extracted. Currently, researchers try to apply transformers to remote sensing tasks. For instance, Li et al. [46] proposed a multi-stream fusion network for remote sensing spatiotemporal fusion based on transformers and CNNs. Xu et al. [47] combined the swin transformer and UPerNet for remote sensing image segmentation. However, these conventional methods used the transformer structure directly without taking into account the characteristics of remote sensing images.

2.4. Application of Edge Information in Remote Sensing

Edge information has been applied for image segmentation tasks but it has been less studied for remote sensing image classification tasks. Constraints on the network by adding prior knowledge [48] in remote sensing image classification can improve the classification performance. Motivated by the successful application of prior knowledge in remote sensing image classification, He et al. [49] argue that edge information can also be used as prior knowledge. This is because edge information reflects significant changes in image brightness, eliminates irrelevant information in the image, and preserves important structural properties. Therefore, He et al. [49] corrected the segmentation results of a fully convolution network (FCN) by edge information. In addition, Xu et al. [47] achieved good

results in the remote sensing image segmentation task by overlaying the edge information with the original remote sensing image. Therefore, edge information, as a constraint, can improve the classification performance of the network.

3. Proposed Method

3.1. Overall Framework

As shown in Figure 2, we proposed a two-stream swin transformer architecture. The first stream extracts the features of the original images I and the second stream extracts the features of edge images T(I). The two streams do not share parameters.

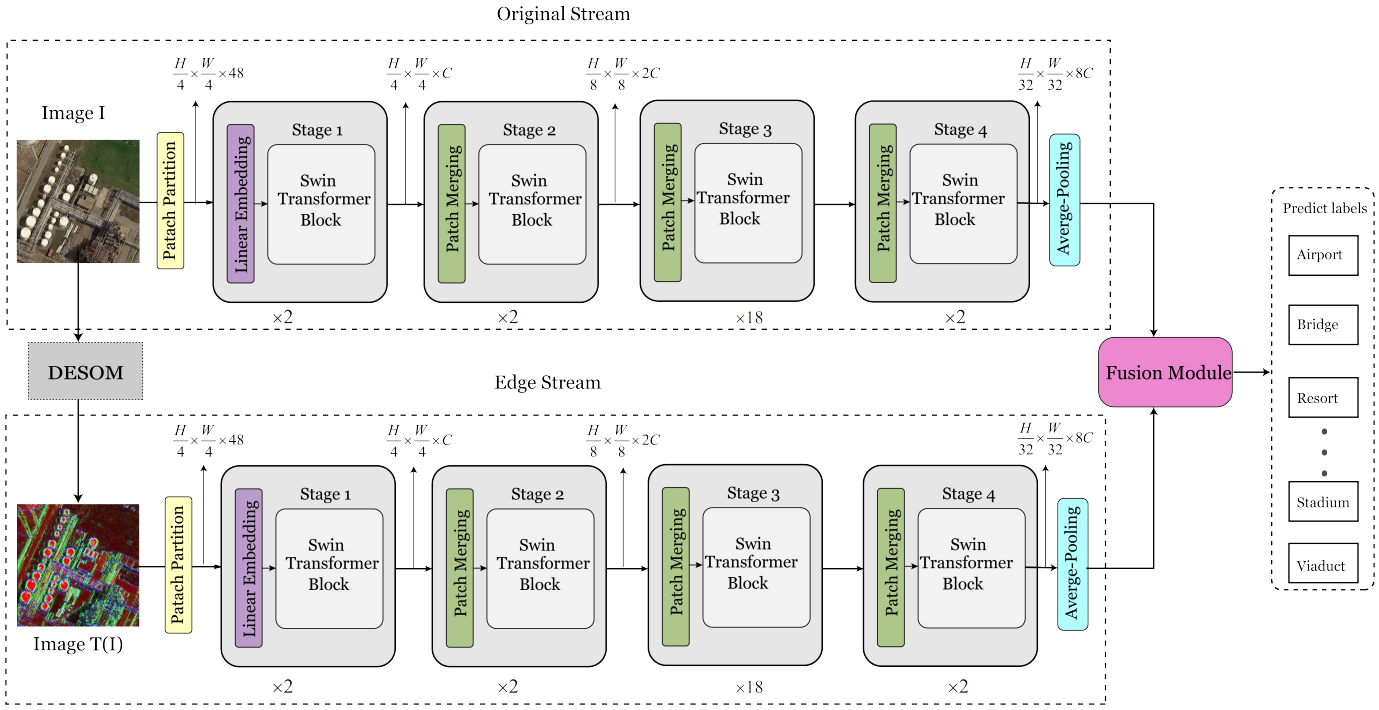


Figure 2. The framework of the two-stream swin transformer. It has two inputs, the original image (I) and the edge image T(I), and the synthetic edge image T(I) generated by differentiable edge Sobel operator module (DESOM). The fusion module fuses the original features and edge features.

In addition, the edge image T(I) is synthesized by DESOM with the function of adaptive learning of remote sensing image edge information. After extracting the features of the original stream and the edge stream, they are concatenated and fused by a fully connected layer. We have designed a weighted loss function. Finally, the classification is completed by softmax. The details are discussed in the following subsections.

3.2. Backbone for Feature Extraction

The overall structure of the swin transformer is shown in Figure 2. By assigning different values to the block numbers and token dimensions, we end up with four versions of swin transformer, Swin-T, Swin-S, Swin-B, and Swin-L, in which *T*, *S*, *B*, *L* refer to the amount of parameters (i.e., tiny, small, base, and large). In this paper, we use Swin-B as our backbone as it has a good balance between the performance and model size. In Figure 2, we can see that Swin-B contains four stages: stage 1 has one linear projection layer and two swin transformer blocks, and the remaining three stages contain a patch merging layer and an even number of swin transformer blocks. The patch merging layer is used to adjust the resolution of the features, usually performing a $\times 2$ resolution drop.

As shown in Figure 2, we usually set the patch size to 4×4 in the initial stage, so on a three-channel image, each patch is $4 \times 4 \times 3 = 48$ after flattening. $\frac{H}{4} \times \frac{W}{4}$ here indicates the number of patches which also represents the resolution of features. In stage 1, a linear

embedding layer is applied on this raw-valued feature to project the flattened 48 patch vector to an arbitrary dimension (denoted as C) and the resolution of features is maintained at $\frac{H}{4} \times \frac{W}{4}$ by several swin transformer blocks. In order to have a hierarchical representation, the swin transformer introduces a patch merging layer to achieve a reduced resolution of features by $\times 2$ downsampling. Therefore, the output feature resolutions of the four stages are $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$. While patch merging performs the merging of neighboring patches and reduces the dimension by two along the dimensions of height and width, the channel dimension will be increased correspondingly, and the output feature dimensions of the four stages are set to c, $2c$, $4c$, and $8c$. After four stages, the final output is $N \times C$, where $N = \frac{H}{32}\frac{W}{32}$ is the token number and $C = 8c$ is the token dimension. Different to ViT, which selects a class token for classification, swin transformer performs average-pooling in N dimensions and places $1 \times C$ into the classifier.

Figure 3 shows the structure of stage 1 which consists of two successive swin transformer blocks, i.e., W-MSA and SW-MSA. W-MSA and SW-MSA denote window-based multi-head self-attention (MSA) using regular window and shifted window partitioning configurations, respectively. The formulas of two consecutive swin transformer blocks are as follows:

$$\begin{aligned} \hat{z}^l &= \text{W-MSA}(\text{LN}(z^{l-1})) + z^{l-1}, \\ z^l &= \text{MLP}(\text{LN}(\hat{z}^l)) + \hat{z}, \\ \hat{z}^{l+1} &= \text{SW-MSA}(\text{LN}(z^l)) + z^l, \\ z^{l+1} &= \text{MLP}(\text{LN}(\hat{z}^{l+1})) + \hat{z}^{l+1}, \end{aligned} \tag{1}$$

where $\hat{z}^l$ and z^l represent the outputs of (S)W-MSA and multilayer perception (MLP) on block l. Each input in the formula is normalized by layer norm (LN).

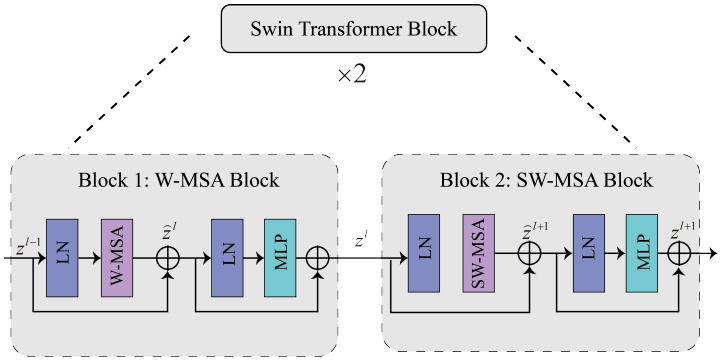


Figure 3. Two successive transformer blocks of the swin transformer. The regular and shifted windows correspond to W-MSA and SW-MSA, respectively.

