A Deep Learning Model With Capsules Embedded for High-Resolution Image Classification

Yujuan Guo, Jingjuan Liao, and Guozhuang Shen, *Member, IEEE*

***Abstract*—Classification of remote sensing (RS) images is a key technology for extracting information on ground objects using RS methods. Inspired by the success of deep learning (DL) in artificial intelligence, researchers have proposed different algorithms based on DL to improve the performance of classification. At present, a DL model represented by the convolutional neural networks (CNNs) can extract the abstract feature, but it loses the spatial context of the ground objects. To solve the problem of lack of spatial information in CNNs, the Capsule network takes the form of vectors that convey location transformation information. This article proposes using a Capsules–Unet model, which incorporates Capsules within the U-net architecture for classification of RS images. The aim is to train better models by encapsulating the multidimensional features of the objects in the form of Capsules, and to reduce parameter space by improving the dynamic routing algorithm. Experiments are conducted on ISPRS Vaihingen and Potsdam datasets. Capsules–Unet slightly outperforms all other approaches with far fewer parameters, a reduction in parameters of over 81.8% compared with U-net and over 13.8% compared with Capsule network.**

***Index Terms*—Capsules–Unet, classification, deep learning (DL), remote sensing (RS).**

I. Introduction

CLASSIFICATION is a fundamental task in remote sensing (RS), and it is also a complex data processing process [1]. Image classification is similar to semantic segmentation tasks. It refers to the recognition of different objects based on their spectral and shape information, and the assignment of each pixel in the image to its real object category [2]–[4]. The process of RS image classification includes preprocessing, extraction of feature, and classifier design. Early classification was for low resolution (10–30 m) images and pixel-leveled images, mainly comprising unsupervised classification (e.g., ISODATA [5] and K-means [6]) and supervised classification (e.g., neural networks [7] and Random Forest [8]). These methods often use limited spectral information of images and have developed sophisticated commercial software modules, which have been widely applied in environment [9], agriculture [10], land resource [11], and other fields.

High resolution (<2m) RS images have detailed shapes, geometries, texture information, and other spatial features. The precise information can be applied to image analysis and interpretation [12], [13]. However, the appearance of a larger number of details in the images and the complexity of the spectral features make the traditional methods based on the spectral statistical features sensitive to noise, and they also lack semantic meaning of the objects [14]. Aiming at the features of high-resolution images, the object-oriented method uses image segmentation approaches to form image objects from adjacent pixels with similar features. In the classification stage, the object features (color, texture, and geometric features) are calculated as the input of supervised or unsupervised classification.

With the advantages of deep learning (DL) in feature extraction, image classification based on the standard model represented by the Convolutional Neural Networks (CNNs) [15] has developed tremendously in recent years. Compared with the traditional classification method, CNNs are stacks of "convolutional-pooling" layers that can learn extremely complicated hierarchical features from massive data. They do not require hand-crafted features designed by domain-specific knowledge, avoiding the problem that hand-crafted features are highly dependent on domain knowledge. Common CNNs such as the fully convolutional network [16]–[18], U-net [19], [20], generative adversarial networks [21]–[24], and other cross-connected CNNs [25], [26] have become the desired models for various image segmentation tasks, which also show great potential in RS applications. For example, automatic extraction of features from RS data has strongly improved the classification accuracy (almost always greater than 90%) for urban complex objects [27]–[30]. In terms of high-resolution image, some studies have improved the CNN model by multitask learning [31]–[33], multiscale feature aggregation [34]–[37], or multimodel fusion [35], [38], [39]. These improved methods have focused on the problem of objects confusion in the semantic representation of multiclass labeled pixels because the objects are usually adjacent or interleaved. Dolz *et al.* [40] proposes a three-dimensional (3-D) fully convolution neural network, which extended the definition of dense connectivity to multimodal segmentation. Husain *et al.* [41] designed a novel CNN-based global model for large-scale image retrieval, which could learn and aggregate the hierarchical structure of deep

Manuscript received July 11, 2020; revised August 11, 2020 and September 26, 2020; accepted October 13, 2020. Date of publication October 21, 2020; date of current version January 6, 2021. This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA19030302, and in part by the National Science Foundation of China under Grant 4159085. *(Corresponding author: Jingjuan Liao.)*

Yujuan Guo is with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institution, Chinese Academy of Sciences, Beijing 100094, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: guoyujuan_edu@163.com).

Jingjuan Liao and Guozhuang Shen are with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institution, Chinese Academy of Sciences, Beijing 100094, China (e-mail: liaojj@aircas.ac.cn; shengz@radi.ac.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3032672

features from multiple CNN layers. Hang *et al.* [42], [43] achieved good results by improving the CNN model for spectral–spatial classification of hyperspectral images. These works have proved that DL has powerful ability to automatically learn complicated and relevant context features, and it has gained great attention in the field of high-resolution image classification.

Despite the success of DL, there are several challenges for high-resolution RS classification [44], [45]. First, the end-to-end learning strategy makes DL representations a black box [48]. Except for the final network output, it is difficult to understand the logic of CNN predictions hidden inside the network. This makes it difficult to understand the process of classification. Second, unlike natural image datasets, RS images are complex [46], [47]. They may involve various types of objects, which are also different in size, color, location, and rotation. Using spectral properties only may be insufficient to distinguish objects, and discriminative appearance-based features are needed. Therefore, how to combine enriched spectral and spatial information as complementary clues to significantly improve the performance of DL in RS classification is a much-study topic. Third, large and labeled datasets do not always exist in RS application. This is because of the sizes of the datasets involved, and also the conceptual difficulty in labeling datasets. At present, the complex deep networks are too complex to be optimized due to the large number of hyperparameters to be configured. Therefore, it is inevitable that there will be over-fitting when training the DL network.

Recognizing these challenges, many approaches have recently been proposed to address them. A number of researchers have realized that the interpretability of DL model is of great significance in theory and practice, and have designed the model with interpretable representation [49]. The main attempts have been focused on the following aspects: 1) visualization of CNN representation [50], [51]; 2) analysis of CNN performance [52], [53]; and 3) establishment of interpretable model [54], [55]. The present study believes that the interpretable DL model can help people definite the concept of network interpretability and guide the development of interpretable network representation learning. More recently, Sabour *et al.* designed novel neural units, namely "Capsules," to substitute for traditional neural units to construct a Capsules network [56]. Each Capsule outputs an activity vector instead of a scalar. Capsules output a high-dimensional vector to the viewer that describes their properties, such as pose, deformation, and texture. The module length of the vector indicates the probability of an object, with the larger the modal value of the high-dimensional vector, the greater probability of the object's existence [57]. The direction of the vector represents the direction of the object, and the relationship in space, which compensates for the shortcomings of CNNs. The training algorithm of Capsules network involves a routing mechanism among Capsules in successive layers of the network [58]–[62]. The part-whole relationship is integrated into the whole process of DL training in the form of Capsule to enhance the further understanding of complex tasks.

