Masked Auto-Encoding Spectral-Spatial Transformer for Hyperspectral Image Classification

Damian Ibañez, Ruben Fernandez-Beltran, *Senior Member, IEEE,* Filiberto Pla and Naoto Yokoya, *Member, IEEE.*

***Abstract*—Deep learning has certainly become the dominant trend in hyper-spectral (HS) remote sensing image classification owing to its excellent capabilities to extract highly discriminating spectral-spatial features. In this context, transformer networks have recently shown prominent results in distinguishing even the most subtle spectral differences because of their potential to characterize sequential spectral data. Nonetheless, many complexities affecting HS remote sensing data (e.g. atmospheric effects, thermal noise, quantization noise, etc.) may severely undermine such potential since no mode of relieving noisy feature patterns has still been developed within transformer networks. To address the problem, this paper presents a novel masked auto-encoding spectral-spatial transformer (MAEST), which gathers two different collaborative branches: (i) a reconstruction path, which dynamically uncovers the most robust encoding features based on a masking auto-encoding strategy; and (ii) a classification path, which embeds these features onto a transformer network to classify the data focusing on the features that better reconstruct the input. Unlike other existing models, this novel design pursues to learn refined transformer features considering the aforementioned complexities of the HS remote sensing image domain. The experimental comparison, including several state-of-the-art methods and benchmark datasets, shows the superior results obtained by MAEST. The codes of this paper will be available at https://github.com/ibanezfd/MAEST.**

***Index Terms*—Hyper-spectral imaging (HS), Vision Transformers (ViT), Mask Auto-Encoders (MAE).**

I. Introduction

OVER the past years, hyper-spectral imaging (HS) has certainly emerged as one of the most important technologies for collecting valuable information from the Earth surface [1]. In this way, numerous remote sensing applications take full advantage of HS imagery to cope with different challenges and societal needs, such as accurate material identification [2]–[4], precision agriculture [5]–[7], target detection [8]–[10] or environmental management [11]–[13] among others. In general, HS instruments are able to acquire data using hundreds of spectral bands in order to provide detailed information across the electromagnetic spectrum, which logically becomes highly beneficial for fine-grained land cover classification [14]. In contrast to other data, HS images allow distinguishing even the most subtle spectral differences among similar pixels that certainly help for the accurate discrimination of remote sensing image components beyond sensor spatial limits whatsoever [15]. Nonetheless, the substantially higher spectral complexities of the HS image domain often bring important difficulties in successfully exploiting such potential capabilities.

Within the remote sensing field, multiple technologies have been proposed for building effective HS image classifiers based on different features and classification paradigms [16]–[18]. Whereas traditional pattern recognition and machine learning approaches are typically constrained by the use of hand-crafted features, deep learning-based methods [19]–[21] have certainly become the dominant trend owing to the superior capabilities of artificial neural networks (ANNs) to extract highly discriminating spectral-spatial features from HS data. The rapid and successful development of numerous backbone networks for HS remote sensing image classification exemplifies this fact. Since Chen *et al.* [22] first proposed using stack auto-encoders (AEs) for characterizing HS images, extensive research has been conducted to generate more accurate classifiers. In [23], the authors were able to improve their former results by means of convolutional neural networks (CNNs). Hang *et al.* [24] took advantage of recurrent neural networks (RNNs) to further exploit the relationships among neighboring HS bands. Zhu *et al.* [25] designed a generative adversarial network (GAN) to effectively classify HS data based on the use of random input noise. Paoletti *et al.* [26] also presented a new HS capsule unit (CapsNet) in order to achieve prominent results while reducing the computational complexity. Besides, Hong *et al.* [27] created a new graph convolutional network (GCN) to address the problems related to large graphs when managing HS data.

Despite the remarkable results provided by these and other relevant networks, their limited ability to characterize sequential spectral data generally made these models unable to really exploit mid-term and long-term dependencies along the spectral dimension [28]. In fact, it was not until recently that some authors were able to address these limitations by using the so-called tranformers [29]. In [30], Dosovitskiy *et al.* proposed the Vision Transformer (ViT) for image classification. In brief, ViT makes use of a positional embedding to sequentially process image patches via the well-known self-attention transformer encoder. More recently, Hong *et al.* [31] implemented a novel transformer network, termed as SpectralFormer (SF), which went a step further by taking also advantage of spatial contexts and long-term skip connections.

This work was supported by Ministerio de Ciencia e Innovación (PID2021-128794OB-I00). *(Corresponding author: R. Fernandez-Beltran.)*

D. Ibañez and F. Pla are with the Institute of New Imaging Technologies, University Jaume I, E-12071 Castellón de la Plana, Spain. (e-mail: ibanezd@uji.es; pla@uji.es).