Different to the ordinary MSA in vision transformer that performs the computation of self-attention directly on the global relations, the W-MSA of the swin transformer uses a window containing $M \times M$ patches (M set to 7 by default) to compute self-attention. The computational complexity is given in the following equation, where the original MSA is quadratic in the number of patches $h \times w$, while the window-based MSA is linear.

$$\begin{aligned} \Omega(\text{MSA}) &= 4hwC^2 + 2(hw)^2C \\ \Omega(\text{W-MSA}) &= 4hwC^2 + 2M^2hwC \end{aligned} \tag{2}$$

W-MSA reduces the computational complexity of self-attention. The computation of attentional relations is performed only within the window, and the global relations are ignored. Therefore, SW-MSA is proposed to encode the global relations. SW-MSA achieves communication across windows by shifting and partitioning operations. The specific shifting and partition mechanism is shown in Figure 4. In Figure 4, the features are first partitioned into a standard window (a), and then (a) is fed into W-MSA to compute attentions between every two individual patches inside the window. Then, (a) is converted

to (b) by another partitioning strategy, and (b) shifts cyclically to (c). Next, (c) is fed to SW-MSA for intra-window attention calculation, so that information interaction across windows is achieved. Finally, (c) is reversed to (e) in order to move to the next loop. For the details, please refer to [31]. In summary, the swin transformer can extract better remote sensing image features compared to ViT, thus we use it as the backbone of our model.

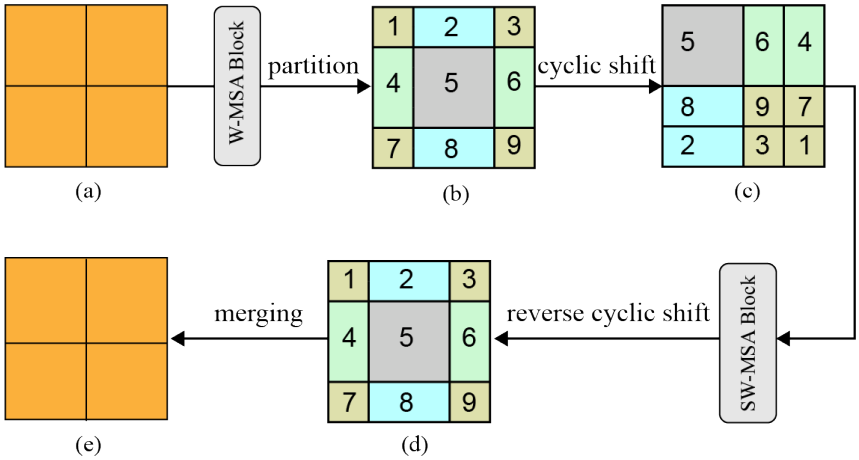


Figure 4. Shifted window mechanism of W-MSA and SW-MSA. (**a**) is the standard window, (**b**) is the repartitioned window, (**c**) is the window after the shift transformation, and (**d**) is the window after the (**e**) reverse shift transformation.

3.3. Differentiable Edge Sobel Operator Module

Different to the ordinary images, remote sensing images usually have larger image resolution and more background noise. Therefore, it is important to eliminate the background noise. To achieve this, we design a differentiable edge Sobel operator module (DESOM). It can extract the image edge information and suppress the scene's irrelevant factors. Meanwhile, during the network learning process, the module parameters are continuously updated to make the learned edges more robust such that they can help the network focus on the scene object. The edge stream and the original stream complement each other, and the fusion of the two streams achieves better performance than using each of them independently.

Many edge detection methods are available, and we choose the most widely used Sobel operator to extract edge information. The Sobel operator is a discrete difference operator which computes the gradient approximation of the image luminance function. Conceptually, the Sobel operator is a small integer filter that convolves the whole image in horizontal and vertical directions, and therefore requires few computing resources. It contains two sets of 3×3 matrices that calculate the luminance difference approximation, or gradient, of the image horizontally and vertically. The form of the operator along the horizontal X and vertical Y directions is as follows:

$$\begin{aligned} G_x &= \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A, \\ G_y &= \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A, \end{aligned} \tag{3}$$

where A represents the grayscale map of the original image, and G_x and G_y represent the edge maps of grayscale images in horizontal and vertical directions, respectively.

Based on the principle of Sobel operator edge detection, we designed the DESOM, as shown in Figure 5. We first converted the original remote sensing image into a grayscale image. Then, the edges of the images in the X-direction and Y-direction were extracted by

two 3×3 convolutional layers, which are represented by the orange convolutional layer and the blue convolutional layer, respectively. The weights of the convolution kernels in the orange and blue convolution layers are initialized by the Sobel operator in the X-direction and Y-direction.

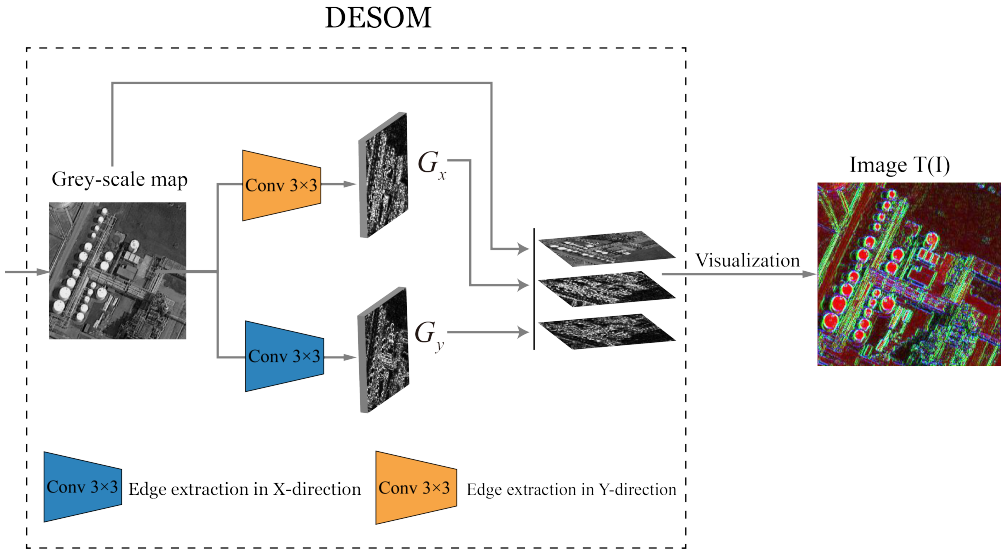


Figure 5. Structure of differentiable edge Sobel operator module (DESOM).

The conventional Sobel edge operator uses default values to extract edges, while our DESOM makes the parameters differentiable. Sobel operators are defined as convolution kernels such that they are able to be updated and fine-tuned during network backpropagation. In this way, the Sobel operator is more flexible and it can learn its own parameters adaptively and extract edges that better match the demands of the classification task. The edge images G_x, G_y in the X-direction and Y-direction are homogeneous and ignore the texture information. To compensate for this, we also add the grayscale map of the image as the input. Finally, we obtain the synthetic three-channel image by stacking G_x , G_y, and the grayscale image together.

3.4. Feature Fusion

Figure 6 shows the details of the fusion module and loss function. F_1 and F_2 denote the original feature and the edge feature which are generated by the pooling layer at the end of each stream, each of which is a vector with dimension d. They are fused in the following way:

$$F' = FC(\text{Concat}(F_1, F_2)) \tag{4}$$

where the final classification vector $F' \in \mathbb{R}^n$ (n denotes the number of categories) is generated by a fully connected (FC) layer that allows the original information and the edge information to be effectively fused.

To emphasize the importance of the first stream, an auxiliary loss is added to the final loss to enlarge the gradients backpropagated to the first stream. As shown in Figure 6, the final loss L is

$$L = \lambda L_F + (1 - \lambda) L_{F1} \tag{5}$$

where L_F is the cross-entropy loss of the fused features $F^{'}$. L_{F1} is the cross-entropy loss of F_1. The original stream plays an important role in the final classification, such that we assign an auxiliary loss to the original stream and add it to the final loss. In order to balance the two losses, we introduce a weight λ with the constraint that the sum of the weights of the two loss functions equals 1 and $\lambda \in [0, 1]$. It has been experimentally demonstrated that $\lambda = 0.8$ can achieve a good balance, which brings good performance.