In view of some limitations of Capsules network, some modification ideas have been proposed to improve performance [63]–[67]. These studies have combined the concept of Capsules in the CNN model [63], [66], [67] and used multiscale feature transformation [64], [65] to improve performance in complex data. Different from the original dynamic routing algorithm, the lower level Capsule can choose a single parent to make the network deeper, instead of allowing the lower level Capsule to send its output to all higher level Capsules [68]. This modification prevents the parameters from increasing rapidly with the number of iterations. Different from natural images, DL models in machine vision cannot be directly used in high-resolution RS applications due to the complexity of RS images. At present, many studies use the spatial relationship capture ability of the Capsules network to extract a single category, such as rice image recognition [69], building footprint extraction [70], and vehicle detection [71]. However, few works have been conducted to improve the speed and structure of the Capsules network, and to apply it to more complex data and various tasks.

Therefore, to further understand the classification process and effectively process large-scale RS images, this article attempts to incorporate the concept of Capsules into the U-net model (it can solve the problem of the vanishing gradients in the deep model and can use limited data for training) and propose Capsules–Unet. The contributions include that 1) this approach treats Capsules–Unet as probabilistic graphical models capable ~~to~~ of inferring the probability dependence relationship among objects, and provides a way to design more effective models with Capsules; 2) the reduction in the number of parameters resulting from the modifications of the original dynamic routing make it possible to rapidly process large-scale RS images; and 3) Capsules–Unet is first applied to high-resolution RS multiclassification tasks. Experiments are conducted on ISPRS Vaihingen and Potsdam datasets [72], and the performance of Capsule network and U-net with regard to accuracy and efficiency are compared.

II. Methods

A. Datasets

The effectiveness of the model was evaluated using the Vaihingen and Potsdam datasets from ISPRS 2D Semantic Labeling Challenge [72].

The Vaihingen dataset consists of 33 images, and each image comprises true orthophoto (TOP) tiles and ground truth labels with a spatial resolution of 9 cm. There are three bands in the TOP tiles: near-infrared, red, and green channels. Each TOP tile comes with ground truth labels from the following set: impervious surfaces, buildings, trees, low vegetation, cars, and clutter. Out of 33 TOP tiles, 31 were used to train the model, while the other 2 were used for test.

The Potsdam dataset contains 38 images with 6000×6000 pixels each, which comprises TOP and digital surface models (DSMs). The DSM is an array with the same size as the input imagery and provides an elevation value at each pixel. Each TOP tile consists of red, green, blue, and infrared bands. The spatial resolution of TOP and DSM is 5 cm. In the experiment, to increase the number of bands in the Potsdam dataset, the normalized difference vegetation index (NDVI) was calculated by using red and infrared bands. Out of 38 images, 24 were

TABLE I
PARTITION OF LABELED DATA INTO TRAINING AND TESTING TESTS

	Vaihingen dataset	Potsdam dataset
Training set and validation set	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 33, 34, 35, 38	2_10, 2_11, 2_12, 3_10, 3_11, 3_12, 4_10, 4_11, 4_12, 5_11, 5_12, 6_7, 6_9, 6_10, 6_11, 6_12, 7_7, 7_8, 7_9, 7_10, 7_11, 7_12
Testing set	32, 37	5_10, 6_8

provided with ground truth and only the 22 TOP tiles with red, green, blue, infrared, DSM, and NDVI bands were used for training the classification model. The remaining two TOP patches were used for testing. The Potsdam dataset had the same ground objects as the Vaihingen dataset. The images used for Vaihingen and Potsdam are listed in Table I.

In the experiment, training and validation of the model were based on the training set results. The test set was not used for training, but as a true value to evaluate the experimental results.

B. Network Architecture

1) Capsules: As mentioned previously, Sabour *et al.* [56] designed new neural units, namely Capsules, to substitute for traditional neural units to construct a Capsules network. A Capsule is a group of neurons that depict properties of various entities present in an image. The output of each Capsule is an activity vector the length and orientation of which give the likelihood of the object and its instantiation parameters. In this sense, each Capsule is in charge of finding some specific object in an input image, instead of calculating a feature map (as in traditional CNNs). The Capsules network aims to overcome the drawbacks of CNNs, especially the inability of recognizing the pose information of an entity and the absence of part-whole relationships among simpler objects.

Each Capsule has two ingredients: weights W_{ij} and coupling coefficients c_{ij}. u_i is an output of a Capsule i, and j is the parent Capsule, the prediction $\tilde{u}i|j$ is calculated as

$$\tilde{u}i|j = Wij \cdot ui \tag{1}$$

where W_{ij} is a weighting matrix that maps the spatial relationship between part and whole, such as attitude (position, size, direction), deformation, texture and so on. Capsules use dynamic routing where the output is sent to all the final Capsules. The coupling coefficient c_{ij}s are assignment probabilities between a pair of part-whole Capsules. The assignment probability between Capsules is shown in Fig. 1. It was then compared with the actual output of parent Capsule. If the outputs matched, the coupling coefficient c_{ij} between the two Capsules was increased.

2) Locally Constrained Dynamic Routing: The original dynamic routing algorithm required a lot of space to store the relevant parameters. As a result, algorithms can be extremely expensive in terms of memory and computational efficiency. This article has improved the dynamic routing algorithm from two aspects to solve the problem of algorithm parameters and running efficiency. First, Lth layer Capsules were routed only to L+1th layer within a defined spatially-local kernel. Second, transformation matrices were shared for each member of the grid within a Capsule type but were not shared across Capsule types.

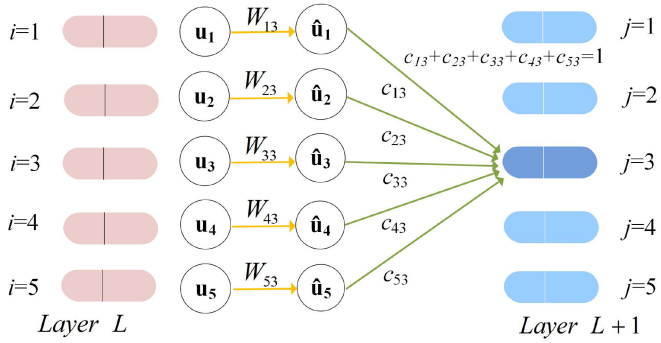

Fig. 1. Assignment probability between Capsules and its standardization requirements.

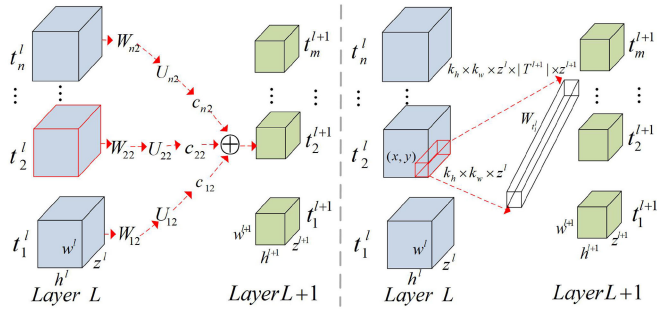

Fig. 2. Comparison of the original dynamic routing (left) and the modified dynamic routing (right).