R. Fernandez-Beltran is with the Department of Computer Science and Systems, University of Murcia, 30100 Murcia, Spain. (e-mail: rufernan@um.es).

N. Yokoya is with the Graduate School of Frontier Sciences, The University of Tokyo, Chiba 277-8561, Japan, and also with RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: yokoya@k.u-tokyo.ac.jp).

With these improvements, SF has become state-of-the-art in HS remote sensing image classification. However, there are some intrinsic data complexities that still remain unsolved when classifying HS imagery from a transformer-based perspective.

In the context of HS remote sensing imaging, one of the main intricacies is related to the noise present in the acquired data [32]. That is, air-borne and space-borne HS instruments aim at measuring the light reflected by the Earth's surface with an exceptional spectral resolution, but in such a goal, many factors may deteriorate the electromagnetic signal falling onto just a few manometers of spectral width. Without a doubt, atmospheric effects together with thermal, quantization, and shot noise are all important factors that may easily affect some pixels of HS remote sensing images [33]. Nonetheless, the most accurate classification technology based on transformers is still unable to take this fact into account inside its own network topology. In other words, although some state-of-the-art models like ViT and SF are able to exploit subtle spectral discrepancies [30], [31], the robustness of such information may be compromised since no mode of relieving noisy patterns is adopted within the own transformer backbone network. Precisely, this is the research gap that motivates this work.

Unlike regular noise reduction methods that can be considered as the first pre-processing step of the standard HS remote sensing classification pipeline [33], this work pursues a different target based on expanding the concept of transformers to promote a more robust feature learning process inside the own architecture. Following this objective, we rethink the way transformers work with HS data by taking advantage of auto-encoding networks [34]. Specifically, we are interested in the denoising capabilities of mask auto-encoders (MAE) [35] and how they can be effectively embedded into a new transformer network specifically designed for HS remote sensing image classification. In response, we propose a novel masked auto-encoding spectral-spatial transformer (MAEST) which gathers two different collaborative branches: (i) a reconstruction path and (ii) a classification path. On the one hand, the reconstruction path adopts an auto-encoding transformer shape which aims at deactivating some pixels from the input HS data while trying to recover such masked data with the objective of identifying the most robust encoding features. On the other hand, the classification path embeds these features onto a transformer network in order to learn how to classify the complete data while focusing on the features that better reconstruct the input. With this novel design, MAEST pursues to uncover refined encoding features to achieve more accurate predictions and faster convergences with respect to existing transformers. For validating the proposed approach, a comprehensive experimental comparison is conducted using three benchmark datasets and different state-of-the-art classification models. Therefore, the key contributions of this paper can be summarized as follows:

- We propose a new transformer network (MAEST) specially designed to classify HS remote sensing data. The presented model aims at exploiting auto-encoding denoising capabilities inside the transformer encoder to relieve HS intrinsic noise.
- We explore the proposed approach performance under multiple configurations in order to provide insights about the working mechanisms and benefits of MAEST with respect to other state-of-the-art classifiers available in the literature.

The remaining parts of this paper are organized as follows. Section II inspects some related works while analyzing their main limitations. Section III describes the proposed classification architecture. Section IV presents and discusses the conducted experiments and studies. Finally, Section V provides the conclusions of this work with an outlook on future research lines.

II. Related Work

A. Hyperspectral Image Classification

Even though the classification of HS imagery is a difficult task due to the limited labeled data, the high number of spectral bands, the similarity between spectral signatures between close related classes and the spectral noise among other concerns, plenty of models have obtained rather good results. Starting from conventional classifiers like K-nearest neighbors (KNN) [36], support vector machines (SVM) [37], or random forest (RF) [38]; to the advances in recent years thanks to deep learning (DL). The emergence of CNNs resulted in a relevant improvement in classification accuracy [39]. In [40] Lee *et al.* used a contextual CNN with multi-scale convolutional filters and a joint spectral-spatial feature map to improve previous CNN classification results. More recently, Cao *et al.* [41] proposed a CNN approach with active learning and a Markov random field optimization, reducing the labeling cost and increasing the HS image classification accuracy. Emerged from CNNs, RNNs which instead of learning from spatial features as 2D-CNNs, use sequential spectral information of the pixel signatures to categorize them also demonstrating effectiveness in HS image classification [42]. Other DL architectures also showed compelling results. For example, despite their computational cost GANs [43] have a higher generalization ability than CNNs. Also GCNs, where Hong *et al.* [27] designed a minibatch GCN (miniGCN) to solve the problems generated by extensive graphs in GCNs for HS imagery. However, these DL classification models usually are unable to properly represent spectral differences between similar pixels and find semantic relations between pixels or patches globally. In this scene, the transformer architecture seems to be a strong competitor, able to acknowledge these obstacles.

