Resolution-agnostic Remote Sensing Scene Classification with Implicit Neural Representations

Keyan Chen, Wenyuan Li, Jianqi Chen, Zhengxia Zou* and Zhenwei Shi, *Member, IEEE*

***Abstract*—Remote sensing scene classification is an important yet challenging task. In recent years, the excellent feature representation ability of Convolutional Neural Networks (CNNs) has led to substantial improvements in scene classification accuracy. However, handling resolution variations of remote sensing images is still challenging because CNNs are not inherently capable of modeling multi-resolution input images. In this letter, we propose a novel scene classification method with scale and resolution adaptation ability by leveraging the recent advances in Implicit Neural Representations (INRs). Unlike previous CNN-based methods that make predictions based on rasterized image inputs, the proposed method converts the images as continuous functions with INRs optimization and then performs classification within the function space. When the image is represented as a function, the image resolution can be decoupled from the pixel values so that the resolution does not have much impact on the classification performance. Our method also shows great potential for multi-resolution remote sensing scene classification. Using only a simple Multilayer Perceptron (MLP) classifier in the proposed function space, our method achieves classification accuracy comparable to deep CNNs but exhibits better adaptability to image scale and resolution changes.**

***Index Terms*—Remote sensing images, scene classification, implicit neural networks, resolution agnostic.**

I. Introduction

WITH the recent advances in high-resolution earth observation [1], [2], remote sensing scene classification has shown increasing attention owing to its advantages for various applications, including surveying and mapping, land use identification, and urban planning.

To accurately obtain the semantic categories of ground objects, some early methods were proposed by utilizing feature engineering, exploring feature construction, feature extraction, and feature selection in an effort to extract more effective feature representation, such as color histogram [3], local binary pattern [4], codebook [5], etc. Recently, deep learning has greatly promoted the research of remote sensing scene classification. Many visual image classification architectures, such as VGGNet [6], ResNet [7] and ViT [8] have been introduced to remote sensing scene classification tasks. The excellent feature representation ability of CNNs and Transformers has led to substantial improvements in scene classification accuracy. Zhang *et al.* [9] create a CNN-CapsNet hybrid architecture by combining the benefits of CNN and CapsNet. Zheng *et al.* [10] propose a technique for deep scene representation to address the lack of geometric invariance of CNN activations. Liu *et al.* [11] propose a multi-scale CNN architecture to address the scale variability of objects in remote sensing images.

Despite the progress in deep learning based methods, remote sensing scene classification still remains a challenge due to the varying resolutions amongst different remote sensors. Most previous methods rarely considered the applicability and transferability in real-world application settings, i.e., the data disturbances generated by diverse imaging environments and remote sensors, which manifest mostly in the resolution disparity across different domains. State-of-the-art deep learning architectures, including CNNs and Transformers, are naturally not capable of modeling multi-resolution input images, and thus it is challenging to represent the homogeneity of heterogeneous remote sensing images in an effective manner.

Recently, the use of neural networks to approximate continuous space and temporal functions has become an emerging research topic, known as implicit neural representations (INRs). INRs can represent the properties of a space point/time point as a function of the corresponding coordinates. The primary issue with INRs is preserving the high-frequency details. To achieve this, position encoding (PE) [12] or periodic activation function [13], also known as Sirens, is typically used. Sirens consider the sinusoidal activation an essential network component. Functa [14] with Sirens structure is the first to explore deep learning in the function space of INRs for tasks such as generative modeling, data interpolation and new view synthesis. However, the datasets utilized in the preceding tasks are quite straightforward.

In this letter, we propose a novel method named Resolution-Agnostic Scene classification Network (RASNet), to improve the transferability and mitigate the scene classification degradation caused by resolution changes in different domains. RASNet adheres to the following two steps: 1) Represent remote sensing images as continuous functions by INRs. In this stage, we present a Synthesizer that converts the continuous coordinates into pixel values for the corresponding location. As a result, each image can be represented by a function, i.e., the Synthesizer's weights. However, employing all of the Synthesizer's parameters for categorization will add a significant burden. To address this issue, we propose

The work was supported by the National Key Research and Development Program of China (Titled "Brain-inspired General Vision Models and Applications"), the National Natural Science Foundation of China under Grant 62125102, and the Fundamental Research Funds for the Central Universities. *(Corresponding author: Zhengxia Zou (e-mail: zhengxiazou@buaa.edu.cn))*