As illustrated in Fig. 2, In the L+1 layer of Capsules, each type of Capsule receives a set of prediction vectors. This group of prediction vectors was $\{\tilde{u}_{xy|t_1^l}, \tilde{u}_{xy|t_2^l}, \ldots, \tilde{u}_{xy|t_n^l}\}$, which is the matrix multiplication of transformation matrix $W_{t_i^l}$ and L layer Capsules output $U_{xy|t_i^l}$, that is, for all t_i^l, $\tilde{u}_{xy|t_i^l} = W_{t_i^l} \times U_{xy|t_i^l}$. The original dynamic routing algorithm would have routed all Capsules in layer L to all the Capsules in layer $L+1$. However, the feature maps resulting from the convolution operation have localized features, thus, adjacent Capsules have similar information. To reduce the number of parameters, the Capsules output $U_{xy|t_i^l}$ of L layer is defined in the window with (x, y) as the center and the size of $k_h \times k_w$, that is, the size of each $U_{xy|t_i^l}$ is $k_h \times k_w \times z^l$. The transformation matrix $W_{t_i^l}$ was not related to the position (x, y), but was shared for each member of the grid within a Capsule type. The process was similar to a defined window scanning the input characteristic map in turn. Therefore, the size of $W_{t_i^l}$ was $k_h \times k_w \times z^l \times |T^{l+1}| \times z^{l+1}$, where $|T^{l+1}|$ is the number of Capsule types in the L+1 layer. Instead of routing each Capsule in the Lth layer individually, redundancy could be eliminated by routing a block of Capsules from the Lth layer to the L+1th layer.

The parameter update process for locally constrained dynamic routing is shown in Fig. 3. The input vector to parent Capsule p_{xy} is calculated as

$$p_{xy} = \sum_n c_{t_i^l|xy} \cdot \tilde{u}_{xy|t_i^l} \tag{2}$$

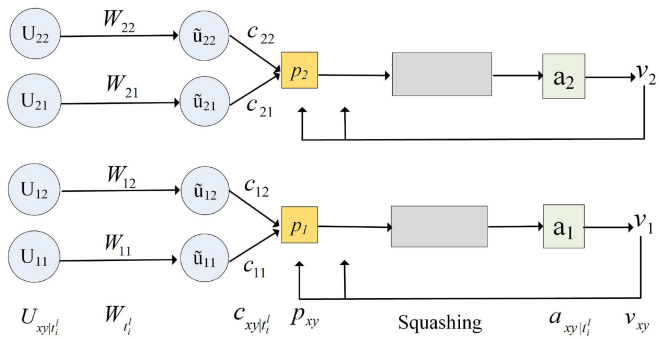

Fig. 3. Process of dynamic routing.

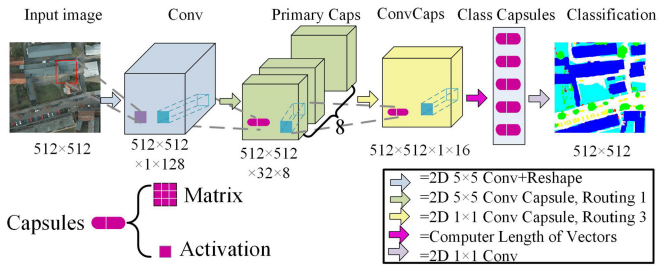

Fig. 4. Simple three-layer Capsule network.

$$c_{t_i^l|xy} = \frac{\exp(b_{t_i^l|xy})}{\sum_k b_{t_i^l|xy}} \text{ with } \sum c_{t_i^l|xy} = 1. \tag{3}$$

Here, the coupling coefficient $c_{t_i^l|xy}$ is determined by the local constraint dynamic routing algorithm and $b_{t_i^l|xy}$ is log of probability Capsule i being coupled with Capsule j. The iterative updating method of $b_{t_i^l|xy}$ is as follows:

$$b_{t_i^l|xy} \leftarrow b_{t_i^l|xy} + \tilde{u}_{xy|t_i^l} \cdot v_{xy}. \tag{4}$$

In Capsules–Unet, the input was a vector in the previous layer. Thus, a nonlinear squashing function was used to restrict the vector length to 1. The expression is shown in formula (5)

$$v_{xy} = \frac{||p_{xy}||^2}{1 + ||p_{xy}||^2} \cdot \frac{p_{xy}}{||p_{xy}||} \tag{5}$$

where p_{xy} was the input to Capsule j and v_{xy} was the output at the spatial location (x, y). Last, the agreement was measured as the scalar product $a_{t_i^l|xy} = v_{xy} \cdot \tilde{u}_{xy|t_i^l}$ and updated to $b_{t_i^l|xy}$.

3) Capsules Network: Sabour *et al.* showed that a simple three-layer Capsule network demonstrated remarkable initial results, producing good classification results for the MNIST and CIFAR10 datasets [54]. The Capsule network is shown in Fig. 4. It included four types of Capsule layers, which could also be used as building blocks in Capsules–Unet. Therefore, we briefly describe them as follows:

1) Convolution layer was used mainly to extract low-level features from the input image. It contains 16 filters with the size of 5 × 5 at the stride of 1, the input of which was 512 × 512 × 3 and the output was 512 × 512×1×128 tensor.
2) PrimaryCaps was the first Capsules layer, where features from previous convolution layer were processed and transitioned into Capsules via convolution filtering.
3) ConvCap layers functioned similarly to CNNs convolutional layers in many aspects. However, they took Capsules as inputs and utilized the routing algorithm to infer outputs, which were also Capsules.
4) Class Capsule layer L was a degenerated layer with one Capsule for each predefined class label $C_k \in C = \{C_1, C_2, \ldots, C_k, \ldots\}$. Each Capsule in the previous layer was fully connected to the Capsules in this layer.

The Capsule network was designed for classification of RS images. The output number of the last layer is equal to the class number of our task this article.

4) Proposed Capsules–Unet Architecture: This article proposed a classification algorithm for high-resolution images based on the Capsules network and U-net. The aim is to design a classification solution using the concept and structure of the Capsules network, with a view to improving classification performance through the viewpoint invariance and network interoperability mechanism.

