RSMamba: Remote Sensing Image Classification with State Space Model

Keyan Chen[1], Bowen Chen[1], Chenyang Liu[1], Wenyuan Li[2], Zhengxia Zou[1], Zhenwei Shi[1,⋆]

Beihang University[1], The University of Hong Kong[2]

Abstract—**Remote sensing image classification forms the foundation of various understanding tasks, serving a crucial function in remote sensing image interpretation. The recent advancements of Convolutional Neural Networks (CNNs) and Transformers have markedly enhanced classification accuracy. Nonetheless, remote sensing scene classification remains a significant challenge, especially given the complexity and diversity of remote sensing scenarios and the variability of spatiotemporal resolutions. The capacity for whole-image understanding can provide more precise semantic cues for scene discrimination. In this paper, we introduce RSMamba, a novel architecture for remote sensing image classification. RSMamba is based on the State Space Model (SSM) and incorporates an efficient, hardware-aware design known as the Mamba. It integrates the advantages of both a global receptive field and linear modeling complexity. To overcome the limitation of the vanilla Mamba, which can only model causal sequences and is not adaptable to two-dimensional image data, we propose a dynamic multi-path activation mechanism to augment Mamba's capacity to model non-causal data. Notably, RSMamba maintains the inherent modeling mechanism of the vanilla Mamba, yet exhibits superior performance across multiple remote sensing image classification datasets. This indicates that RSMamba holds significant potential to function as the backbone of future visual foundation models. The code will be available at https://github.com/KyanChen/RSMamba.**

Index Terms—**Remote sensing images, image classification, foundation model, backbone network, Mamba**

I. Introduction

THe advancement of remote sensing technology has significantly heightened interest in high-resolution earth observation. Remote sensing image classification, serving as the bedrock of remote sensing image intelligent interpretation, is a crucial element for subsequent downstream tasks. It plays a pivotal role in applications such as land mapping, land use, and urban planning. Nonetheless, the complexity and diversity of remote sensing scenarios, coupled with the variable spatio-temporal resolution, present substantial challenges to automated remote sensing image classification [1–4].

Researchers have been diligently working towards alleviating these challenges and enhancing the models' applicability across diverse application scenarios. Early methodologies predominantly focused on feature construction, extraction, and selection, investigating feature engineering machine learning methods represented by SIFT, LBP, color histograms, GIST, BoVW [5], *etc*. In recent years, the advent of deep learning has revolutionized the conventional paradigm that heavily relied on specialized human prior knowledge. Deep learning possesses the capability to autonomously mine effective features from data and output classification probabilities in an end-to-end manner. In terms of network architecture, it can primarily be categorized into CNNs and attention networks. The former abstracts image features layer by layer through two-dimensional convolution operations, as demonstrated by ResNet [6]. The latter captures long-distance dependencies between local areas of the entire image through the attention mechanism, thereby achieving a more robust semantic response, represented by ViT [7], SwinTransformer [8], *etc*. Substantial progress has also been made in remote sensing image classification. For instance, ET-GSNet [9] distills the rich semantic prior of ViT into ResNet18, fully capitalizing on the strengths of both. P2Net [10] introduces an asynchronous contrastive learning method to address the issue of small inter-class differences in fine-grained classification.

To a certain extent, the classification accuracy heavily depends on the model's ability to effectively handle the impact of complex and diverse remote sensing scenarios and variable spatio-temporal resolution. Transformer [11], based on the attention mechanism and capable of obtaining responses from valuable areas across the entire image, presents an optimal solution to these challenges. However, its attention calculation, characterized by square complexity, poses significant challenges in terms of modeling efficiency and memory usage as the input sequence length increases or the network deepens. The State Space Model (SSM) [12] can establish long-distance dependency relationships through state transitions and execute these transitions via convolutional calculations, thereby achieving near-linear complexity. Mamba [13] proves highly efficient for both training and inference by incorporating time-varying parameters into the plain SSM and conducting hardware optimization. Vim [14] and VMamba [15] have successfully introduced Mamba into the two-dimensional visual domain, achieving a commendable balance of performance and efficiency across multiple tasks.