Keyan Chen, Wenyuan Li, Jianqi Chen, and Zhenwei Shi are with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with Shanghai Artificial Intelligence Laboratory. Zhengxia Zou is with the Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China, and also with Shanghai Artificial Intelligence Laboratory.

a Modulator that can transfer low-dimensional latent code to high-dimensional Synthesizer weights, hence facilitating classification. Details are shown in Sec. II-C and Sec. II-D. 2) Design and train a classification network in the function space. In the process of modeling an image as a function, the image's resolution can be decoupled from the pixel data so that the impact on classification performance at various resolutions can be greatly reduced. Ideally, the data points in the function space no longer include resolution information.

Our contributions are summarized as follows:

1. We propose a novel method to achieve the resolution-agnostic remote sensing image classification by decomposing the task into a data space transfer task and a classifier construction task in the functional space.

2. We propose a network modulator to generate the modulation parameters of the synthesizer, which reduces the dimension of the data sample in the function space and improves the optimization of the latent codes.

3. We develop a residual Sirens network with perceptual loss that allows the implicit neural synthesizer to employ semantic context to optimize the latent codes and enhance the performance of downstream classification tasks.

II. Methodology

A. Overview

The proposed RASNet decomposes scene classification into two subtasks: 1. Optimize each image as a data point in function space. 2. Create a classifier in the function space. The proposed Modulator and Synthesizer constitute subtask 1, as shown in Fig. 1. The Modulator converts the unique latent code into the shift of bias of each fully connected (FC) layer in the Synthesizer, a process known as shift modulation [14].

In subtask 1, the Synthesizer directly maps the coordinates of image pixels to the corresponding pixel's RGB values. In addition to the Modulator and the Synthesizer, we also design a Preceptor to increase the semantic expression capability. Both the pixel-level consistency and the semantic-level consistency are considered in the optimization. Since fitting each sample demands a substantial amount of computation, meta-learning is used to learn a better initialization to accelerate the optimization of the latent code in RASNet. The parameters of Modulator and Synthesizer are shared across data to describe the common structure of images, which not only reduces the dimension of data points in the function space but also mines the differences between data points.

In subtask 2, we aim to categorize the data (latent codes) in the function space. We show that with the help of implicit modulation and meta learning, only a few basic MLP layers are sufficient to get considerable classification performance.

B. Revisiting the Implicit Neural Representation

This section provides a quick overview of the INRs. INRs refer to a mapping $F_\theta : R^n \to R^m$, which encodes signals such as amplitude, pixel value, 3D shape, etc. as a function of time or spatial location. Taking representing an image as an example, F_θ, usually an MLP, converts coordinates to pixel values. The image is fitted by minimizing the mean square error (MSE) of the pixel value, so that each image is determined by the parameter θ^* of the optimized function F_θ. Since F_θ decouples the image resolution, theoretically, images of any resolution can be obtained by different sampling intervals in R^n space.

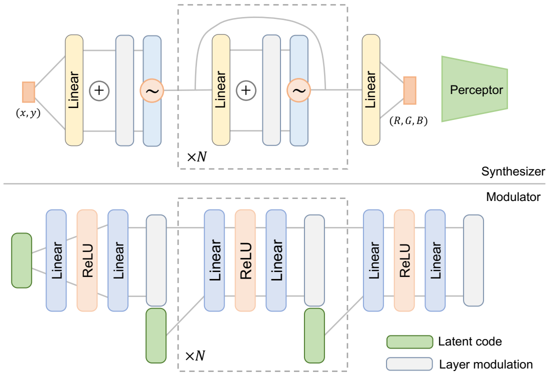


Fig. 1. An overview of the Modulator and Synthesizer in our method.

C. The Synthesizer

The Synthesizer aims to map image coordinates to pixel values, as shown in Fig. 1. The Synthesizer has N intermediate residual layers that are written as follows,

$$h_i = \sin\left(\omega(w_i^T h_{i-1} + b_i + m_i)\right) + h_{i-1}, \tag{1}$$

where the sinusoidal periodic activation functions [13] are embedded into the residual layers. $h_{i-1} \in \mathcal{R}^{d_{i-1}}$ and $h_i \in \mathcal{R}^{d_i}$ are the hidden features of the $(i-1)$th and ith layers, respectively. $w_i \in \mathcal{R}^{d_{i-1} \times d_i}$ and $b_i \in \mathcal{R}^{d_i}$ are the weight and bias of the $(i-1)$th linear layer. $m_i \in \mathcal{R}^{d_i}$ is the shift bias produced by the Modulator, and ω is a pre-defined scale factor. We set $\omega = 30$ according to [13].