The general structure of our Capsules–Unet, which consists of three components, is shown in Fig. 5. The input of Capsules–Unet can be an image of any size; here, it was a 512 × 512 pixel multibands image. Capsules–Unet started with a feature extraction module. It contained 16 filters with a size of 5 × 5 at the stride of 1 to the input image. Through a 2-D convolution layer, the image output 16 feature maps with the same spatial dimension. Then, though two Capsule filters with the size of 5 × 5 at the stride of 2 to the feature maps, the output of the feature extraction module formed a 256 × 256 × 2 Capsule type, where each Capsule was a 16-D vector. The feature extraction module captured the discriminative features of the input data to be fed into the later modules. U-net has a deep encoder–decoder structure, its architecture consists of a contracting path and an expansive path. The lineup of the Capsule layers is shown as green and orange arrows in Fig. 5. The contracting path consists of four consecutive PrimaryCaps layers. By incorporating Capsule layers instead of convolutional layers in the U-net, the correlation among different objects could be extracted. The extension path consists of a series of PrimaryCaps and DeconCaps layers alternately. To compensate for the loss of global connectivity caused by locally constrained routing, DeconCaps was used for transposing operations, as well as parameter updates through local constraint routing. Skip connections, shown as blue arrows in Fig. 5, are connections that pass over one or more layers. Through the skip connections, feature maps from the contracting path were cropped and copied for the correspondingly upsamplings in the expansive path. Finally, the end of the network follows a class Capsule layer for the pixel-wise prediction.

C. Network Training

To prove the efficiency of the Capsules–Unet model, the Capsule network was compared with U-net using the same dataset and in the same experimental environment. The general procedure of the training stage is shown in Fig. 6. Random crop

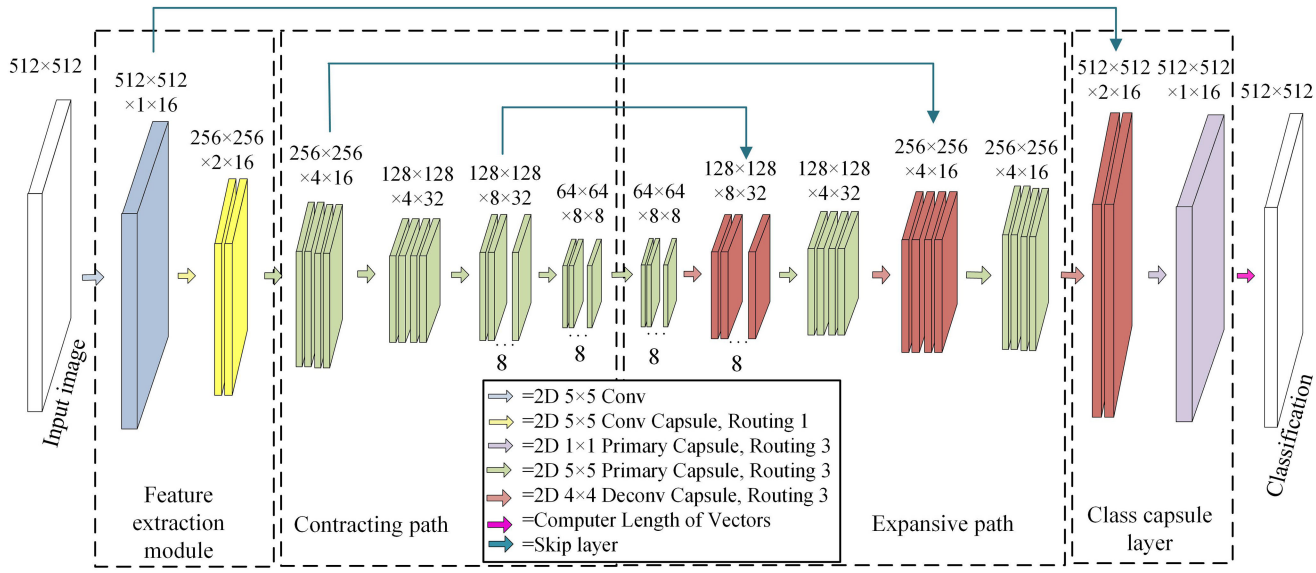

Fig. 5. Architecture of our Capsules–Unet.

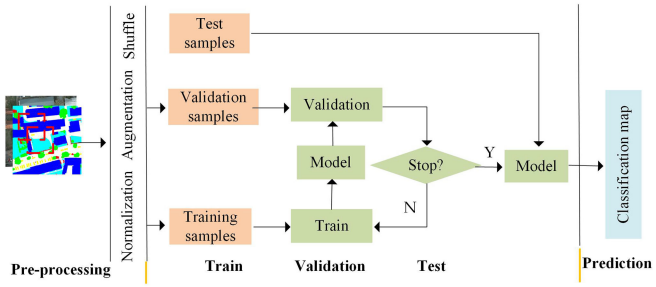

Fig. 6. Procedure of network training.

is an effective strategy in data augmentation. To better capture the spatial information among objects, the image was randomly crop to 512 × 512 pixels from the patches during training. To alleviate the prediction bias, all the datasets were first shuffled and then a random order of the images used. In addition, the proposed model used pixel-level samples instead of patch-level samples, which could train more complex models with limited data. The maximum number of training epochs was set to 10 000. Due to the memory limit of GPUs, the batch size of each training step was 30, which meant that each time, 30 samples were input to fit the model. The learning rate was 0.01. The Adam optimization [73] was adopted to update all the parameters of our network.

Our model was trained on a PC with a 3.7-GHz 8-core CPUs and 32-GB memory. An NVIDIA Quadro P600 GPUs was used for acceleration.

D. Evaluation Metrics

The evaluation strategy from 2-D Semantic Labeling Contest relies on the pixel-based confusion matrix. Overall Accuracy (OA) and Kappa coefficient [74] are computed based on the confusion matrix. OA is the percentage of the correctly classified images among all the testing set. The Kappa coefficient is another widely used evaluation standard, which is based on the confusion matrix to assess the precision of RS classification.

III. Results and Discussion

A. Comparison of Different Proportion for Training Data

It is known that depending on the split of the data, the performances of the different methods may vary as simpler or more difficult examples are involved in the training or test set. With the objective of understanding the robustness of the current method with respect to this phenomenon, three different random 70%–90% splits of the dataset were built.

For the Vaihingen dataset, the number of samples in the car and clutter classes was 1.21% and 0.76%, respectively. The percentage of samples in the impervious, building, low vegetation, and tree classes in the Vaihingen dataset were about 25%, and the proportion of samples was almost same. In the Potsdam dataset, the distribution proportion of samples was still unbalanced. The number of samples in the car and clutter classes were less than 5%, and the proportion of impervious surfaces, buildings, low vegetation, and trees classes were 26.69%, 24.95%, 28.67%, and 13.60%, respectively. These underrepresented classes should make these two datasets more challenging for training.