In this paper, we introduce RSMamba, an efficient state space model for remote sensing image classification. Owing to its robust capability in modeling global relationships within an entire image, RSMamba can also exhibit potential versatility across a broad spectrum of other tasks. RSMamba is based on the previous Mamba [13], but has introduced a dynamic multi-path activation mechanism to alleviate the limitations of the plain Mamba, which can only model in a single direction and is position-agnostic. Significantly, RSMamba is designed to preserve the inherent modeling mechanism of the original Mamba block, while introducing non-causal

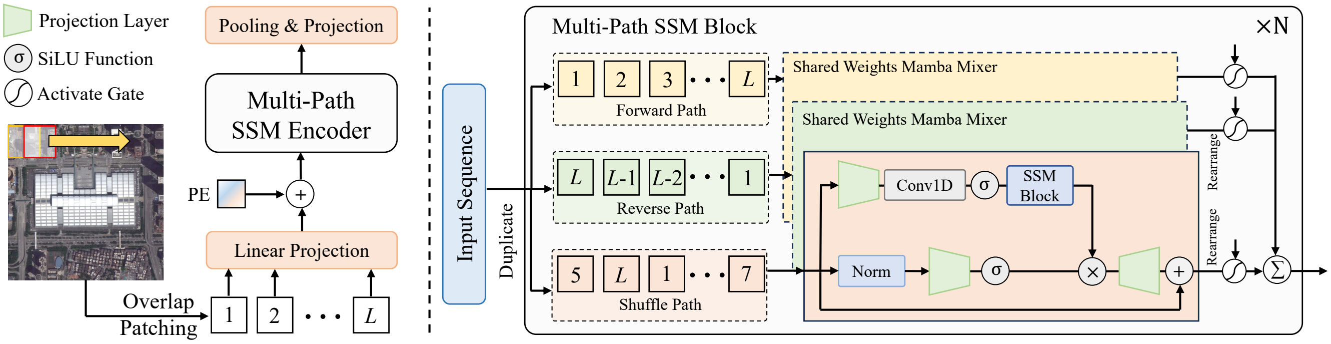

Fig. 1. An overview of the proposed RSMamba.

and position-positive improvements external to the block. Specifically, the remote sensing image is partitioned into overlapping patch tokens, to which position encoding is added to form a sequence. We construct three path copies, namely forward, reverse, and random. These sequences are modeled to incorporate global relationships through the Mamba block using shared parameters, and subsequently activated through linear mapping across different paths. Given the efficiency of the Mamba block, large-scale pre-training of RSMamba can be achieved cost-effectively.

The primary contributions of this paper can be summarized as follows:

i) We propose RSMamba, an efficient global feature modeling methodology for remote sensing images based on the State Space Model (SSM). This method offers substantial advantages in terms of representational capacity and efficiency and is expected to serve as a feasible solution for handling large-scale remote sensing image interpretation.

ii) Specifically, we incorporate a position-sensitive dynamic multi-path activation mechanism to address the limitation of the original Mamba, which was restricted to modeling causal sequences and was insensitive to the spatial position.

iii) We conducted comprehensive experiments on three distinct remote sensing image classification datasets. The results indicate that RSMamba holds significant advantages over classification methods based on CNNs and Transformers.

II. Methodology

Leveraging the inherent characteristics of the SSM model, RSMamba is proficient in effectively capturing the global dependencies within remote sensing images, thereby yielding a wealth of semantic category information. This section will begin with an introduction to the preliminaries of SSM, followed by an overview of RSMamba. Subsequently, we will explore the dynamic multi-path activation block in depth. Finally, we will elaborate on the network structure for three distinct versions of RSMamba.

A. Preliminaries

The State Space Model (SSM) is a concept derived from modern control theory's linear time-invariant system which maps the continuous stimulation $x \in \mathbb{R}^N$ to response $y \in \mathbb{R}^N$. This process can be formulated through the subsequent linear ordinary differential equation (ODE),

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t) \\ y(t) &= \mathbf{C}h(t) \end{aligned} \tag{1}$$

where $y \in \mathbb{R}^N$ is derived from the input signal $x \in \mathbb{R}^N$ and the hidden state $h \in \mathbb{R}^N$. $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the state transition matrix. $\mathbf{B} \in \mathbb{R}^N$ and $\mathbf{C} \in \mathbb{R}^N$ are the projection matrices. To realize the continuous system depicted in Eq. 1 in a discretized form and integrate it into deep learning methods. $\mathbf{A}$ and $\mathbf{B}$ are discretized using a zero-order hold (ZOH) with a time scale parameter Δ. The process is shown as follows,