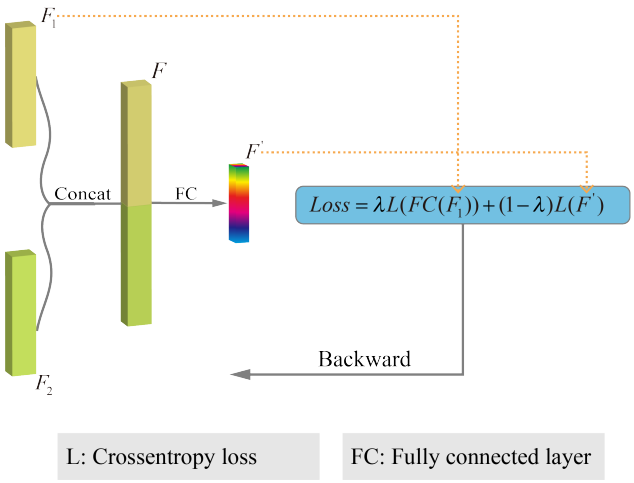


Figure 6. Design of the fusion module and loss function. An auxiliary cross-entropy loss about F_1 is added to the cross-entropy loss of the fused feature F'. The balance between the two loss functions is controlled by λ.

4. Experiments

In this section, we evaluate our proposed method TSTNet on three publicly available datasets for remote sensing image classification. First, we introduce the dataset used and describe the experimental setup. Second, the effectiveness of our proposed structure is demonstrated through ablation experiments. Finally, our proposed method is compared with the state-of-the-art methods.

4.1. Datasets

We test the proposed method on three remote sensing images datasets, namely Aerial Image dataset (AID) [50], NWPU-RESISC45 dataset (NWPU) [51], and UCMerced Land-Use dataset (UCM) [52].

(1) AID: As shown in Figure 7, the large image size and rich scene coverage of the AID dataset make it suitable as a baseline dataset for high-level networks. The images are collected in different countries and regions, at different times of the year, in different seasons, and under different imaging conditions, covering 30 scenes. The number of sample images per scene varies widely, ranging from 220 to 420 images. In addition, the size of each image is 600 $\times$ 600 pixels and the resolution of each pixel is 8 to 0.5 m.

(2) NWPU: As shown in Figure 8, the NWPU dataset is a large-scale, scene-rich dataset that is challenging for deep learning networks. A total of 31,500 remote sensing images are divided into 45 scenes. The images are extracted from Google Earth, and the resolution of each pixel is about 30 to 0.2 m. The size of each image is 256 $\times$ 256 pixels. The AID dataset and the NWPU dataset are more challenging datasets and are generally used to test some high-performance image classification methods.

(3) UCM: As shown in Figure 9, this dataset is one of the most used classical datasets. It has fewer images compared to the AID and NWPU datasets, so it is much less difficult to classify. A total of 21,000 remote sensing images are available, with 21 scenes, so each scene has 100 images. The images in the dataset are derived from the United States Geological Survey (USGS). The size of each image is 256 $\times$ 256 pixels.

Figure 7. Example images of AID dataset: (1) airport; (2) bare land; (3) baseball field; (4) beach; (5) bridge; (6) centre; (7) church; (8) commercial; (9) dense residential; (10) desert; (11) farmland; (12) forest; (13) industrial; (14) meadow; (15) medium residential; (16) mountain; (17) park; (18) parking; (19) playground; (20) pond; (21) port; (22) railway station; (23) resort; (24) river; (25) school; (26) sparse residential; (27) square; (28) stadium; (29) storage tanks; (30) viaduct.

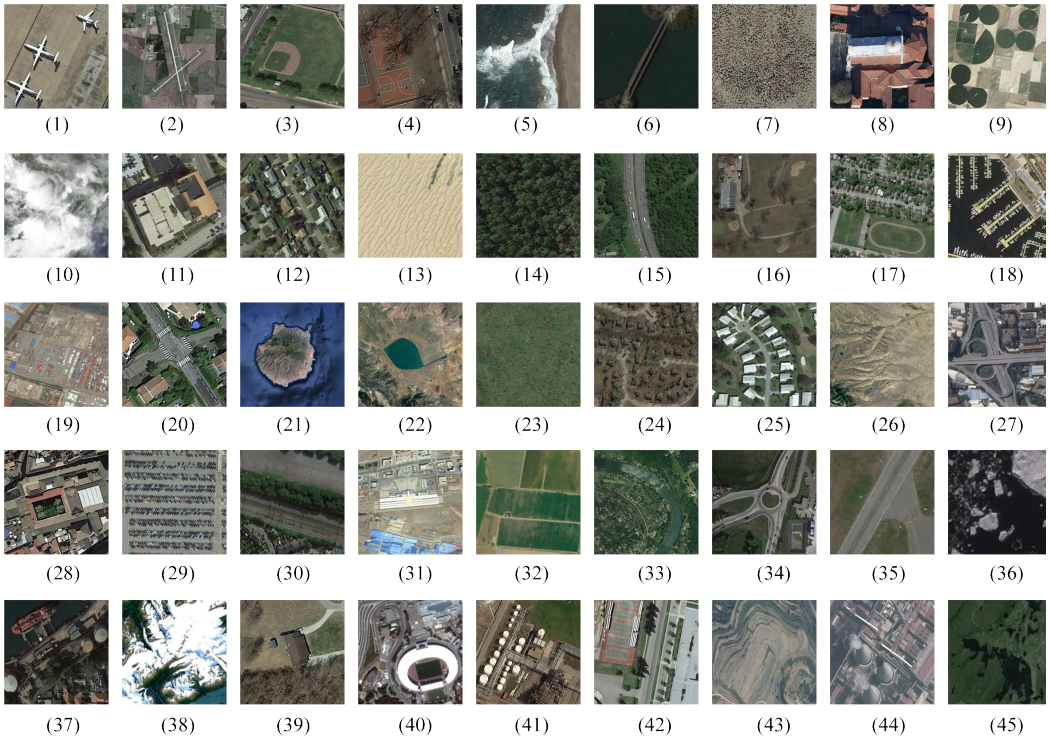


Figure 8. Example images of NWPU dataset: (1) airplane; (2) airport; (3) baseball diamond; (4) basketball court; (5) beach; (6) bridge; (7) chaparral; (8) church; (9) circular farmland; (10) cloud; (11) commercial area; (12) dense residential; (13) desert; (14) forest; (15) freeway; (16) golf course; (17) ground track field; (18) harbor; (19) industrial area; (20) intersection; (21) island; (22) lake; (23) meadow; (24) medium residential; (25) mobile home park; (26) mountain; (27) overpass; (28) palace; (29) parking lot; (30) railway; (31) railway station; (32) rectangular farmland; (33) river; (34) roundabout; (35) runway; (36) sea ice; (37) ship; (38) snow berg; (39) sparse residential; (40) stadium; (41) storage tank; (42) tennis court; (43) terrace; (44) thermal power station; (45) wetland.

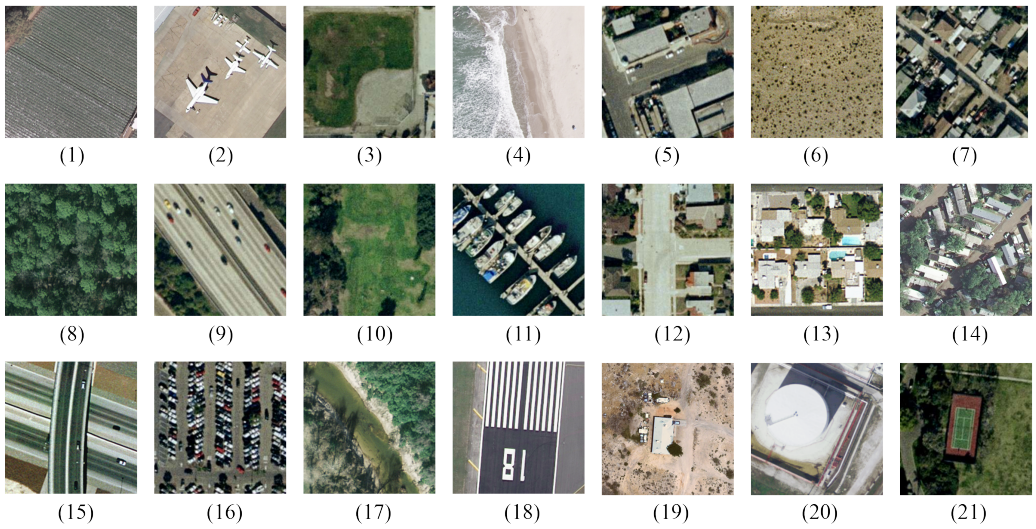


Figure 9. Example images of UCM dataset: (1) agriculture, (2) airplane, (3) baseball diamond, (4) beach, (5) buildings, (6) chaparral, (7) dense resi- dential, (8) Forest, (9) Freeway, (10) golf course, (11) harbor, (12) intersection, (13) medium residential, (14) mobile home park, (15) overpass, (16) parking lot, (17) river, (18) runway, (19) sparse residential, (20) storage tanks, and (21) tennis court.