B. Transformers

Transformers are the result of the explosion in machine learning due to the use of the self-attention mechanism, not only in previous existing models as CNNs, but also in entirely new architectures. First in natural language processing (NLP) [44], and now in several image processing fields with the introduction of the ViT [30]. In contrast with classic CNNs [41], GANs [43] or GCNs [27], transformers analyze and process the data in a sequential way. In the ViT, the input data is forwarded through several transformer encoders. The

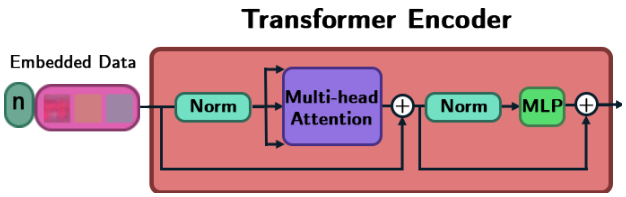


Fig. 1. Transformer Encoder from ViT architecture. The embedded data in the position n is sequentially forwarded through normalization layers (Norm), a multi-head attention block and a fully connected layer (MLP) with residual skip connections.

transformer encoder architecture is shown in Figure 1. First, image patches are embedded with positional information and fed to a multi-head attention block. In this block, the most relevant features of an embedded patch, and its global relations with others are enhanced. As opposed to CNNs and GANs, thanks to the model knowledge of the relative position of all the input sequences, transformers do not rely on past hidden states and recurrence to capture and model middle and long-range dependencies as well. This is specially important in HS imagery because there is a limitation in the classification of spectral pixel signatures using spatial information (CNNs), as relations between spectral bands inside the pixel and might contain more information in a similar fashion when there is a number of categories of materials with similar spectral absorption. In the case of RNNs, even though they can learn sequentially ordered spectral information, this also obscures global long-range dependencies between pixels and spectral signatures. Besides, RNNs are not very efficient in real-world applications. However, directly applying ViT to HS image classification tasks also has some drawbacks, including the loss of local and semantic information. In this context, the SpectralFormer [31] was able to use spectral information from groups of bands with skip connections (SC) to maintain the semantic features. Yet, other issues as the high spectral noise in HS imagery significantly affect sequential learning. Therefore, a higher generalization capability and robustness are required in HS image classification. For this kind of feature extraction, in image processing, it is common to use AEs [45].

C. Auto-encoding

The technique of sequentially using an encoder, which generates a latent representation space from the input, and a decoder that uses this representation to reconstruct the input information is called auto-encoding [46]. There are several uses for AEs like data compression [47], unmixing [48] or data generation [49]. AEs have also been applied in the remote sensing field as well, in anomaly detection [50] or even feature extraction for classification [45] [51]. Another common use for AEs is denoising [52] [53]. Denoising AEs (DAEs) learn to reconstruct the original data from a corrupted input. Some of these DAEs use masked pixels [35] as corruption in the input. Later, DAEs were used for robust feature extraction [53], and then from this idea, MAEs for image reconstruction and pre-training were designed.

D. Masked image reconstruction

Masked image reconstruction methods use corrupted images as inputs, learn their latent representations, and perform a reconstruction of the uncorrupted image. Starting from DAEs [53] and thanks to the success of transformers in NLP [44], from the ViT [30] masked patch reconstruction in a self-supervised fashion for feature extraction has been widely used. A great example is BEiT [54], which uses masked tokenized patches of the image to predict it, learning to extract general and robust representations as a pre-training for ViT. In [35] Chen *et al.* used masked and auto-regressive pixel prediction to improve the learned deep representations as well. He *et al.* [55] used a self-supervised asymmetric transformer MAE to reconstruct masked patches from a limited amount of unmasked data.

E. Self-supervised learning

In self-supervised learning, unlabeled or pseudo-labeled data can be exploited in a supervised manner. Self-supervised learning as a semi-supervision technique has been explored for a long time [56][57], and later in many fields [58] [59] [60]. Though pseudo-labeling has proven to be an effective self-supervised learning method for CNNs [61], it still requires generating fabricated labels. In addition, self-supervised learning in AEs allows to use completely unlabeled data to learn deep representations. In HS imagery this is critical as a result of the small percent of labeled data. In response, inspired by the MAE [55] and the SF [31] in this work we propose the MAEST, a two-branched model with a self-supervised spectral adapted transformer with a masked reconstruction auto-encoder to learn generalizing, noise-free representations from HS imagery to later classify them.