Fig. 7 shows the classification results versus the different proportions of training samples provided by different models, i.e., Capsules–Unet, Capsule network, and U-net for the Vaihingen and Potsdam datasets. The vertical axis and the horizontal axis in the Fig. 7 denote the accuracy and the different proportion for training, respectively. The general trend shows that classification accuracy increases as the percentage of training samples involved for all three models. In the Vaihingen dataset, the Capsules–Unet and Capsule network did not perform well when the number of samples was limited. For a smaller training

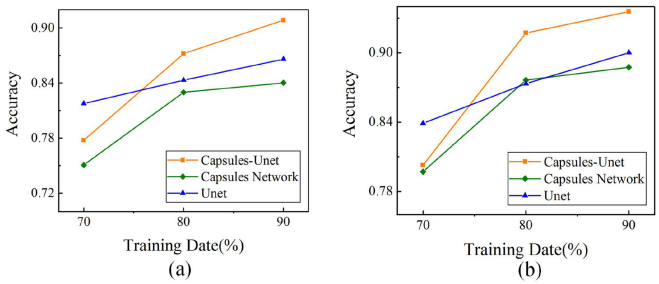

Fig. 7. Classification accuracy of samples with different proportions, i.e., 70%, 80%, and 90% of the (a) Vaihingen dateset and (b) Potsdam dataset is used as training data.

number (<80%), the OA for Unet was 81%, which was substantially higher than the values obtained from the Capsules–Unet (77%) and the Capsule network (75%). However, for the Capsules–Unet and Capsule network models, the accuracies and the corresponding stability increased rapidly with the higher quantity of training samples in the learning phase. For U-net, the overall accuracies achieved at 70% and 90% training samples differed by approximately 4%. This suggested that U-net is less sensitive to the training sample number. The performance of the three models in the Potsdam dataset was similar to that in the Vaihingen, but the OA was generally better than that for Vaihingen. Clearly, the characteristics of the datasets are critical for the accuracy statistics. There are six bands in the Potsdam dataset: red, green, blue, infrared, DSM, and NDVI channels. The information from additional bands is beneficial to feature extraction of the model, and to a certain extent, it makes up for the shortage of sample number. To ensure the stability and excellent performance of the three networks, 80% of the training samples were used for the next experiment in the following comparative.

B. Comparison and Evaluation of Classification

The results of three models for the Vaihingen and the Potsdam datasets were next compared from the qualitative and quantitative perspectives.

Two examples of evaluation results in the Vaihingen dataset were selected (Fig. 8). The ground truth labels are shown in column (a) and results from Capsules–Unet, Capsule network, and U-net are displayed in column (b)–(d), respectively. For impervious surfaces and low vegetation, the present method shows an improvement compared with other methods. However, due to the spectral similarity, parts of trees are mistaken as low vegetation in the prediction of Capsules–Unet. Buildings were classified with high accuracy in Capsules–Unet. The predictions of roofs with a complex structure using U-net were not accurate. All the methods displayed a poor performance for cars, which were misclassified as building due to the limited training samples and similarities in color. The clutter class was hard to identify in this dataset, because limited training examples were available for most objects in the class.

Table II presents the evaluation results of the Vaihingen test images, which shows that the proposed method is better than the Capsule network and U-net in the OA. The indicators show

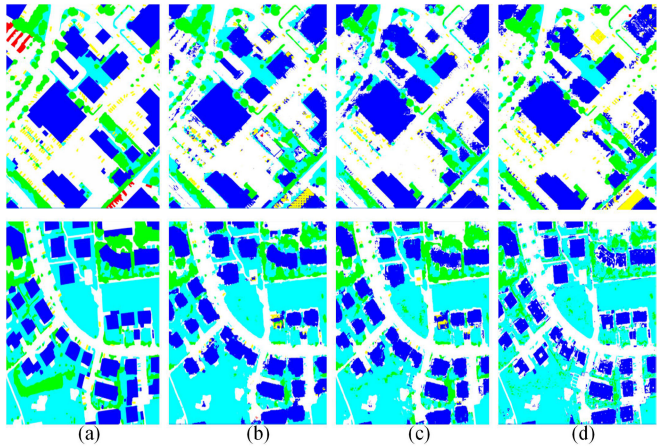

Fig. 8. Example results of test images in the Vaihingen dataset. (a) Ground truth label. (b) Capsules–Unet results. (c) Capsule network results. (d) U-net results. White: impervious surfaces, blue: buildings, cyan: low vegetation, green: trees, yellow: cars, red: clutter.

TABLE II
EVALUATION RESULTS ON VAIHINGEN AND POTSDAM DATASETS

Classes	Vaihingen data set			Potsdam data set		
	Capsule s-Unet	Capsule network	U-net	Capsules -Unet	Capsule network	U-net
Impervious	90.28	87.15	88.19	94.23	88.89	91.70
Building	87.41	85.06	83.85	96.54	93.98	90.03
Low vegetation	89.82	82.01	84.92	87.99	86.43	85.61
Tree	85.01	83.93	83.65	90.97	87.79	83.48
Car	34.80	28.57	32.99	57.76	47.32	53.39
Clutter	0	0	2.06	6.12	7.46	8.12
Overall Accuracy	87.21	82.99	84.32	91.72	87.64	87.33
Kappa	0.81	0.77	0.78	0.83	0.81	0.80

TABLE III
NUMBER OF LAYERS AND PARAMETERS OF THE THREE NETWORKS

Networks	Number of layers	Number of parameters	Running time	(***/Unet)×100 %
			Vaihingen/ Potsdam	Vaihingen/ Potsdam
Capsules-Unet	14	~5.6M	16.0 h / 25.5 h	128.6% / 150.0%
Capsule network	3	~6.5M	13.8 h / 22.8 h	110.4% / 134.1%
U-net	19	~30.8M	12.5 h / 17.0 h	100.0% / 100.0%

that Capsules-Unet's OA was 4.22% higher than that of the Capsule network and 2.89% higher than that of the U-net. The Kappa coefficient were 0.81, 0.77, and 0.78, respectively, for Capsules–Unet, Capsule network, and U-net. The classification accuracies of Capsules–Unet in the impervious surfaces and tree classes were 90.28% and 85.01%, respectively. These were much higher than the accuracies of the Capsule network and U-net. For the clutter objects, all the methods displayed poor performance. Although the spatial features of the target were extracted in the form of "Capsules," the spatial information was often limited due to the single-scale input and ambiguity in the ground truth.

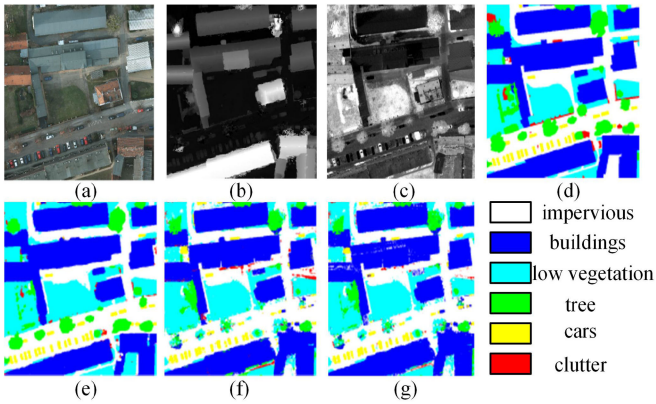

Fig. 9. Classification results of test images in the Potsdam dataset. (a) Original images. (b) DSM. (c)NDVI. (d) Ground truth label. (e) Capsules–Unet results. (f) Capsule network results. (g) U-net results.