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta\mathbf{A}) \\ \bar{\mathbf{B}} &= (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \end{aligned} \tag{2}$$

After discretization, Eq. 1 can be rewritten as,

$$\begin{aligned} h_k &= \bar{\mathbf{A}}h_{k-1} + \bar{\mathbf{B}}x_k \\ y_k &= \bar{\mathbf{C}}h_k \end{aligned} \tag{3}$$

where $\bar{\mathbf{C}}$ represents $\mathbf{C}$. At last, the output can be calculated in a convolution representation, as follows,

$$\begin{aligned} \bar{\mathbf{K}} &= (\bar{\mathbf{C}}\bar{\mathbf{B}}, \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}}, \cdots, \bar{\mathbf{C}}\bar{\mathbf{A}}^{L-1}\bar{\mathbf{B}}) \\ \mathbf{y} &= \mathbf{x} * \bar{\mathbf{K}} \end{aligned} \tag{4}$$

where L is the length of the input sequence, and $\bar{\mathbf{K}} \in \mathbb{R}^L$ denotes the structured convolutional kernel.

B. RSMamba

RSMamba transforms 2-D images into 1-D sequences and captures long-distance dependencies using the Multi-Path SSM Encoder, as depicted in Fig. 1. Given an image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$, we employ a 2-D convolution with a kernel of k and a stride of s to map local patches into pixel-wise feature embeddings. Subsequently, the feature map is flattened into a 1-D sequence. To preserve the relative spatial position relationship within the image, we incorporate position encoding P. The entire process is as follows,

$$\begin{aligned} T &= \Phi_{\text{Flatten}}(\Phi_{\text{Conv2D}}(\mathcal{I}, k, s)) \\ T &= T + P \end{aligned} \tag{5}$$

where Φ_{Conv2D} represents the 2-D convolution, while Φ_{Flatten} signifies flattening operation. $T \in \mathbb{R}^{L\times d}$ and $P \in \mathbb{R}^{L\times d}$

correspond to the input 1-D sequence and positional encoding, respectively.

In RSMamba, we have not utilized the [CLS] token to aggregate the global representation, as is done in ViT. Instead, the sequence is fed into multiple dynamic multi-path activation Mamba blocks for long-distance dependency modeling. Subsequently, the dense features necessary for category prediction are derived through a mean pooling operation applied to the sequence. This procedure can be iteratively delineated as follows,

$$\begin{aligned} T^i &= \Phi^i_{\text{mp-ssm}}(T^{i-1}) + T^{i-1} \\ \hat{s} &= \Phi_{\text{proj}}(\Phi_{\text{LN}}(\Phi_{\text{mean}}(T^N))) \end{aligned} \tag{6}$$

where i signifies the ith layer, while T^i represents the output sequence of the ith-layer, with $T^0 = T \in \mathbb{R}^{L \times d}$. $\Phi_{\text{mp-ssm}}$ denotes the dynamic multi-path activation Mamba block, with a total number of N. Φ_{mean} symbolizes mean pooling operation with the sequence dimension and Φ_{LN} is layer normalization. Φ_{proj} is used to project the latent dimension d to the number of classes.

C. Dynamic Multi-path Activation

The vanilla Mamba is employed for the causal modeling of 1-D sequences. It encounters difficulties in modeling spatial positional relationships and unidirectional paths, thereby limiting the applicability to visual data representation. To augment its capacity for 2-D data, we introduce a dynamic multi-path activation mechanism. Importantly, this mechanism, to preserve the structure of the vanilla Mamba block, exclusively operates on the block's input and output. Specifically, we duplicate three copies of the input sequence to establish three different paths, namely the forward path, reverse path, and random shuffle path, and leverage a plain Mamba mixer with shared parameters to model the dependency relationships among tokens within these three sequences, respectively. Subsequently, we revert all tokens in the sequences to the correct order and employ a linear layer to condense sequence information, thereby establishing the gate of the three paths. This gate is then used to activate the representation of the three different information flows as shown in Fig. 1. The process of the ith block is delineated as follows,

$$\begin{aligned} T^i_k &= \Phi^k_{\text{pather}}(T^i) \\ \hat{T}^i_k &= \Phi^\theta_{\text{mixer}}(E^i_k) \\ \hat{T}^i_k &= \Phi^k_{\text{revert-pather}}(\hat{E}^i_k) \\ g &= \Phi_{\text{softmax}}(\Phi_{\text{gate-proj}}(\Phi_{\text{mean}}(\Phi_{\text{cat}}(\{\hat{E}^i_k\})))) \\ T^{i+1} &= \sum\nolimits_{k=0}^{2} g_k \cdot \hat{T}^i_k \end{aligned} \tag{7}$$