III. Methodology

The purpose of this work is to introduce the Masked Auto-Encoder Spectral-Spatial Transformer (MAEST), a new general hyperspectral classification pipeline based on the success of MAE and ViT architectures in computer vision tasks, focused on learning robust features from highly noisy spectral information. The general architecture is organized into two connected pipelines: the reconstruction pipeline and the classification pipeline. The general strategy of the MAEST is described in Section III-A. These pipelines, are composed of three main blocks: the Reconstruction Encoder (RE) Section III-B and the Reconstruction Decoder (RD) Section III-C for the first one, and the Classification Encoder (CE) Section III-D in the second. Each of these blocks is designed to specifically learn information from HS image data through specialized modules. A general scheme of the complete MAEST is given in Figure 2.

A. MAEST Strategy

The MAEST hyperspectral image classification approach consists of two interconnected pipelines. In the first one, a masked self-supervised auto-encoder learns to reconstruct the

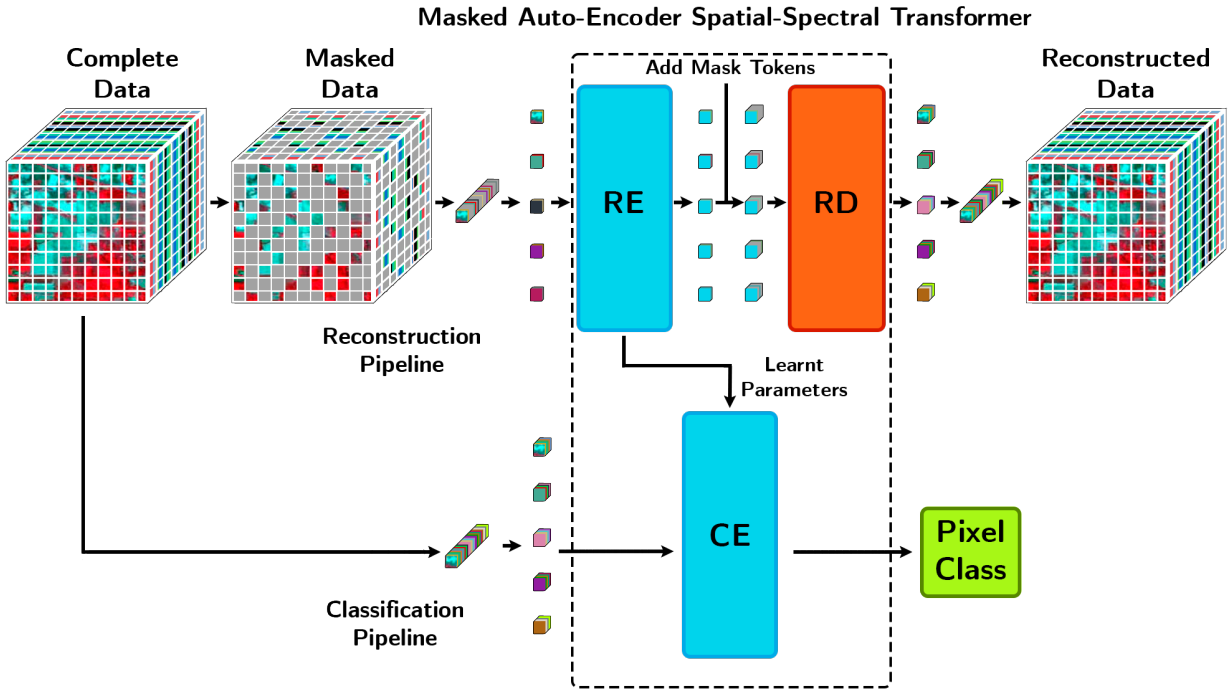


Fig. 2. General scheme of the MAEST, including the reconstruction and classification pipelines.

original HS image, given a partial observation. This auto-encoder is made of two blocks: the RE, an encoder which extracts a latent representation for unmasked segments of the spectral signature of each pixel; and the RD, a decoder that reconstructs the masked data from this latent representation. During this process, the RE learns to extract robust spectral-spatial features from the masked HS image, eliminating spectral noise and redundancy. Masking a high percentage of the input data completely changes the easily solved reconstruction task to a challenging process, where robust and generalizing features to represent the small subset of available information are required. Thus, masking the HS image data in random spatial pixels or patches, and groups of spectral bands prevent a possible location bias towards the image reconstruction. In addition, it allows to use the encoder efficiently, as it only uses a percentage of the training data. Also, to train the RE and RD we use the unlabeled training data in a self-supervised way. To this end, a simple Mean Squared Error reconstruction loss $\mathcal{L}_r$ (Equation (1)) is used

$$\mathcal{L}_r = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{1}$$

where N is the number of sampled values and $\hat{y}_i$ and y_i are ith output and expected ith output respectively.