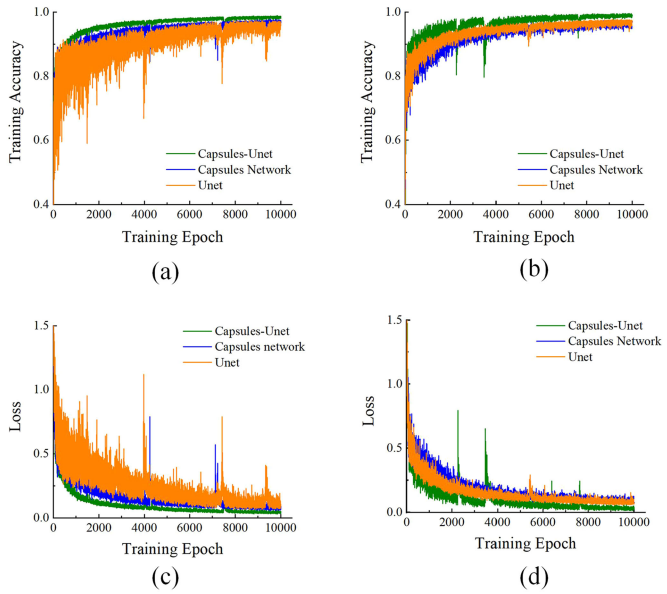

Fig. 10. Training accuracy of (a) Vaihingen dataset and (b) Potsdam dataset. Loss of (c) Vaihingen dataset and (d) Potsdam dataset.

Like the Vaihingen dataset, predictions of test images in the Potsdam dataset are displayed to evaluate the proposed model. To visualize the evaluation results, an example of evaluation results for the Potsdam dataset is selected (Fig. 9). With respect to the impervious surfaces and clutter classes, the present method could predict clutter near the buildings in Fig. 9. However, the other approaches falsely labeled a part of impervious surfaces class as clutter or buildings/impervious surfaces. The Capsules–Unet model distinguished low vegetation in shade from trees, while the other methods misclassified a part of trees as low vegetation. This may be because Capsules–Unet learns the characteristics of ground objects themselves, and also learns the part-whole relationship of the ground objects, through the use of Capsules. For cars, the three methods showed a poor performance (<60%) in a complex situation compare with the Potsdam dataset. The advantage of Capsule method is the spatial context information, but this information may be not significance for small objects.

Table II lists the accuracy of classification results for the Potsdam dataset, and shows that Capsules–Unet can effectively increase the classification accuracy and decrease the confusion condition among different ground components. The buildings detection rate of Capsules–Unet reached 96.54%, which was a gain of 2.56 percentage points over the Capsule network and 6.51 percentage points over U-net. The classification accuracy of Capsules–Unet in trees was 90.97%, which was much higher than the accuracy of the other models. As trees are typically adjacent to impervious surfaces and low vegetation, meaningful part-to-whole relationships were considered in classifying these objects, which is not done with U-net. In addition, compared with Vaihingen dataset, the accuracy of the three models for the car and cluster classes was significantly improved. By integrating the Capsules, the proposed method could effectively extract high-level semantic features and obtain highly accurate classification results when there were sufficient training data.

C. Comparison of Model Parameters

The original Capsule Network used the concept of vector Capsules and dynamic routing to combine key features from the convolution to produce a more robust model. But it also comes at a high cost of computing, as a large number of "invalid" Capsules are involved in the calculation of dynamic routing. For example, there were 32 types of Capsules in layer L, each of which had a size of $6 \times 6 \times 8$, and was routed to $10 \times 1 \times 10$ Capsules in layer $L+1$. The number of parameters is $(6 \times 6 \times 32) \times 10 \times 16 \times 8 = 1474560$. Coincidentally, this layer contained about the same number of parameters as the Capsules–Unet with locally constrained dynamic routing, which operates under the input of 512×512 pixels.

The final quantitative results of these experiments are shown in Table III. The number of network layers of Capsules–Unet, Capsule network, and U-net were 14, 3, and 19, respectively, but the total number of trainable parameters was 1 419 472, 1 680 480, and 7 858 598. Capsules–Unet slightly outperforms all other compared approaches with far fewer parameters: a reduction in parameters of over 81.8% from U-net and over 13.8% compared with Capsule network. By improving the original dynamic routing algorithm, Capsules–Unet ensures the depth of the model, and also greatly reduces the number of parameters, so that the model could train a deeper model with limited samples.

To demonstrate the complexity of Capsules–Unet compared with the Capsule net and U-net, Table III shows the total time consumption of the models of 10 000 epochs trained on two datasets. Capsules–Unet consumed 30%–50% more time in the Potsdam dataset than the U-net model. The relative time consumption was smaller when the model was trained on the Vaihingen dataset. This was expected because there were fewer Vaihingen datasets than the Potsdam datasets.

Fig. 10 displays the evolution of the training accuracy of the three models per epoch and computational loss on the Vaihingen and Potsdam datasets. Capsules–Unet required only a reduced number of epochs and a very short time to reach almost optimal performance in two kinds of datasets, which highlights the remarkably fast convergence of the Capsules–Unet architecture.

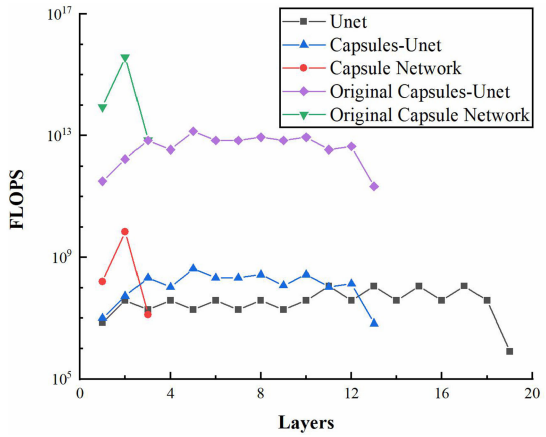

Fig. 11. FLOPS of each layer in the model.

FLOPS (Floating-point Operations Per second) [75] is an important index to evaluate the calculation and complexity of a model. Fig. 11 shows the FLOPS of each layer of the three models. The FLOPS values in Capsules–Unet and the Capsule network were much smaller than those of the model using the original dynamic routing, thus leading to more compact models. This makes it possible for the Capsules–Unet to be applied in large-scale areas.