where T^i represents the input sequence for the ith layer. $\Phi^k_{\text{pather}}, k \in \{0, 1, 2\}$ denotes the kth sequence path, including the forward path, reverse path, and random shuffle path. $\Phi^\theta_{\text{mixer}}$ is the vanilla Mamba mixer with parameter θ. $\Phi^k_{\text{revert-pather}}$ denotes the operation to revert all tokens to the forward order. Φ_{cat} signifies sequence concatenation with the feature dimension. Φ_{mean} denotes mean pooling along the sequence length dimension. $\Phi_{\text{gate-proj}}$ linearly projects the $3d$ dimension to 3 for sequence information activation. Φ_{softmax} denotes Softmax operation. $\sum$ gathers features from the three different information flows.

D. Model Architecture

The Mamba mixer $\Phi^\theta_{\text{mixer}}$ represents the standard mixer block within the Mamba [13] framework. Drawing upon the principles of ViT, we have developed three distinct versions of RSMamba characterized by different parameter sizes: base, large, and huge. The specific hyperparameters for each version are detailed in Tab. I. Details about the hyperparameter meaning can be found in [13].

TABLE I
THE HYPERPARAMETER SETTINGS FOR DIFFERENT RSMAMBA VERSIONS. N: NUMBER OF BLOCKS, HS: HIDDEN SIZE, IS: INTERMEDIATE SIZE, TSR: TIME STEP RANK, SSMSS: SSM STATE SIZE.

Version	N	HS	IS	TSR	SSMSS
Base	24	192	384	12	16
Large	36	256	512	16	16
Huge	48	320	640	20	16

III. EXPERIMENTAL RESULTS AND ANALYSES

A. Dataset Description

To evaluate the efficacy of the proposed method, we undertook extensive experiments on three distinct remote datasets: UC Merced Land-Use Dataset (UC Merced) [2], AID [1], and NWPU-RESISC45 Dataset (RESISC45) [3]. Each encompasses a unique assortment of categories and image quantities.

UC Merced [2]: The UC Merced is composed of 21 distinct scene categories, with each category containing 100 aerial images of 256 × 256 pixel resolution. The images possess a spatial resolution of 0.3m, culminating in a total of 2100 images. We randomly extracted 70 images from each category for training.

AID [1]: The AID incorporates 30 categories and an aggregate of 10,000 images sourced from Google Earth. The sample quantity varies across different scene types, ranging from 220 to 420. Each aerial image measures 600 × 600 pixels, with spatial resolutions spanning from 8m to 0.5m, thereby encapsulating a multitude of resolution scenarios. We designated 50% of the images from each category as training data.

RESISC45 [3]: The RESISC45 comprises 31,500 remote sensing images obtained from Google Earth, segregated into 45 scene categories. Each category contains 700 RGB images with 256 × 256 pixel resolution. The spatial resolution fluctuates between approximately 30m to 0.2m per pixel. We allocated 70% of the images from each category for training purposes.

B. Implementation Details

In our paper, we employ a fixed input image size of 224 × 224 and implement data augmentation techniques including random cropping, flipping, photometric distortion, mixup, cutMix, *etc*. Images are processed into sequential data

TABLE II
COMPARISONS WITH OTHER METHODS ACROSS DIFFERENT TEST SETS.