During the optimization, we build a Perceptor as a loss function to introduce the global semantic consistency [15]. The Synthesizer's total loss function is presented,

$$\begin{aligned}\mathcal{L}_s = & \frac{1}{H \times W} \sum_i^H \sum_j^W \|\mathrm{S}(i, j; w, b, m) - I(i, j)\|_2^2 \\ & + \lambda(1 - \cos\left(\mathrm{P}(\hat{I}; \theta), \mathrm{P}(I; \theta)\right),\end{aligned} \tag{2}$$

where H, W are the image height and width, w, b, m are the weight, bias, and shift bias of the Synthesizer, I is the original image, θ is the Perceptor's parameters, $\hat{I}$ is the synthesized image using coordinates, and λ is a hyperparameter that balances pixel value loss and perceptual loss. We set $\lambda = 0.2$. The θ is fixed during the whole procedure.

D. The Modulator

Although the optimized Synthesizer parameters can be already used for scene classification, we found that the number of parameters far exceeds the image itself, which is not friendly to downstream classification tasks. Some recent studies, e.g., Functa [14], propose taking a latent code to regulate the frequency and phase of the periodic activation function. Similar to Functa [14], we control the frequency and phase of the activation in Synthesizer by adjusting the Synthesizer's bias shift. An input-sensitive Modulator instead of the basic single-layer linear map in Functa [14] is proposed to address the challenge of complex scenes in remote sensing images.

We design the structure of this input-sensitive Modulator as an MLP, as shown in Fig. 1. The input mapping layer transfers the latent code z to the layer modulation vector of the first layer in the Synthesizer. Subsequent layers concatenate the preceding modulation vector and the same latent code z. Through this design, the loss gradient in the optimization process can be easily back-propagated to the latent code z and speeds up the optimization process. The formulation of the ith layer can be written as:

$$\begin{aligned} x &= [z, m_{i-1}] \\ m_i &= {\phi^2_i}^T \max(0, {\phi^1_i}^T x), \end{aligned} \tag{3}$$

where $z \in \mathcal{R}^{d_z}$ is the Modulator input, i.e., the latent code vector. $m_{i-1} \in \mathcal{R}^{d_m}$ and $m_i \in \mathcal{R}^{d_m}$ are the modulation vectors generated by the $(i-1)$th and ith layers in Modulator, as well as the $(i-1)$th and ith bias shift in Synthesizer. ${\phi^1}_i \in \mathcal{R}^{(d_z+d_m)\times d_m}$ and ${\phi^2}_i \in \mathcal{R}^{d_m \times d_m}$ are the parameters of the Modulator linear layers. By mapping the low-dimensional latent code to the bias shift of each layer in Synthesizer, Modulator can modify the mapping function of Synthesizer.

Algorithm 1 Meta-learning to transform images to data points in function space

```
Input: $\mathcal{I}$, the input images
Input: $N_e$, the max epoch while learning
Input: $N_i$, the max inner loop while learning
Input: $\epsilon$, $\epsilon'$, the learning rate of the inner loop and the outer loop
Input: $x$, the coordinate vector
Output: $\theta_s$ and $\theta_m$, weights of the Synthesizer and the Modulator
Initialize $\theta_s$ and $\theta_m$
for $i = 0$ to $N_e$ do
    Randomly divide all samples to batch $\mathcal{B}_j$            ▷ j is batch id
    for  each $\mathcal{B}_j$ do
        Set all $z_k \rightarrow 0$                          ▷ k is sample id of $\mathcal{B}_j$
        for step = 0 to $N_i$ do
            $z_k \leftarrow z_k - \epsilon \nabla_z \mathcal{L}_s(x, \mathcal{I}^j_k)|_{z=z_k}$  ▷ $\mathcal{I}^j_k$ denotes the k$^{\text{th}}$
sample in j$^{\text{th}}$ batch
        end for
        $\theta \leftarrow \theta - \epsilon' \nabla_\theta \frac{1}{|B_j|} \sum_{k=0}^{|B_j|} \mathcal{L}_s(x, \mathcal{I}^j_k)$      ▷ $\theta = \{\theta_s, \theta_m\}$
    end for
end for
Return: $\theta_s$ and $\theta_m$
```