4.2. Experimental Setup

All experiments were implemented using Pytoch 1.8. The models were trained on NVIDIA RTX 3090 (24 GB) GPU $\times$ 4 in a single machine. We used Swin-B (with 70 M parameters) as the backbone of each stream. Two input image sizes, 224 $\times$ 224 pixels and 384 $\times$ 384 pixels, were used for different datasets. Swin-B was used as the backbone, and we removed its fully connected layer. In addition, the pretrained parameters of ImageNet-21K were loaded on Swin-B (except the fully connected layer). The official pretrained parameters are found at https://github.com/microsoft/Swin-Transformer (accessed on 1 March 2022). When training TSTNet, we chose the Adamw optimizer with an initial learning rate of 0.0001 and a weight decay of 0.05 to optimize the parameters of TSTNet. In the optimizer, we improved the stability of the model by taking a 10-epoch warm-up strategy. Considering the limited GPU memory resource, we set the batch size to 16 for the AID dataset and to 32 for the NWPU and UCM datasets.

Image Size: For the AID dataset, we reshaped the images to 384 $\times$ 384 pixels in order to reduce content loss. For the NWPU and UCM datasets, the image size was set to 224 $\times$ 224. We also implemented data augmentation techniques, such as random flipping, random rotation, random erasing, etc.

Training & Testing Ratios: We split the dataset into training and testing samples according to the commonly used division ratio. Among them, the AID dataset used training ratios of 20% and 50%, the NWPU dataset used 10% and 20%, and the UCM dataset used 50% and 80%.

Evaluation Metric: Common evaluation metrics for scene classification include overall accuracy (OA) and confusion matrix (CM). In addition, all experiments were repeated ten times in order to reduce the random influence. The experimental results are mean overall accuracy with standard deviation (OA $\pm$ STD).

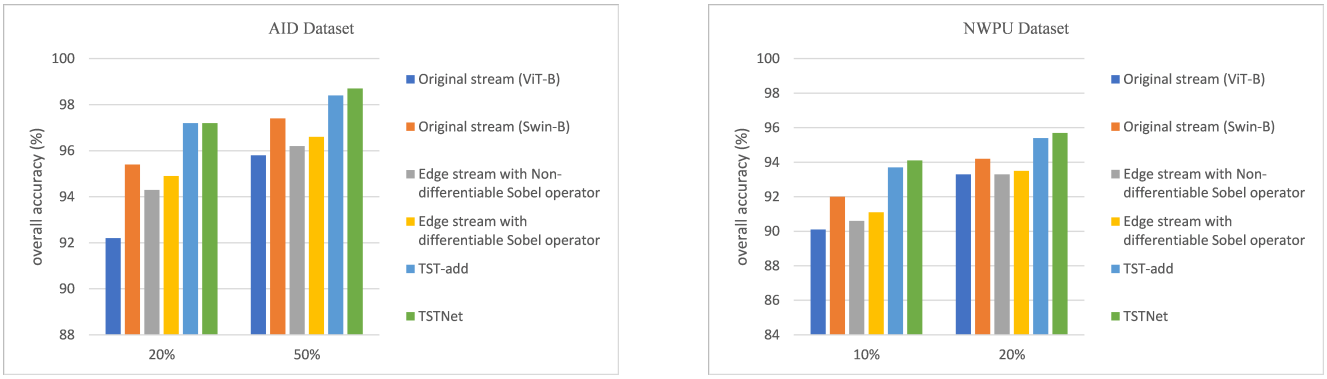
4.3. Ablation Study

The proposed TSTNet extracts features through a two-stream swin transformer, which includes an original stream and an edge stream with DESOM. Finally, a fusion module is added to fuse the features. Therefore, to demonstrate the effectiveness of the proposed method, we designed a series of ablation experiments using the AID dataset and NWPU dataset with different training ratios. In addition, we also visualized the results of the attention mapping.

(1) Analysis of ablation experimental results:

- First, to demonstrate the good performance of Swin-B over ViT-B, we conducted experiments using a single-branch original stream which was either based on ViT-B or Swin-B.
- In addition, to demonstrate the effectiveness of DESOM, we compared the edge stream using Sobel operator with fixed values and edge stream with differentiable Sobel operator whose values can be learned adaptively.
- Finally, our proposed architecture (i.e., TSTNet) concatenates the features from the two branches directly and it has a weighted loss function which balances the contribution of the two streams. To check the effectiveness of the loss function, we replaced the concatenation operation with a simple add operation. This baseline is named TST-add and uses a simple cross entropy loss treating the two streams equivalently.

Swin-B vs. ViT-B: Figure 10 shows the accuracy histograms of different architectures on different datasets with different training ratios. In Figure 10a,b, we can observe that the accuracy of Original (Swin-B) is consistently higher than Original (ViT-B). To be specific, the Swin-B-based branch is 3.2% and 1.6% higher than the ViT-B-based one on the AID dataset with two different training/testing split ratios. Similarly, on the NWPU dataset, the Swin-B-based branch is 1.9% and 0.9% higher than the ViT-B-based one with two different training/testing split ratios. This observation shows that Swin-B consistently outperforms ViT-B, and we use Swin-B as our backbone.



(**a**) (**b**)

Figure 10. Ablation Study: Accuracy comparison between different methods under the training ratios of (**a**) 20% and 50% on AID dataset; (**b**) 10% and 20% on NWPU dataset. Our proposed method, TSTNet, consistently has the best performance across different datasets with different training ratios.

Differentiable vs. Non-differentiable Sobel Operator: The initial and final learned values of X-operator and Y-operator on different datasets are reported in Figure 11. We can observe that different values of X-operator and Y-operator are learned, which suggests that our differentiable operator can be tuned adaptively according to different datasets and better serves the final classification task. In other words, the differentiable operator can bring extra performance gain. As shown in Figure 10a,b, the accuracy of edge stream with differentiable Sobel operator is consistently higher than edge with non-differentiable Sobel operator. To be specific, the edge stream with differentiable Sobel operator is 0.6% and 0.4% higher than the edge with non-differentiable Sobel operator on the AID dataset with two different training/testing split ratios. Similarly, on the NWPU dataset, the edge stream with differentiable Sobel operator is 0.5% and 0.2% higher than the edge with non-differentiable Sobel operator with two different training/testing split ratios. This shows that our DESOM is effective and has better edge feature extraction capability.

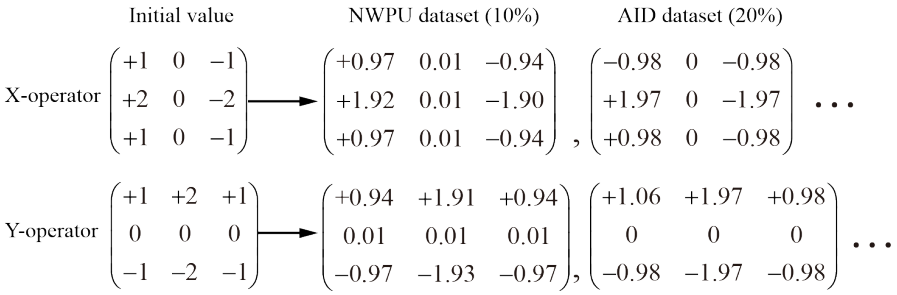


Figure 11. The optimal values learned by the differentiable Sobel operator under the training ratio of 10% on the NWPU dataset and under the training ratio of 20% on the AID dataset.

Weighted Loss vs. Original Loss: In Figure 10a,b, our proposed TSTNet with the weighted loss shows a small improvement in accuracy over TST-add with the original cross-entropy loss on the AID dataset and the NWPU dataset. The TSTNet is 0.3% higher than TST-add on the AID dataset with 50% training ratio, while they have the same performance with 20% training ratio. Similarly, on the NWPU dataset, the TSTNet is 0.4% and 0.3% higher than the TST-add with two different training ratios.

With vs. Without Edge Stream: Finally, our proposed method TSTNet, which incorporates both original and edge streams, improves the accuracy by 1.8% and 1.3% compared with the single original stream (Swin-B) on the AID dataset with two different training ratios. Similarly, on the NWPU dataset with two different training/testing split ratios, the accuracy is improved by 2.1% and 1.5%, respectively. This demonstrates that our auxiliary edge stream with DESOM helps the network achieve better performance.

(2) Attention Map Visualization:

As shown in Figure 12, columns (a) and (b) are the original images and the synthetic edge images of the scene, respectively. Column (c) shows the attention maps of the last layer of original stream, and column (d) shows the attention maps of the last layer of the proposed method TSTNet. We can observe that for the "airplane", whose edge map is synthesized by the differentiable edge Sobel operator module (DESOM), its airplane shapes are highlighted by the Sobel operator because of the large gradient of the object edges. These airplanes thus receive more attention when training a network. In contrast, the pixels on the ground are suppressed by the Sobel operator and receive less attention. This can be reflected in the attention map of airplane, where original stream without the auxiliary edge stream pays little attention to the airplane, and TSTNet with the auxiliary edge stream shifts the attention to all airplanes. Similarly, in the attention map of "ship", Swin-B paid too much attention to the sea surface, while TSTNet paid precise attention to the ship. The underlying reason is that the Sobel operator can help the network to focus on the ship while ignoring the sea surface, whose gradient is small. In the attention maps of scene "island" and scene "storage tank", TSTNet is also able to focus more on the object of the scene.