IV. Conclusion

This article has proposed a Capsules–Unet that incorporates Capsules within the U-net architecture for classification of high resolution RS images. The approach treats Capsules–Unet as probabilistic graphical models that can infer the probability dependence relationship among objects, through which part-whole relationships can be explicitly constructed. The experiments have proved the effectiveness of the proposed method for the classification using two common datasets. In addition, modifications of original dynamic routing have allowed the use of large-scale images due to the reduction of parameters in two key ways. These improvements have allowed the model to operate on large image sizes, whereas previous Capsule networks have been restricted to very small inputs. Compared with the Capsule network and U-net, the accuracy of Capsules–Unet on the Vaihingen and Potsdam datasets was slightly improved. In particular, the proposed Capsules–Unet architecture has 13.8% less parameters than the Capsule network and 81.8% less than U-net. The proposed algorithm has fundamentally improved the current state-of-the-art classification approaches, and has proven that Capsules can successfully simulate the spatial relationships of the objects better than traditional CNNs. In the future, more effort should be devoted to investigating the difference in the mechanism between the Capsules and other CNNs.

Acknowledgment

The authors would like to thank ISPRS 2D Semantic Labeling Challenge for providing the datesets, and the anonymous reviewers for their voluntary and the constructive comments that helped to improve this article.

References

[1] L. Mou and X. X. Zhu, "Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018.

[2] P. Le-Hong and A. Le, "A comparative study of neural network models for sentence classification," in *Proc. 18th NAFOSTED Conf. Inf. Comput. Sci.*, 2018, pp. 360–365.

[3] W. Han, R. Feng, L. Wang, and L. Gao, "Adaptive spatial-scale-aware deep convolutional neural network for high-resolution remote sensing imagery scene classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 4736–4739.

[4] J. Zhao, Y. Zhong, H. Shu, and L. Zhang, "High-resolution image classification integrating spectral-spatial-location cues by conditional random fields," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4033–4045, Sep. 2016.

[5] S. Hemalatha, S. Anouncia, and Margret, "Unsupervised segmentation of remote sensing images using FD based texture analysis model and ISODATA," *Int. J. Ambient Comput. Intell.*, vol. 8, no. 3, pp. 58–75, 2017.

[6] A. Yaseen, W. Laftah, Z. Ali Othman, and M. Zakree Ahmad Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Syst. Appl.*, vol. 6, no. 7, pp. 296–303, 2017.

[7] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *J. Mach. Learn. Res.*, vol. 15, pp. 315–323, 2011.

[8] O. Tokar, O. Vovk, L. Kolyasa, S. Havryliuk, and M. Korol, "Using the random forest classification for land cover interpretation of landsat images in the Prykarpattya region of Ukraine," in *Proc. IEEE 13th Int. Sci. Tech. Conf. Comput. Sci. Inf. Tech.*, 2018, pp. 241–244.

[9] X. Wang *et al.*, "Augmented reality in built environment: Classification and implications for future research," *Automat. Construct.*, vol. 32, pp. 1–13, 2013.

[10] M. Sharif *et al.*, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Elect. Agriculture*, vol. 150, pp. 220–234, 2018.

[11] K. E. Sawaya *et al.*, "Extending satellite remote sensing to local scales: Land and water resource monitoring using high-resolution imagery," *Remote Sens. Environ.*, vol. 88, no. 1–2, pp. 144–156, 2003.

[12] P. Zhang, M. Gong, L. Su, J. Liu, and Z. Li, "Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 24–41, 2016.

[13] F. Mirzapour and H. Ghassemian, "Improving hyperspectral image classification by combining spectral, texture, and shape features," *Int. J. Remote Sens.*, vol. 36, pp. 1070–1096, 2015.

[14] L. Zhilei and Y. Luming, "The object classification algorithm and application for hyperspectral imagery based on BDT-SVM," *Remote Sens. Tech. Appl.*, vol. 31, no. 1, pp. 177–185, 2016.

[15] A. Krizhevsky, I. Sutskever, and G. E Hinton, "ImageNet classifification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, Dec. 2012, pp. 3–8.

[16] C. Peng, Y. Li, L. Jiao, Y. Chen, and R. Shang, "Densely based multi-scale and multi-modal fully convolutional networks for high-resolution remote-sensing image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2612–2626, Aug. 2019.

[17] H. Lin, Z. Shi, and Z. Zou, "Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1665–1669, Oct. 2017.

[18] Y. Wang, C. He, X. Liu, and M. A. Liao, "Hierarchical fully convolutional network integrated with sparse and low-rank subspace representations for PolSAR imagery classification," *Remote Sens.*, vol. 10, no. 2, 2018, Art. no. 342.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assisted Interv.*, Oct. 2015, pp. 234–241.

[20] R. Li *et al.*, "DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 3954–3962, Nov. 2018.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, and B. Xu, "Generative adversarial nets," In *Proc. Int. Conf. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2672–2680.

[22] M. Gong, X. Niu, P. Zhang, and Z. Li, "Generative adversarial networks for change detection in multispectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2310–2314, Dec. 2017.

[23] B. Benjdira, Y. Bazi, A. Koubaa, and K. Ouni, "Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images," *Remote Sens.*, vol. 11, pp. 1–23, 2019.

[24] S. A. Israel *et al.*, "Generative adversarial networks for classification," *IEEE Appl. Imag. Pattern Recognit. Workshops*, 2017, pp. 1–4.

[25] Y. Que and H. J. Lee, "Densely connected convolutional networks for multi-exposure fusion," in *Proc. Int. Conf. Comput. Sci. Comput. Intell.*, 2018, pp. 417-–420.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 770–778.

[27] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 155–165, 2016.

[28] S. De, L. Bruzzone, A. Bhattacharya, F. Bovolo, and S. Chaudhuri, "A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 154–170, Jan. 2018.

[29] J. Heikkonen and A. Varfis, "Land cover/land use classification of urban areas," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 12, pp. 475–489, 1998.

[30] X. Huang, Q. Lu, and L. Zhang, "A multi-index learning approach for classification of high-resolution remotely sensed images over urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, pp. 36–48, 2014.

[31] K. Yue, L. Yang, R. li, W. hu, F. Zhang, and W. li, "TreeUNet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 1–13, 2019.

[32] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017.

[33] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high resolution aerial imagery," 2016, *arXiv:1606.02585*.

[34] F. Yang *et al.*, "Dually supervised feature pyramid for object detection and segmentation," 2019, *arXiv: 1912.03730*.

[35] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. Pensa, and S. Dupuy, "M3 fusion: A deep learning architecture for multiscale multimodal multitemporal satellite data fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4939–4949, Dec. 2018.

[36] T. Su, "Scale-variable region-merging for high resolution remote sensing image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 319–334, 2019.

[37] C. Peng, Y. Li, L. Jiao, Y. Chen, and R. Shang, "Densely based multi-scale and multi-modal fully convolutional networks for high-resolution remote-sensing image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 8, pp. 2612–2626, Aug. 2019.

[38] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Feb. 2019.