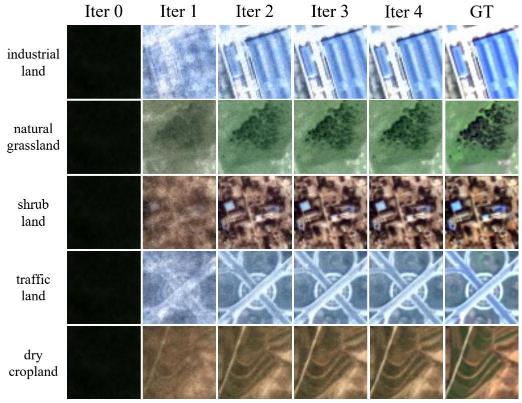


Fig. 2. Visualization of optimized images. Rows show various image samples. Columns show images reconstructed after various optimization iterations.

E. Optimization Procedure

To jointly optimize the network parameters and image-specific latent codes, the most straightforward way is to first update the network parameters on the entire dataset and then freeze the networks and update the latent for each image. However, we found that the optimization process is time-consuming in this manner. Therefore, we introduce meta-learning to initialize the parameters of Synthesizer and Modulator and consider optimization of each image as a subtask [16].

We design the optimization process of images from euclidean space to function space based on MAML [17]. Algorithm 1 provides the pseudo code. The biases of the Synthesizer's linear layers are initialized to all zero and the weight of the first layer is initialized with Uniform($-\frac{1}{nc}, \frac{1}{nc}$), while the remaining layers are initialized with Uniform($-\frac{1}{w}\sqrt{\frac{6}{nc}}, \frac{1}{w}\sqrt{\frac{6}{nc}}$), where $w = 30$, where nc denotes the number of input channels for each linear layer.

In the inner loop, the model parameters are fixed, and only the latent codes are updated; in the outer loop, the loss is calculated using the batch data and the model parameters of the Modulator and Synthesizer are updated. After obtaining θ_s and θ_m by meta-learning, we can go for the inner loop to retrieve the latent code (i.e., data point in the function space) of each sample. During meta-learning, we set $N_i = 5$. Fig. 2 demonstrates that a modest number of optimization steps (e.g., 4) may provide a nice visualization, but in order to obtain high accuracy on the downstream task, we take more steps.

F. The Classifier

We perform scene classification directly on the data points in the function space, which not only saves memory, but also reduces the effect of resolution variations on classification performance. With a tiny MLP, we can achieve equivalent accuracy to popular CNN-based classification networks and show better adaptation ability to image resolution changes. The MLP only has 5 256-dim FC layers with ReLU activation, resulting in low computational expense and rapid training speed. More specifically in classification process, when all training images are mapped to function points (latent codes), these codes can be used to train the classifier. The well-trained classifier can classify a new image after fitting the new image into a latent code.

III. Experiment and Analysis

A. Dataset Description

We conduct our experiments on the Gaofen Image Dataset (GID) [18] to validate the efficacy of our method. In our experiments, we only employ RGB bands from the fine land-cover classification subset with 15 categories. Each class contains 1000, 600, and 400 samples with image size of 56×56, 112×112, and 224×224 pixels, respectively. The images are produced by the Gaofen-2 satellite multispectral camera with a spatial resolution of 4m/pixel. To prepare our training set, we sample 600 images with 56×56 pixels from each class, and resize them to 28×28 pixels. The training set is denoted as $\mathcal{D}_{28}^{tr}$. To prepare our test set, we sample 200 images of 56×56, 112×112, and 224×244 pixels, respectively. We then resize them to 28×28, 56×56, and 112×112 pixels, respectively. A center crop of 84×84 pixels is applied to the 112×112 images. As a consequence, three test sets ($\mathcal{D}_{28}^{te}$, $\mathcal{D}_{56}^{te}$, $\mathcal{D}_{84}^{te}$) finally have a spatial resolution of 8m/pixel covering different ground scales.