In summary, the edge stream with DESOM can make it easier for the network to focus on the scene object.

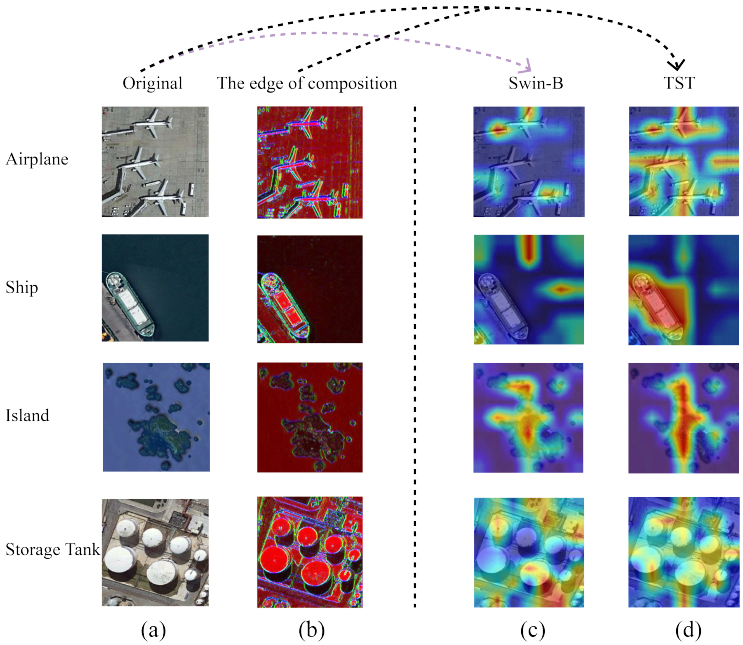


Figure 12. (**a**) The original remote sensing images. (**b**) Synthesized edge images. (**c**) Attention map of the last layer of original stream (Swin-B). (**d**) Attention map of the last layer of TSTNet. The upper dashed line represents that original has only the original input images, and TST has both original input images and synthetic edge input images.

4.4. Experimental Results

Three public benchmark datasets were used to compare our proposed method and other state-of-the-art baselines. The overall classification accuracy (OA) was used as the evaluation metric. In addition, confusion matrices which are generated based on the small training ratio split were used to show how well each category is classified.

(1) AID Dataset: The methods in Table 1 are divided into CNN-based methods and ViT-based methods. It shows that the overall accuracy of our proposed TSTNet is larger than that of the current state-of-the-art methods. From Table 1, we can observe that

- Compared with the Inception-v3-CapsNet which is based on Inception-v3, the overall accuracy is increased by 3.41% and 2.47% using our TSTNet.
- Our TSTNet is 1.7% higher than the CNN-based method KFBNet (DenseNet-121) for the 10% training ratio. In the case of the 20% training ratio, it is 1.05% higher.
- Currently, there are also many vision transformer (ViT)-based remote sensing image classification methods that further improve the upper limit of classification performance. The overall accuracy of our TSTNet is 1.66% higher than the TRS method with 10% training ratio and 0.22% higher with 20% training ratio.

To sum up, we can see that in the case where there is a large amount of training data, our TSTNet is able to obtain better overall accuracy compared with other baselines. Furthermore, in the case where there is only a small number of training data, our TSTNet still outperforms other baselines. This shows the robustness and effectiveness of our TSTNet.

Figure 13 shows the confusion matrix of the classification results when using 20% training ratio. We can find that most of the categories have achieved 100% accuracy, and the lowest accuracy is 92%. The scenes that are poorly classified include “park” and “medium residential”. The scenes “park” are most easily confused with “resort”. This is probably because they are visually similar, as both of them have buildings and small lakes in the scenes, which are even indistinguishable to humans. It is also difficult for the network to distinguish them from each other. Another misclassification scenario is between “medium residential” & “dense residential”. This may be caused by the fact that the boundary between medium and dense is not very clear and there might be some overlaps between them.

Table 1. Classification accuracy of different methods on AID dataset (OA ± STD).

Method Type	Method	20% Train Ratio	50% Train Ratio
△	VGGNet [50]	86.59 ± 0.29	89.64 ± 0.36
	GoogleNet [50]	83.44 ± 0.40	86.39 ± 0.55
	ARCNet-VGG16 [26]	88.75 ± 0.40	93.10 ± 0.55
	EfficientNet-B0-aux [53]	93.96 ± 0.11	–
	EfficientNet-B3-aux [53]	94.19 ± 0.15	–
	GBNet [43]	90.16 ± 0.24	93.72 ± 0.34
	GBNet+global feature [43]	92.20 ± 0.23	95.48 ± 0.12
	Inception-v3-CapsNet [42]	93.79 ± 0.13	96.32 ± 0.12
	ACNet [54]	93.33 ± 0.29	95.38 ± 0.29
	KFBNet(DenseNet-121) [55]	95.50 ± 0.27	97.40 ± 0.10
	EFPN-DSE-TDFF [56]	94.02 ± 0.21	94.50 ± 0.30
	Xu's method [44]	94.74 ± 0.23	97.65 ± 0.25
♢	V16_21K[384 × 384] [28]	95.86 ± 0.28	–
	V16_21K[384 × 384] [pruning 50%] [28]	94.27 ± 1.41	–
	CTNet(ResNet34) [29]	96.35 ± 0.13	97.56 ± 0.20
	CTNet(MobileNet_v2) [29]	96.25 ± 0.10	97.70 ± 0.11
	TRS [30]	95.54 ± 0.18	98.48 ± 0.06
⊙	**TSTNet(ours)**	**97.20 ± 0.22**	**98.70 ± 0.12**

△: CNN-based methods; ♢: ViT-based methods; ⊙: Proposed method.

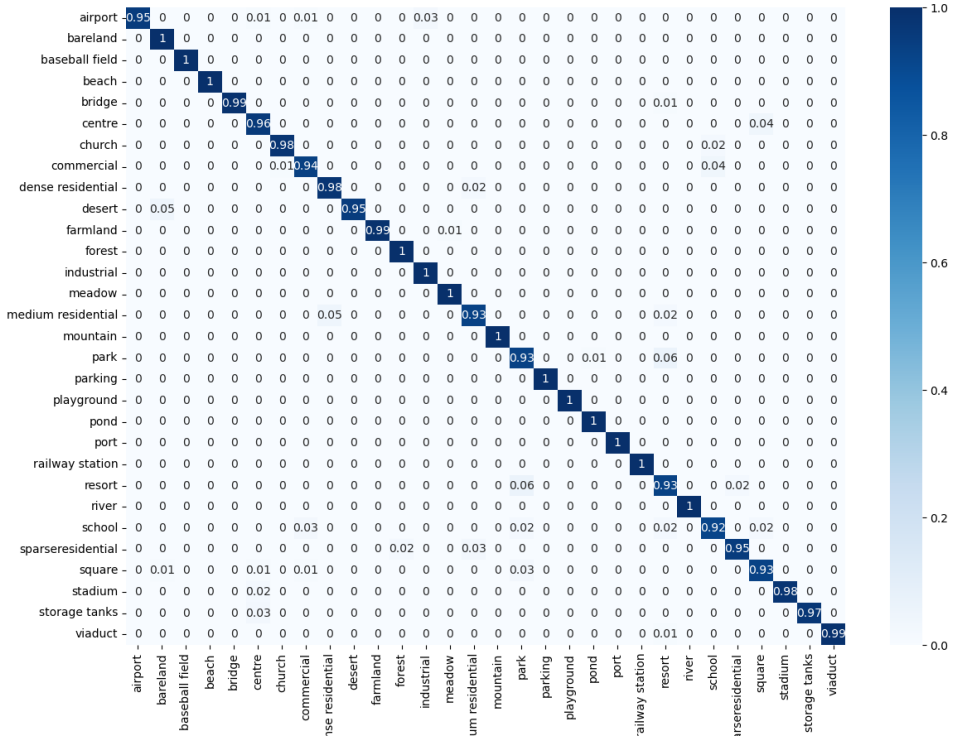


Figure 13. Confusion matrix for AID dataset under 20% training rate, with true labels on the vertical axis and predicted values on the horizontal axis.