Method	Params	UC Merced			AID			RESISC45		
	(M)	P	R	F1	P	R	F1	P	R	F1
ResNet-18 [6]	11.2	87.98	87.46	87.40	88.70	88.17	88.30	88.73	88.44	88.45
ResNet-50 [6]	23.6	91.99	91.74	91.65	89.44	88.66	88.87	92.67	92.47	92.47
ResNet-101 [6]	42.6	92.40	92.22	92.12	91.03	90.63	90.81	92.75	92.57	92.56
DeiT-T [16]	5.5	86.92	86.66	86.53	85.23	84.52	84.52	87.66	86.78	86.79
DeiT-S [16]	21.7	88.95	88.41	88.41	85.88	85.19	85.34	88.21	87.47	87.43
DeiT-B [16]	85.8	89.14	88.73	88.70	87.32	86.07	86.07	89.04	88.62	88.65
ViT-B [7]	88.3	91.09	90.79	90.77	89.39	88.65	88.86	88.84	88.65	88.62
ViT-L [7]	303.0	91.98	91.32	91.26	90.19	88.86	89.17	91.22	91.08	91.04
Swin-T [8]	27.5	90.87	90.63	90.40	86.49	85.66	85.77	90.15	90.06	90.06
Swin-S [8]	48.9	91.08	90.95	90.82	87.50	86.80	86.89	92.05	91.88	91.84
Swin-B [8]	86.8	91.85	91.74	91.62	89.84	89.01	89.07	93.63	91.58	93.56
Vim-Ti† [14]	7.0	89.06	88.73	88.68	87.76	86.98	87.13	89.24	89.02	88.97
VMamba-T [15]	30.0	93.14	92.85	92.81	91.59	90.94	91.10	93.97	93.96	93.94
RSMamba-B (Ours)	6.4	94.14	93.97	93.88	92.02	91.53	91.66	94.87	94.87	94.84
RSMamba-L (Ours)	16.2	95.03	94.76	94.74	92.31	91.75	91.90	95.03	95.05	95.02
RSMamba-H (Ours)	33.1	**95.47**	**95.23**	**95.25**	**92.97**	**92.51**	**92.63**	**95.22**	**95.19**	**95.18**

through a two-dimensional convolution with a kernel size of 16 ($k = 16$) and a stride of 8 ($s = 8$). Position encodings are represented by randomly initialized learnable parameters. For supervised training, we employ the cross-entropy loss function and utilize the AdamW optimizer with an initial learning rate of $5e - 4$ and a weight decay of 0.05. The learning rate is decayed using a cosine annealing scheduler with a linear warmup. The batch size for training is set at 1024, and the training process spans a total of 500 epochs. We employ Precision (P), Recall (R), and F1-score (F1) as performance metrics.

C. Comparison with the State-of-the-Art

We compare our proposed RSMamba with other prevalent deep learning methods for image classification, including the ResNet [6] series underpinned by CNN architecture, and the DeiT [16], ViT [7], and Swin Transformer [8] series, all of which are grounded in Transformer architecture. The comparative classification performance of these methods across the UC Merced, AID, and RESISC45 datasets is presented in Tab. II. The experimental results reveal that: i) RSMamba exhibits robust performance across datasets of varying sizes, with its efficacy being minimally impacted by the volume of training data. This could be attributed to its relatively fewer parameters, negating the need for extensive data for inductive bias. ii) An increase in the depth and width of RSMamba contributes to a performance enhancement across the three datasets. However, the rate of improvement is less pronounced compared to the ResNet and Transformer series. This could be because the base version of RSMamba has already achieved a high degree of accuracy relative to other methods, suggesting that the base version could be a viable starting point for other application tasks. iii) Our experiments also indicate that while CNN architectures converge readily, the superior performance of Transformer architectures hinges on the induction and bias of general features across large-scale training data. In contrast, RSMamba's performance does not rely on extensive data accumulation, but a longer training duration can further lead to substantial performance gains.

D. Ablation Study

To verify the effectiveness of each component, ablation experiments were conducted on the AID dataset. Unless explicitly stated, the base version of the model was utilized, with no modifications made to the associated hyperparameters.

1) Effect of Class Tokens: To obtain dense semantic features for classification, we leveraged mean pooling in RSMamba to amalgamate global information, as opposed to using class tokens akin to ViT [7]. Tab. III delineates the effect of incorporating class tokens at varying positions and mean pooling on the classification performance. The experimental findings indicate that the insertion of class tokens at the head, tail, or both does not yield superior performance. However, insertion in the middle of the sequence can result in a substantial enhancement in performance. Moreover, mean pooling on the sequence can exhibit optimal performance. These observations suggest that the direction of information flow in Mamba significantly influences performance. Concurrently, it was observed during the experiment that mean pooling can expedite the network's convergence.

2) Effect of Multiple Scanning Paths: The vanilla Mamba, derived from modeling causal sequences, poses a significant challenge applying to two-dimensional image data devoid of causal relationships. To address this issue, we propose the multiple scanning path mechanism, *i.e.*, forward, reverse, and random shuffling. To fuse the information flow from these diverse paths, the most straightforward method would be averaging. However, our objective is to adaptively activate the information derived from each path. Consequently,

TABLE III
EFFECT OF CLASS TOKENS AND MEAN POOLING ON PERFORMANCE.