B. Experimental Setup

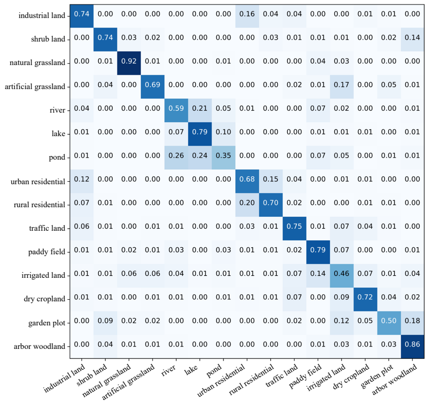
Except for Synthesizer, which follows Sec. II-E to initialize its parameters, other networks are initialized by the PyTorch default configuration. The Preceptor is a pretrained Resnet18 [7]. In Sec. II-E, we set the number of training epochs to $N_e = 1000$ and the number of inner loops to $N_i = 5$. The optimizer in the outer loop is AdamW with a learning rate of $1e-5$ and the inner is momentum SGD of $1e-2$. The input coordinates are normalized to $[-1, 1]$, and an additional 0.5 is added to the output. We employ 7 repetition blocks with 256 neurons for the Synthesizer, i.e., $N = 7$ in Fig. 1. Blocks in Modulator match the Synthesizer. The dim of the optimized latent code is set to 512. The classifier is trained for 100 epochs using AdamW with a learning rate of $1e-4$. Notably, we only train the classifier using $\mathcal{D}_{28}^{tr}$ since we want to evaluate resolution changes. We employ Precision, Recall, and F1 to measure the performance.

C. Comparison with State-of-the-art Methods

We compare the proposed RASNet with other deep learning-based image classification methods, such as ResNet18 [7], VGG16 [6], and INR-based Functa [14] on different test datasets. Tab. I shows the comparison results except for Functa as a part of ablation study in Tab. III. We have the following observations. 1) When evaluated on $\mathcal{D}_{28}^{te}$, RASNet has an up-to-par performance, but is inferior to VGG16. This is because we only utilize a tiny MLP classifier and there is still room for accuracy improvement in data space transfer. 2) When the spatial resolution is held constant, Res. = 8, and only the scale of the scene varies in $\mathcal{D}_{56}^{te}$ and $\mathcal{D}_{84}^{te}$, the performance of CNN-based methods drops significantly, whereas RASNet still maintains the performance at a high score. This suggests that RASNet can enhance the adaptability of various spatial ranges, i.e., spatial dimension agnostic. 3) When we downsize images from $\mathcal{D}_{56}^{te}$ and $\mathcal{D}_{84}^{te}$ to 28×28 (the same size as training set $\mathcal{D}_{28}^{tr}$), and modify the spatial resolutions, we observe that CNN-based classifiers perform better, but still experience a considerable performance decrease. RASNet still maintains performance despite a slight drop, demonstrating that RASNet can expand its adaptability at multiple resolutions, i.e., resolution dimension agnostic. In contrast to CNN-based approaches, the modest decline of RASNet may be due to the loss of image details at a low resolution, whereas this can be easily fixed by encoding the input image at a higher resolution. 4) With perceptor loss, RASNet* achieves sota. More information is provided in Sec. III-D3. Furthermore, Fig. 3 reports the confusion matrix, where the entry in the ith row and jth column denotes the rate of images from the ith class classified as the jth class. Tab. II displays the accuracy per class. Ponds and irrigated land can be difficult to accurately categorize. Pond is easily confused with river and lake, whereas irrigated land is easily confused with other land and land-like area.

TABLE I
COMPARISONS ACROSS DIFFERENT TESTSETS. **SIZE**: THE HEIGHT AND WIDTH OF THE IMAGE. **RES.**: THE SPATIAL RESOLUTION.

Testset	**Method**	**Size**	**Res.** (m/pixel)	**Precision** (%)	**Recall** (%)	**F1** (%)
$\mathcal{D}_{28}^{te}$	**Resnet18**	28	8	62.81	60.67	61.41
	VGG16	28	8	67.93	65.57	66.63
	RASNet	28	8	61.07	61.17	60.93
	RASNet*	28	8	69.28	68.70	68.99
$\mathcal{D}_{56}^{te}$	**Resnet18**	56 28	8 16	32.54 52.32	23.93 48.47	18.92 49.12
	VGG16	56 28	8 16	57.61 59.73	51.47 54.83	52.71 54.57
	RASNet	56 28	8 16	59.67 58.14	58.35 57.27	58.38 57.26
	RASNet*	56 28	8 16	63.48 64.65	63.23 62.70	63.13 63.11
$\mathcal{D}_{84}^{te}$	**Resnet18**	84 28	8 24	30.74 47.13	18.50 44.10	13.11 43.73
	VGG16	84 28	8 24	55.32 56.94	48.43 49.13	47.43 47.92
	RASNet	84 28	8 24	57.15 56.91	55.40 54.60	54.12 54.17
	RASNet*	84 28	8 24	61.98 60.20	59.03 58.23	59.78 57.30