(2) NWPU Dataset: The NWPU dataset is more challenging compared with the AID dataset. It has 31,500 images. Current methods do not perform well on NWPU dataset. From Table 2, we can observe that

- The overall accuracy of our TSTNet is 94.08% and 95.70% with the training ratio of 10% and 20%, and it outperforms the state-of-the-art baselines, such as CNN-based method, by a large margin. This is especially true for the small training ratio.
- Under the 10% training ratio, our TSTNet increases the acuracy by 2.99% and 2.17% compared with ACNet and Xu's method. Under the 20% training ratio, 3.28% and 1.27% performance improvement is achieved.
- The performance of our TSTNet also outperforms the ViT-based method. Under the 10% training ratio, TSTNet improves 0.18% and 1.02% compared with CTNet (mobileNet_v2) and TRS. Under the 20% training ratio, the improvement is 0.30% and 0.14%.

Under the 10% training ratio, we plot a confusion matrix to observe the classification errors of each category. From Figure 14, we can see that the overall classification performance is good. The accuracy of most of the categories is larger than 94%. This demonstrates that our network can still maintain good classification performance on the challenging NWPU dataset. Scenes that are easy to be confused include "church" and "palace". This is probably because they have large inter-class similarity as they have similar architectures. How to solve the inter-class similarity problem is our future work and beyond the scope of this paper.

Table 2. Classification accuracy of different methods on NWPU dataset (OA ± STD).

Method Type	Method	10% Train Ratio	20% Train Ratio
△	VGGNet [50]	76.69 ± 0.19	76.85 ± 0.18
	GoogleNet [50]	76.19 ± 0.38	78.48 ± 0.26
	EfficientNet-B0-aux [53]	89.96 ± 0.27	–
	EfficientNet-B3-aux [53]	91.08 ± 0.14	–
	Contourlet CNN [57]	85.93 ± 0.51	89.57 ± 0.45
	ResNeXt-101+MTL [58]	91.91 ± 0.18	94.21 ± 0.15
	VGG_MS2AP [59]	92.27 ± 0.21	93.91 ± 0.15
	Inception-v3-CapsNet [42]	89.03 ± 0.21	92.60 ± 0.11
	KFBNet(DenseNet-121) [55]	93.08 ± 0.14	95.11 ± 0.10
	ACNet [54]	91.09 ± 0.13	92.42 ± 0.16
	Xu's method [44]	91.91 ± 0.15	94.43 ± 0.16
♢	V16_21K[384 × 384] [28]	93.83 ± 0.46	–
	V16_21K[384 × 384] [pruning 50%] [28]	93.05 ± 0.46	–
	CTNet(ResNet34) [29]	93.86 ± 0.22	95.49 ± 0.12
	CTNet(MobileNet_v2) [29]	93.90 ± 0.14	95.40 ± 0.15
	TRS [30]	93.06 ± 0.11	95.56 ± 0.20
⊙	**TSTNet(ours)**	**94.08 ± 0.24**	**95.70 ± 0.10**

△: CNN-based methods; ♢: ViT-based methods; ⊙: Proposed method.

(3) UCM Dataset: UCM is a widely used dataset for remote sensing image classification. The number of images in the dataset is small, so the training set ratio is usually large, and is generally set to 50% and 80%. In addition, the number of categories in the dataset is also very small; this dataset is relatively easy. As a result, most state-of-the-art methods have saturated performance. The experimental results in Table 3 demonstrate that our TSTNet still achieves better performance compared to the state-of-the-art methods.

In Table 3, the overall accuracy of our TSTNet is 98.95% and 99.64% with the training ratios of 50% and 80%, which are 0.34% and 0.67% higher than those of Xu's method. In addition, our TSTNet is 0.29% and 0.12% higher than the TRS method with the training ratios of 50% and 80%. The above shows that the performance of our model is consistent regardless of the dataset size.

The confusion matrix for the UCM dataset under the 50% training ratio is shown in Figure 15. We can observe that it achieves 100% accuracy in 19 out of 21 categories, and only two categories are misclassified.

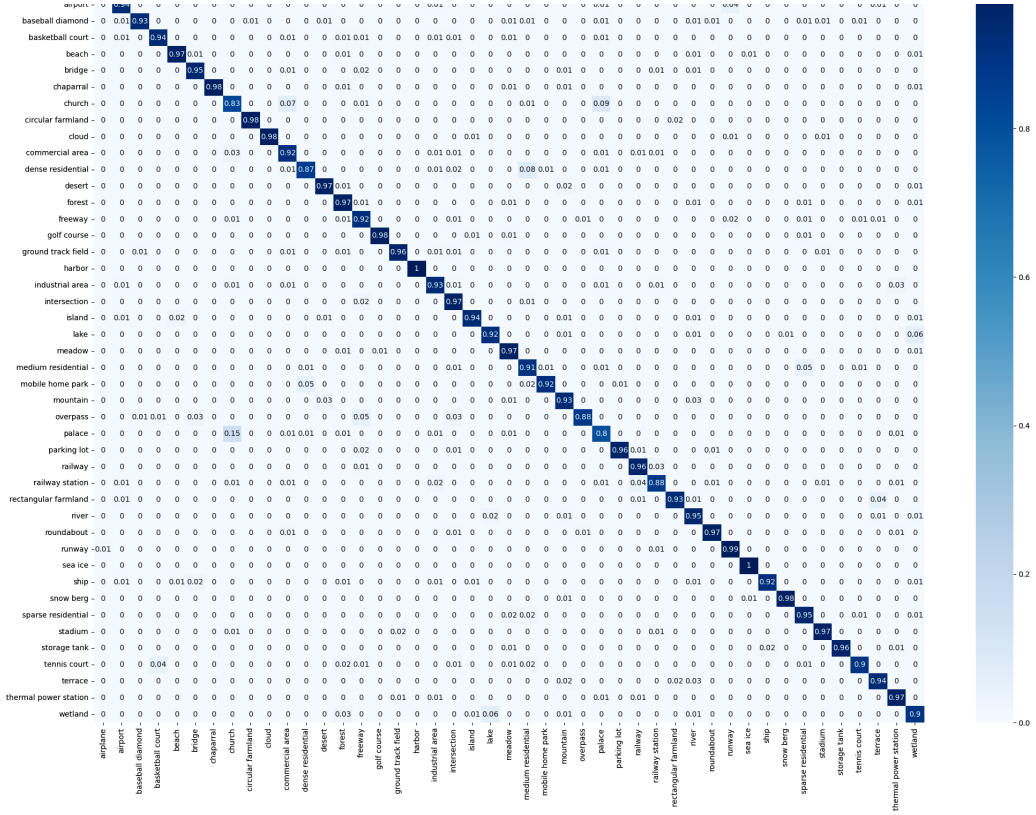


Figure 14. Confusion matrix for NWPU dataset under 10% training rate, with true labels on the vertical axis and predicted values on the horizontal axis.

Table 3. Classification accuracy of different methods on UCM dataset (OA ± STD).

Method Type	Method	50% Train Ratio	80% Train Ratio
△	VGGNet [50]	94.14 ± 0.69	95.21 ± 1.20
	GoogleNet [50]	92.70 ± 0.60	94.31 ± 0.89
	APDCNet [60]	95.01 ± 0.43	97.05 ± 0.43
	SRSCNN [61]	97.88 ± 0.31	98.13 ± 0.33
	EfficientNet-B0-aux [53]	98.01 ± 0.45	–
	EfficientNet-B3-aux [53]	98.22 ± 0.49	–
	Contourlet CNN [57]	–	98.97 ± 0.21
	VGG-16-CapsNet [42]	98.81 ± 0.22	95.33 ± 0.18
	Inception-v3-CapsNet [42]	97.59 ± 0.16	99.05 ± 0.24
	ACNet [54]	–	**99.76 ± 0.10**
	EFPN-DSE-TDFF [56]	96.19 ± 0.13	99.14 ± 0.22
	Xu's method [44]	98.61 ± 0.22	98.97 ± 0.31
♢	V16_21K[384 × 384] [28]	98.49 ± 0.43	–
	V16_21K[384 × 384] [pruning 50%] [28]	97.90 ± 0.10	–
	TRS [30]	98.76 ± 0.13	99.52 ± 0.17
⊙	**TSTNet(ours)**	**98.95 ± 0.24**	99.64 ± 0.21

△: CNN-based methods; ♢: ViT-based methods; ⊙: Proposed method.