[39] F. Zhou, R. Hang, and Q. Liu, "Class-guided feature decoupling network for airborne image segmentation," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3006872.

[40] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. B. Ayen, "HyperDense-Net: A hyper-densely connected CNN for multi-modal image segmentation," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1116–1126, May 2019.

[41] S. S. Husain and M. Bober, "REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5201–5213, Oct. 2019.

[42] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3006872.

[43] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, "Hyperspectral image classification with attention-aided CNNs," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3007921.

[44] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Data Mining Knowl. Discovery*, vol. 8, 2018, Art. no. e1264.

[45] J. Ball, D. Anderson, and C. S Chan, "A comprehensive survey of deep learning in remote sensing: Theories, tools and challenges for the community," *J. Appl. Remote Sens.*, vol. 11, pp. 1–54, 2017.

[46] K. Arai, "Nonlinear mixture model of mixed pixels in remote sensing satellite images based on monte carlo simulation," *Adv. Space Res.*, vol. 41, pp. 1715–1723, 2008.

[47] W. P. Kustas *et al.*, "Instantaneous and daily values of the surface energy balance over agricultural fields using remote sensing and a reference field in an arid environment," *Remote Sens. Environ.*, vol. 32, pp. 125–141, 1990.

[48] Q. Zhasng and S. Zhu, "Visual interpretability for deep learning: A survey," *Frontiers Inf. Tech. Elect. Eng.*, vol. 19, pp. 30–42, 2018.

[49] M. Kwabena Patrick, A. Felix Adekoya, A. A. Mighty, and B. Y. Edward, "Capsule networks—A survey," *J. King Saud Univ.*, to be published, doi: 10.1016/j.jksuci.2019.09.014.

[50] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2017, pp. 3510–3520.

[51] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," Distill, Nov. 2017.

[52] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz, "Identifying unknown unknowns in the open world: Representations and policies for guided exploration," in *Proc. 31st AAAI Conf. Artif. Intell*., 2017, pp. 2124–2132.

[53] Q. Zhang W. Wang, and S. C. Zhu, "Examining CNN representations with respect to dataset bias," in *Proc. 32nd AAAI Conf. Artif. Intell*., 2018, *arXiv:1710.10577*. [Online]. Available: https://arxiv.org/abs/1710.10577

[54] Q. Zhang, Y. N. Wu, and S. C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2018, pp. 8827–8836, *arXiv:1710.10577*.

[55] T. F. Wu *et al.*, "Interpretable R-CNN," 2017, *arXiv:1711.05226*.

[56] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, Dec. 2017, pp. 1–11.

[57] G. Hinton, A. Krizhevsky, and S. Wang, "Transforming auto-encoders," in *Proc. Artif. Neural Netw. Mach. Learn.*, Jun. 2011, pp. 14–17.

[58] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *Proc. 6th Int. Conf. Learn. Representation*, May 2018, pp. 1–15.

[59] J. Zhao, L. Jiao, S. Xia, and V. Basto Fernandes, "Multi objective sparse ensemble learning by means of evolutionary algorithms," *Decis. Support Syst.*, vol. 111, pp. 86–100, 2018.

[60] J. Hsu, C. Kuo, and D. Chen, "Image super-resolution using capsule neural networks," *IEEE Access*, vol. 8, pp. 9751–9759, 2020.

[61] H. Guan *et al.*, "A convolutional capsule network for traffic-sign recognition using mobile LiDAR data with digital images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1067–1071, Jun. 2020.

[62] R. Varghese, S. Sharma, and M. Premalatha, "Transforming auto-encoder and decoder network for pediatric bone image segmentation using a state-of-the-art semantic segmentation network on bone radiographs," in *Proc. Int. Conf. Intell. Inf. Biomed. Sci.*, 2018, pp. 251–256.

[63] A. Hoogi, B. Wilcox, Y. Gupta, and D. L. Rubin, "Self-attention capsule networks for object classification," 2019, *arXiv:1904.12483*.

[64] C. Xiang L. Zhang, Y. Tang W. Zou, and C. Xu, "MS-CapsNet: A novel multi-scale capsule network," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1850–1854, Dec. 2018.

[65] R. Phaye *et al.*, "Dense and diverse capsule networks: Making the capsules learn better," 2018, *arXiv:1805.04001*.

[66] A. Jaiswal *et al.*, "CapsuleGAN: Generative adversarial capsule network," 2018, *arXiv:1802.06167v7*.

[67] A. Deliège, Adrien, M. Cioppa, and Van Droogenbroeck, "HitNet: A neural network with capsules embedded in a Hit-or-Miss layer, extended with hybrid data augmentation and ghost capsules," 2018, *arXiv:1806.06519*.

[68] R. LaLonde and U. Bagci, "Capsules for object segmentation," 2018, *arXiv:1804.04241*.

[69] Y. Li *et al.*, "The recognition of rice images by UAV based on capsule network," *Cluster Comput.*, vol. 22, no. 4, pp. 9515–9952, 2019.

[70] Y. Yu *et al.*, "Capsule feature pyramid network for building footprint extraction from high-resolution aerial imagery," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.2986380.

[71] Y. Yu, T. Gu, H. Guan, and D. Li, "Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1894–1898, Dec. 2019.

[72] M. Gerke *et al.*, "ISPRS 2D semantic labeling contest." *PCV - Photogrammetric Computer Vision*, Sep. 2014, doi: 10.13140/2.1.3570.9445.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[74] W. D. Thompson and S. Walter, "A reappraisal of the kappa coefficient," *J. Clin. Epidemiol.*, vol. 41, pp. 949–958, 1998.

[75] P. Molchanov *et al.*, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*.

Yujuan Guo received the M.S. degree in geographic information system from Xi'an University of Science and Technology, Xi'an, China, in 2015. She is currently working toward the Ph.D. degree in geographic information system at the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China.

Her research interests include remote sensing image processing, image segmentation, and deep learning.

Jingjuan Liao received the B.S. and M.S. degrees in geosciences from Nanjing University, Nanjing, China, in 1987 and 1990, respectively, and the Ph.D. degree in geophysics from the Institute of Geophysics, Chinese Academy of Sciences, Beijing, China, in 1993.

Since 1993, she has been working on radar remote sensing applications as a Researcher with the Institute of Remote Sensing Applications, Chinese Academy of Sciences. She has rendered the institute great service in a number of research projects. Since 2007, she has been working on microwave remote-sensing application as a Professor with the Center for Earth Observation and Digital Earth and the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences. She is currently working with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences. Her research interests include microwave scattering model, data processing, and surface parameters estimation.

Guozhuang Shen (Member, IEEE) received the B.S. degree in geosciences from Zhejiang University, Hangzhou, China, in 2003, and the Ph.D. degree in geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently working with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests include microwave remote sensing for lake/wetland ecosystems and Moon-based remote sensing.