	industrial land	shrub land	natural grassland	artificial grassland	river	lake	pond	urban residential	rural residential	traffic land	paddy field	irrigated land	dry cropland	garden plot	arbor woodland
industrial land	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.04	0.04	0.00	0.00	0.01	0.01	0.00
shrub land	0.00	0.74	0.03	0.02	0.00	0.00	0.00	0.00	0.03	0.01	0.01	0.01	0.00	0.02	0.14
natural grassland	0.00	0.01	0.92	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.04	0.03	0.00	0.00	0.00
artificial grassland	0.00	0.04	0.00	0.69	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.17	0.00	0.05	0.01
river	0.04	0.00	0.00	0.00	0.59	0.21	0.05	0.01	0.00	0.00	0.07	0.02	0.00	0.00	0.01
lake	0.01	0.00	0.00	0.00	0.07	0.79	0.10	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.01
pond	0.01	0.00	0.00	0.00	0.26	0.24	0.35	0.00	0.00	0.00	0.07	0.05	0.00	0.01	0.01
urban residential	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.68	0.15	0.04	0.00	0.01	0.01	0.00	0.00
rural residential	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.20	0.70	0.02	0.00	0.00	0.01	0.01	0.00
traffic land	0.06	0.01	0.00	0.01	0.01	0.00	0.00	0.03	0.01	0.75	0.01	0.07	0.04	0.01	0.00
paddy field	0.01	0.01	0.02	0.01	0.03	0.00	0.03	0.01	0.01	0.02	0.79	0.07	0.00	0.00	0.01
irrigated land	0.01	0.01	0.06	0.06	0.04	0.01	0.01	0.01	0.01	0.07	0.14	0.46	0.07	0.01	0.04
dry cropland	0.01	0.01	0.00	0.01	0.01	0.00	0.01	0.01	0.01	0.07	0.00	0.09	0.72	0.04	0.02
garden plot	0.00	0.09	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.12	0.05	0.50	0.18
arbor woodland	0.00	0.04	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.01	0.03	0.86

Fig. 3. Confusion matrix of RASNet* on $\mathcal{D}_{28}^{te}$.

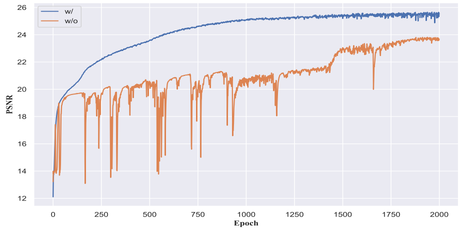


Fig. 4. Reconstruction accuracy (PSNR) on different optimization steps of the Synthesizer w/ or w/o using our residual design.

D. Ablation Study

1) Residual Connection: To improve the optimization, our RASNet designs a residual mapping between layers, as depicted in Fig. 1. Fig. 4 shows the reconstruction accuracy (PSNR) w/ or w/o using residual mapping. It can be seen that the proposed residual design in Synthesizer increases the accuracy, optimization speed, and stability significantly.

TABLE II
THE ACCURACY PER CLASS OF RASNET* ON $\mathcal{D}_{28}^{te}$.

	industrial land	shrub land	natural grassland	artificial grassland	river	lake	pond	urban residential	rural residential	traffic land	paddy field	irrigated land	dry cropland	garden plot	arbor woodland
Precision (%)	68.52	77.08	87.62	84.76	58.91	62.95	63.39	62.50	74.21	70.75	67.81	40.89	78.38	74.07	67.32
Recall (%)	74.00	74.00	92.00	69.50	59.50	79.00	35.50	67.50	70.50	75.00	79.00	46.00	72.50	50.00	86.50
F1 (%)	71.15	75.51	89.76	76.37	59.20	70.07	45.51	64.90	72.31	72.82	72.98	43.29	75.32	59.70	75.71

TABLE III
ABLATION STUDY ON THE MODULATOR AND PERCEPTOR. OURS (W/O MODULATOR) IS THE INR-BASED FUNCTA.