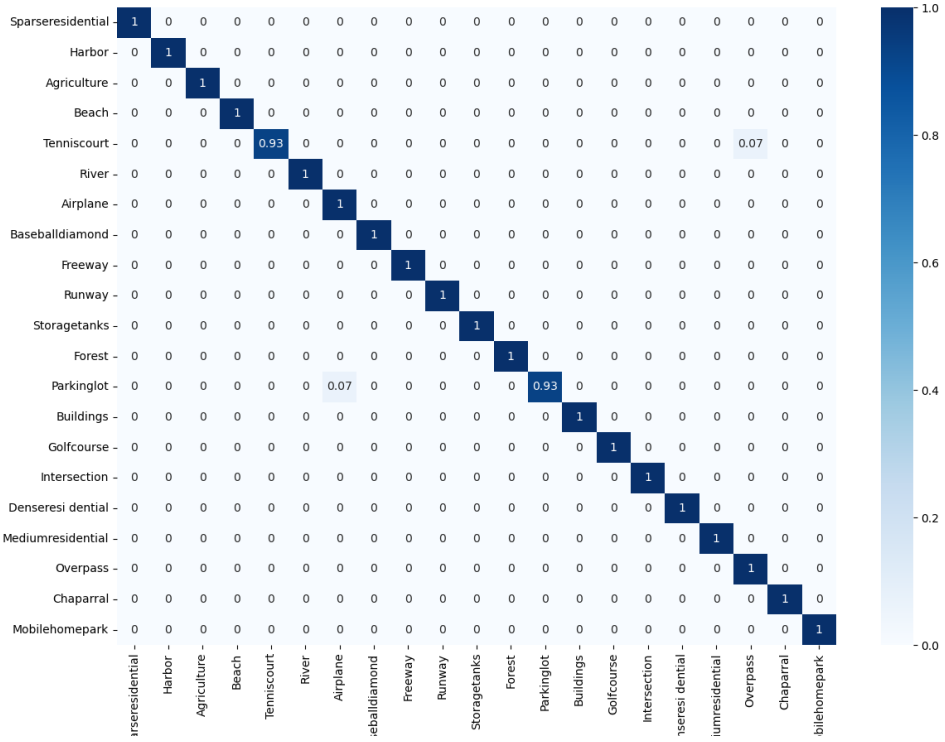


Figure 15. Confusion matrix for UCM dataset under 50% training rate, with true labels on the vertical axis and predicted values on the horizontal axis.

4.5. Running Time and Parameters of the TSTNet

In addition to the accuracy of the model, the running time is also an important indicator of how good the model is. Therefore, we show the parameters and running time of the TSTNet and baseline models in Table 4, where "FLOPs" is the floating point operations, and "throughput" is the number of images processed by the model in one second. "Acc." in the table is the overall classification accuracy of the NWPU dataset with 10% training samples. All experiments were conducted on an NVIDIA RTX 3090 GPU, using a 224 × 224 image size as input. As shown in Table 4, ViT-B and Swin-B have smaller amount of parameters and less running time, but they have lower accuracy. ViT-L and Swin-L have higher accuracy with the increased structural complexity, but the huge number of parameters and floating point operations leads to more running time. Our proposed method, the TSTNet, improves the accuracy substantially with slightly increased parameters. TSTNet has fewer parameters compared to ViT and swin's large version, faster image processing speed, and higher accuracy. For example, TSTNet has 24 M less parameters and 4.3 G less floating point operations than Swin-L with 1% higher accuracy. Thus, TSTNet sacrifices some of its running time for higher accuracy.

Table 4. Inference speed of TSTNet and transformer baseline models.

Method	Parameters	FLOPs	Throughput (Image/s)	Acc. (NWPU 10%)
ViT-B	86 M	16.8 G	540	90.1
Swin-B	87 M	15.4 G	589	92.0
ViT-L	304 M	59.6 G	160	91.3
Swin-L	197 M	34.5 G	270	93.1
TSTNet	173 M	30.2 G	320	94.1

5. Discussion

5.1. Study the Invariance of Transformations

Neural networks, especially convolutional neural networks, usually have certain translation invariance and rotation invariance. Therefore, we tested TSTNet to see if it also has such properties for remote sensing image classification. We selected the two common transformation operations, i.e., rotation and scale.

The experiments are tested on the model after the training is completed, and features α and β are extracted from the original image and the transformed image by TSTNet, and the similarity between feature α and feature β is observed. Cosine similarity is usually used to compare the similarity of two vectors by the cosine of the angle. The closer the cosine similarity of two features is to 1, the more similar they are. Therefore, we can use it to measure the feature similarity of the original and transformed images in order to validate rotation and scale invariance.

The results of the rotation experiment are shown in Table 5; accuracy difference between the original images and the transformed images is also shown in Table 5. In Table 5, regardless of the $\pm 5^\circ$ rotation or the $\pm 15^\circ$ rotation, the cosine similarity between the features extracted after rotation and those extracted without rotation is larger than 0.96. At the same time, the overall classification accuracy of the model is not greatly affected after the rotation. Therefore, the model has some rotation invariance for remote sensing image classification. The results of the scale experiment are shown in Table 6, where we adjust the image scale to $1.1\times$, $1.2\times$, $1.3\times$, and $1.4\times$. The cosine similarities are both over 0.96 within the scale adjustment range of $1.1\times$ and $1.2\times$, indicating that the scale-transformed features remain extremely similar to the untransformed features. Therefore, the proposed TSTNet is proved to have scale invariance. Interestingly, the images which are scaled by $1.1\times$ have a higher accuracy of 0.42%. Probably, it is because the scene is more centered by enlarging and center cropping the image, which improves the recognition of the scene. Of course, very large scale adjustments can cause a decrease in recognition accuracy. For example, a $1.4\times$ scale adjustment will cause a 0.539% decrease in accuracy.

In addition, during the training process, data augmentation, which uses some random transformations of rotation, scale, etc., makes the model learn certain rotation and scale invariance. Therefore, the proposed TSTNet has rotation, scale, etc., invariance.

Table 5. Cosine similarity of features after rotation transformation, and model accuracy.

Transformation	Cosine Imilarity	Accuracy	Accuracy Difference
without rotation	-	95.690	-
rotation $+5^\circ$	0.977	95.875	$+0.185$
rotation $+15^\circ$	0.972	95.315	-0.375
rotation -5°	0.971	95.750	$+0.060$
totation -15°	0.965	95.147	-0.543

Table 6. Cosine similarity of features after scale transformation, and model accuracy.

Transformation	Cosine Similarity	Accuracy	Accuracy Difference
without scale	-	95.690	-
scale $1.1\times$	0.974	96.110	$+0.420$
scale $1.2\times$	0.971	95.857	$+0.167$
scale $1.3\times$	0.965	95.607	-0.083
scale $1.4\times$	0.965	95.151	-0.539

5.2. Selection of Edge Detection Operator

The common edge detection operators are Sobel, Prewitt, and Roberts, with the Sobel operator being the most widely used. We use the Sobel operator in our DESOM and make it learnable. Meanwhile, we also perform experiments by replacing the Sobel operator

in DESOM with other operators. The performance of the Sobel operator compared with the Prewitt operator and Roberts operator is shown in Table 7. Accuracy is the overall classification accuracy of the NWPU dataset with 20% training samples.

The Canny edge detection algorithm contains operations such as Gaussian filtering and non-maximal value suppression, and finally extracts a single edge image. Therefore, it is not suitable to be transformed into a differentiable edge algorithm to be added to DESOM. However, to be fair, we added the original Canny edge detection algorithm as shown in Table 7 for Canny* in the data preprocessing stage. It is fixed and cannot be updated or fine-tuned. For the internal operator of Canny* we choose Prewitt.

In Table 7, it is obvious that the performance of using the Sobel operator is higher than the other operators. Among them, Prewitt is the best performer besides the Sobel operator, reaching 95.483%. The worst result is Roberts, with a 1.013% decrease in accuracy compared to Sobel. Therefore, the Sobel operator and DESOM can work better together to improve the classification accuracy of the network.

Table 7. Accuracy of different edge operators of the NWPU dataset with 20% training samples.

Operator Type	Accuracy (NWPU 20%)
Canny * [62]	95.427
Prewitt	95.483
Roberts	94.678
Sobel	95.691

Canny * indicates that it is the original Canny algorithm and is not differentiable.

6. Conclusions

In this paper, we have proposed a two-stream swin transformer framework (TSTNet) for remote sensing image classification. TSTNet consists of an original stream and an edge stream. By fusing the features from the two streams, we have achieved the state-of-the-art performance. In addition, in order to extract more informative edge information, we have designed a novel differentiable edge Sobel operator module (DESOM) whose parameters can be learned adaptively to better serve the scene classification task. In addition, we have demonstrated the importance of edge stream through ablation study. Experimental results on three publicly available and challenging datasets show that TSTNet has higher classification performance compared with other state-of-the-art methods.

Author Contributions: All the authors made significant contributions to this work. Project administration, S.H.; software, K.Z.; writing—original draft preparation, B.W.; writing—review and editing, Y.Y. and W.W. All authors have read and agreed to the published version of the manuscript

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62171247 and 41921781.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: We would like to thank the editors and reviewers for their careful reading and helpful criticism.