Head	Tail	Middle	Mean Pooling	P	R	F1
✓				87.71	86.71	86.92
	✓			88.68	87.58	87.74
✓	✓			87.92	86.35	86.79
		✓		91.63	91.19	91.24
			✓	92.02	91.53	91.66

TABLE IV
EFFECT OF DIFFERENT SCANNING PATHS ON PERFORMANCE.

Forward	Reverse	Shuffle	Mean/Gate	P	R	F1
✓			-	88.14	87.11	87.24
✓	✓		Mean	89.55	88.45	88.61
✓	✓		Gate	90.83	89.87	90.07
✓	✓	✓	Gate	92.02	91.53	91.66

we have designed a gate to regulate the information flow from the various paths. Tab. IV illustrates the performance enhancements achieved through these designs. An increase in the number of paths correlates with an improvement in classification effectiveness. The gating mechanism also offers certain advantages over feature averaging. It is important to note that we utilized average pooling features for classification in this instance. If we were to adopt a ViT-like class token design, the absence of a multi-path scheme would lead to a substantial decline in performance.

3) Effect of Positional Encoding: To enhance RSMamba with the capacity to model relative spatial relationships, we incorporate position encoding into the flattened image sequence. Tab. V delineates the influence of the presence, absence, and type of position encoding on the classification performance. The lack of position encoding leads to a degradation in performance, whereas both Fourier encoding and learnable encoding contribute to performance enhancements. It should be noted that, given RSMamba's ability to restore the tokens of different paths to their original order, the impact of the presence or absence of position encoding is somewhat mitigated. However, the integration of position encoding can still yield a slight incremental improvement.

4) Effect of the Number of Tokens: RSMamba's proficient capability in global feature abstraction significantly alleviates the complications associated with the length of tokens. As a result, in this paper, we employ an overlapping image patch division method. Tab. V elucidates the effects of the presence or absence of overlap, as well as the enlargement of image size. The division of image patches with overlap allows each token to encapsulate more exhaustive information, thereby leading to an enhancement in performance. Augmenting the image size facilitates the inclusion of more details, which correspondingly yields substantial performance gains. The linear modeling complexity employed by SSM enables a considerable increase in sequence length, even under conditions constrained by resources.

IV. DISCUSSION AND CONCLUSION

In this paper, we introduce a novel state space model for remote sensing image classification, referred to as RSMamba. RSMamba concurrently harnesses the advantages of CNNs and Transformers, specifically their linear complexity and global receptive field. We introduce a dynamic multi-path activation mechanism to alleviate the limitations of unidirectional modeling and position insensitivity inherent in the vanilla Mamba. RSMamba maintains the internal structure of the Mamba and offers the flexibility to easily expand parameters to accommodate various application scenarios. Experimental evaluations conducted on three distinct remote sensing image classification datasets demonstrate that RSMamba can outperform other state-of-the-art classification methods based on CNN and Transformer. Consequently, RSMamba exhibits considerable potential to serve as the backbone network for next-generation visual foundation models.

TABLE V
EFFECT OF POSITIONAL ENCODING.

Design		P	R	F1
PE	None	90.64	90.22	90.25
	Fourier	91.62	90.85	91.04
	Learnable	92.02	91.53	91.66
Token	224^2px, $k = 16, s = 16$	89.93	89.31	89.38
	224^2px, $k = 16, s = 8$	92.02	91.53	91.66
	384^2px, $k = 16, s = 8$	92.75	91.98	92.16

REFERENCES

[1] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[2] Y. Yang and S. Newsam, "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*, 2010, pp. 270–279.

[3] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[4] K. Chen, W. Li, J. Chen, Z. Zou, and Z. Shi, "Resolution-agnostic remote sensing scene classification with implicit neural representations," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2022.

[5] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, 2018.

[6] 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.

[7] 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.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[9] K. Xu, P. Deng, and H. Huang, "Vision transformer: An excellent teacher for guiding small networks in remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[10] J. Chen, K. Chen, H. Chen, W. Li, Z. Zou, and Z. Shi, "Contrastive learning for fine-grained ship classification in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[13] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[14] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[15] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "Vmamba: Visual state space model," *arXiv preprint arXiv:2401.10166*, 2024.

[16] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.