Testset	**Method**	**Size**	**Res.** (m/pixel)	**Precision** (%)	**Recall** (%)	**F1** (%)
$\mathcal{D}_{28}^{te}$	**w/o Modulator**	28	8	59.90	58.90	58.20
	w/ Perceptor	28	8	69.28	68.70	68.99
$\mathcal{D}_{56}^{te}$	**w/o Modulator**	56	8	57.11	55.57	54.44
		28	16	56.33	53.87	53.46
	w/ Perceptor	56	8	63.48	63.23	63.13
		28	16	64.65	62.70	63.11
$\mathcal{D}_{84}^{te}$	**w/o Modulator**	84	8	53.13	51.53	51.48
		28	24	54.12	50.60	51.99
	w/ Perceptor	84	8	61.98	59.03	59.78
		28	24	60.20	58.23	57.30

2) Modulator: Datasets such as CelebA [19] are frequently used in INRs of natural images. It is easy to optimize these data due to the simple image content and similar image structures. However, remote sensing scenes are more complex and diverse. We take Modulator in place of Shift Modulation in Functa [14] to get modulations. We show that this design is superior to that of Functa, as evidenced by Tab. III. Furthermore, if we simply utilize the Synthesizer without the Modulator to optimize each latent code (the weights of the entire Synthesizer), the function space dim will increase from 512 to 0.5M. The classifier will face a considerable hurdle.

3) Perceptor: Scene classification can benefit from the regional/global receptive field, whereas the INRs reconstruct images solely at the pixel level. For this purpose, we propose Perceptor to regularize semantic consistency. The w/ Perceptor rows in Tab. III shows accuracy gains with the Perceptor. The adaptability in various scene scales and spatial resolutions also shows superior to that of Resnet18 and VGG16.

IV. DISCUSSION AND CONCLUSION

In this letter, we propose a novel method named RASNet for remote sensing scene classification. RASNet is designed based on the recent advances of INRs and can adapt to both scale and spatial resolution changes. Using INRs, the proposed RASNet transforms images from pixels to the function space, where they are then classified. Our method achieves classification accuracy comparable to deep CNNs yet shows much better adaptivity to image scales and resolution changes. We show that for complex and diverse scenes and large-scale images, the proposed modulation of INRs is necessary for accurate classification. Also, a basic MLP classifier can produce results on par with CNNs, but the BN layer, the optimizer, etc. must be considered specially. The method is "resolution-agnostic" within a certain range, and it can't perform well in extreme cases, the performance of which is still better than that of CNNs. Our future research will consider hyper-networks, multi scales, transformers, etc. to handle high-resolution images and improve downstream performance.

REFERENCES

[1] W. Li, K. Chen, H. Chen, and Z. Shi, "Geographical knowledge-driven representation learning for remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[2] K. Chen, Z. Zou, and Z. Shi, "Building extraction from remote sensing images with sparse token transformers," *Remote Sensing*, vol. 13, no. 21, p. 4441, 2021.

[3] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 7, pp. 729–736, 1995.

[4] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2889–2901, 2017.

[5] S. Jiang, G. Chen, X. Song, and L. Liu, "Deep patch representations with shared codebook for scene classification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1s, pp. 1–17, 2019.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[9] W. Zhang, P. Tang, and L. Zhao, "Remote sensing image scene classification using cnn-capsnet," *Remote Sensing*, vol. 11, no. 5, p. 494, 2019.

[10] X. Zheng, Y. Yuan, and X. Lu, "A deep scene representation for aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4799–4809, 2019.

[11] Y. Liu, Y. Zhong, and Q. Qin, "Scene classification based on multiscale convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7109–7121, 2018.

[12] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[13] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[14] E. Dupont, H. Kim, S. Eslami, D. Rezende, and D. Rosenbaum, "From data to functa: Your data point is a function and you should treat it like one," *arXiv preprint arXiv:2201.12204*, 2022.

[15] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[16] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng, "Learned initializations for optimizing coordinate-based neural representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2846–2855.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[18] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sensing of Environment*, vol. 237, p. 111322, 2020.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.