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

References

1. Zheng, X.; Gong, T.; Li, X.; Lu, X. Generalized Scene Classification From Small-Scale Datasets with Multitask Learning. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5609311. [CrossRef]
2. Zheng, X.; Chen, X.; Lu, X.; Sun, B. Unsupervised change detection by cross-resolution difference learning. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5606616. [CrossRef]
3. Zheng, X.; Wang, B.; Du, X.; Lu, X. Mutual attention inception network for remote sensing visual question answering. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5606514. [CrossRef]

4. Ye, Y.; Bruzzone, L.; Shan, J.; Bovolo, F.; Zhu, Q. Fast and robust matching for multimodal remote sensing image registration. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9059–9070. [CrossRef]
5. Zhou, L.; Ye, Y.; Tang, T.; Nan, K.; Qin, Y. Robust Matching for SAR and Optical Images Using Multiscale Convolutional Gradient Features. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 4017605. [CrossRef]
6. Hu, Q.; Wu, W.; Xia, T.; Yu, Q.; Yang, P.; Li, Z.; Song, Q. Exploring the use of Google Earth imagery and object-based methods in land use/cover mapping. *Remote Sens.* **2013**, *5*, 6026–6042. [CrossRef]
7. Gómez-Chova, L.; Tuia, D.; Moser, G.; Camps-Valls, G. Multimodal classification of remote sensing images: A review and future directions. *Proc. IEEE* **2015**, *103*, 1560–1584. [CrossRef]
8. Longbotham, N.; Chaapel, C.; Bleiler, L.; Padwick, C.; Emery, W.J.; Pacifici, F. Very high resolution multiangle urban classification analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 1155–1170. [CrossRef]
9. Tayyebi, A.; Pijanowski, B.C.; Tayyebi, A.H. An urban growth boundary model using neural networks, GIS and radial parameterization: An application to Tehran, Iran. *Landsc. Urban Plan.* **2011**, *100*, 35–44. [CrossRef]
10. Wang, Y.; Zhang, L.; Tong, X.; Zhang, L.; Zhang, Z.; Liu, H.; Xing, X.; Mathiopoulos, P.T. A three-layered graph-based learning approach for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6020–6034. [CrossRef]
11. Yang, Y.; Newsam, S. Geographic image retrieval using local invariant features. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 818–832. [CrossRef]
12. Huang, X.; Wen, D.; Li, J.; Qin, R. Multi-level monitoring of subtle urban changes for the megacities of China using high-resolution multi-view satellite imagery. *Remote Sens. Environ.* **2017**, *196*, 56–75. [CrossRef]
13. Zhang, T.; Huang, X. Monitoring of urban impervious surfaces using time series of high-resolution remote sensing images in rapidly urbanized areas: A case study of Shenzhen. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2692–2708. [CrossRef]
14. Ghazouani, F.; Farah, I.R.; Solaiman, B. A multi-level semantic scene interpretation strategy for change interpretation in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8775–8795. [CrossRef]
15. Li, X.; Shao, G. Object-based urban vegetation mapping with high-resolution aerial photography as a single data source. *Int. J. Remote Sens.* **2013**, *34*, 771–789. [CrossRef]
16. Mishra, N.B.; Crews, K.A. Mapping vegetation morphology types in a dry savanna ecosystem: Integrating hierarchical object-based image analysis with Random Forest. *Int. J. Remote Sens.* **2014**, *35*, 1175–1198. [CrossRef]
17. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
18. Li, Y.; Zhang, Y.; Huang, X.; Yuille, A.L. Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 182–196. [CrossRef]
19. Cheng, G.; Han, J.; Zhou, P.; Xu, D. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. *IEEE Trans. Image Process.* **2018**, *28*, 265–278. [CrossRef]
20. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
21. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-insensitive and context-augmented object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2337–2348. [CrossRef]
22. Cheng, G.; Zhou, P.; Han, J. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2884–2893.
23. Cheng, G.; Han, J.; Guo, L.; Liu, T. Learning coarse-to-fine sparselets for efficient object detection and scene classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1173–1181.
24. Cheng, G.; Ma, C.; Zhou, P.; Yao, X.; Han, J. Scene classification of high resolution remote sensing images using convolutional neural networks. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 767–770.
25. Zhou, W.; Shao, Z.; Cheng, Q. Deep feature representations for high-resolution remote sensing scene classification. In Proceedings of the 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA), Guangzhou, China, 4–6 July 2016; pp. 338–342.
26. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167. [CrossRef]
27. 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 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Bazi, Y.; Bashmal, L.; Rahhal, M.M.A.; Dayil, R.A.; Ajlan, N.A. Vision transformers for remote sensing image classification. *Remote Sens.* **2021**, *13*, 516. [CrossRef]
29. Deng, P.; Xu, K.; Huang, H. When CNNs Meet Vision Transformer: A Joint Framework for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8020305. [CrossRef]
30. Zhang, J.; Zhao, H.; Li, J. TRS: Transformers for Remote Sensing Scene Classification. *Remote Sens.* **2021**, *13*, 4143. [CrossRef]
31. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.

32. Swain, M.J.; Ballard, D.H. Color indexing. *Int. J. Comput. Vis.* **1991**, *7*, 11–32. [CrossRef]
33. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *6*, 610–621. [CrossRef]
34. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
35. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
36. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
41. Fan, R.; Wang, L.; Feng, R.; Zhu, Y. Attention based residual network for high-resolution remote sensing imagery scene classification. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1346–1349.
42. Zhang, W.; Tang, P.; Zhao, L. Remote sensing image scene classification using CNN-CapsNet. *Remote Sens.* **2019**, *11*, 494. [CrossRef]
43. Sun, H.; Li, S.; Zheng, X.; Lu, X. Remote sensing scene classification by gated bidirectional network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 82–96. [CrossRef]
44. Xu, C.; Zhu, G.; Shu, J. A Lightweight and Robust Lie Group-Convolutional Neural Networks Joint Representation for Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5501415. [CrossRef]
45. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, Virtual, 13–15 April 2021; pp. 10347–10357.
46. Li, W.; Cao, D.; Peng, Y.; Yang, C. MSNet: A Multi-Stream Fusion Network for Remote Sensing Spatiotemporal Fusion Based on Transformer and Convolution. *Remote Sens.* **2021**, *13*, 3724. [CrossRef]
47. Xu, Z.; Zhang, W.; Zhang, T.; Yang, Z.; Li, J. Efficient Transformer for Remote Sensing Image Segmentation. *Remote Sens.* **2021**, *13*, 3585. [CrossRef]
48. He, C.; He, B.; Yin, X.; Wang, W.; Liao, M. Relationship prior and adaptive knowledge mimic based compressed deep network for aerial scene classification. *IEEE Access* **2019**, *7*, 137080–137089. [CrossRef]
49. He, C.; Li, S.; Xiong, D.; Fang, P.; Liao, M. Remote sensing image semantic segmentation based on edge information guidance. *Remote Sens.* **2020**, *12*, 1501. [CrossRef]
50. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
51. Cheng, G.; Li, Z.; Yao, X.; Guo, L.; Wei, Z. Remote sensing image scene classification using bag of convolutional features. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1735–1739. [CrossRef]
52. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
53. Bazi, Y.; Al Rahhal, M.M.; Alhichri, H.; Alajlan, N. Simple yet effective fine-tuning of deep CNNs using an auxiliary classification loss for remote sensing scene classification. *Remote Sens.* **2019**, *11*, 2908. [CrossRef]
54. Tang, X.; Ma, Q.; Zhang, X.; Liu, F.; Ma, J.; Jiao, L. Attention consistent network for remote sensing scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2030–2045. [CrossRef]
55. Li, F.; Feng, R.; Han, W.; Wang, L. High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8077–8092. [CrossRef]
56. Wang, X.; Wang, S.; Ning, C.; Zhou, H. Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7918–7932. [CrossRef]
57. Liu, M.; Jiao, L.; Liu, X.; Li, L.; Liu, F.; Yang, S. C-CNN: Contourlet convolutional neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 2636–2649. [CrossRef]
58. Zhao, Z.; Luo, Z.; Li, J.; Chen, C.; Piao, Y. When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework. *Remote Sens.* **2020**, *12*, 3276. [CrossRef]
59. Bi, Q.; Zhang, H.; Qin, K. Multi-scale stacking attention pooling for remote sensing scene classification. *Neurocomputing* **2021**, *436*, 147–161. [CrossRef]
60. Bi, Q.; Qin, K.; Zhang, H.; Xie, J.; Li, Z.; Xu, K. APDC-Net: Attention pooling-based convolutional network for aerial scene classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 1603–1607. [CrossRef]

61. Liu, Y.; Zhong, Y.; Fei, F.; Zhu, Q.; Qin, Q. Scene classification based on a deep random-scale stretched convolutional neural network. *Remote Sens.* **2018**, *10*, 444. [CrossRef]
62. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]