The second pipeline is responsible for the supervised classification. In this branch, the complete labeled training data is used as input by a single block, the CE. This encoder exploits the robust feature extraction learned parameters in the reconstruction pipeline to categorize pixels after a short fine-tuning of the encoder and the classification layer. For this training, we chose the Cross Entropy loss $\mathcal{L}_c$, which is defined as

$$\mathcal{L}_c = -\sum_{c=1}^{C} y_c \log \hat{y}_c, \tag{2}$$

where $\hat{y}_c$ is the network output, y_c is the expected class, where c and i are the class indices ranging from $[0, C)$ for a C number of classes.

B. Reconstruction Encoder (RE)

The RE block has the objective of learning robust spectral-spatial features from the masked HS image, to then feed these features to the RD and provide guided weights to the CE. The RE, as well as all the blocks composing the complete MAEST are based on the ViT backbone architecture with some adaptations for HS images. Due to HS images containing hundreds of spectral bands, the information along the electromagnetic spectrum from HS images suggests a more continuous behavior than RGB images from the visible spectrum. Besides, the information to classify many interesting categories in RS lies in intervals or groups of spectral bands, as characteristic absorption wavelengths. To take advantage of the spectral information, instead of using image patches as input i.e. ViT, in [31] Hong *et al.* introduced the Groupwise Spectral Embedding (GSE), which uses groups of spectral bands along a spatial patch or pixel as input. Being $\mathbf{X} \in \mathbb{R}^{(H\times W\times M)}$ a HS image of height H, width W and M spectral bands, and $\mathbf{p} = [p_1, p_2, ..., p_M] \in \mathbb{R}^{(1\times M)}$ a single pixel along the spectral dimension of $\mathbf{X}$, the embedded features $\mathbf{E} \in \mathbb{R}^{(D\times M)}$ are obtained by the standard ViT:

$$\mathbf{E} = \mathbf{wp}, \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^{(D\times 1)}$ is the linear projection used for every band. In the case of the GSE, the embedded features are generated from local spectral groups which are defined as $\mathbf{p}_G = f(\mathbf{p}) = [p_1, ..., p_Q, ..., p_M]$, where the function $f()$ generates the overlapping groups of bands and $\mathbf{p}_G \in \mathbb{R}^{(N\times M)}$. With N being the number of bands for each group, $\mathbf{p}^Q = [p_{Q-(N/2)}, \cdots, p_Q, ..., p_{Q+(N/2)}]^T \in \mathbb{R}^{(N\times 1)}$. Equation (4) shows the definition of the embedded features generated by the GSE:

$$\mathbf{E}_G = \mathbf{W}\mathbf{p}_G, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{(D\times N)}$ are the grouped linear projection for each group $\mathbf{p}_G$ generating the embedded dimensions $\mathbf{E}_G \in \mathbb{R}^{(D\times M)}$. However, in the RE we use a modified version of the GSE, the Masked Groupwise Spectral Embedding (MGSE). Instead of using as input for the encoders the complete set of embedded feature groups $\mathbf{E}$, once the groups are projected we add a learnable positional encoding to the feature groups, and randomly mask a K percentage of them. The masking is done after the projection in order to exploit the $\mathbf{W}$ weights in the CE as well. Due to the electromagnetic spectrum of an HS image containing sampled bands each few nm, even though this provides more information of the properties of specific materials, it also includes noise and irrelevant information. The RE uses the MGSE to learn features that are robust to this noise, improving the classification results and performance. The learnable absolute positional embedding also retains the information of the location of each group in the complete pixel spectrum, even after masking a high percentage of the data. Then, MGSE embedded features are modeled as

$$\mathbf{E}_M = g(\mathbf{s}_G + \mathbf{W}\mathbf{p}_G). \tag{5}$$

In this case, $\mathbf{s}_G \in \mathbb{R}^{(D\times M)}$ are the absolute positional embedding, and $g()$ is a function which masks a k percentage of the groups and withdraws them from the input groups. The resulting unmasked embedded dimensions, $\mathbf{E}_M \in \mathbb{R}^{(D\times K)}$ where $K = (k-1)\%N$ are used as input. To take advantage of spatial information, the MAEST can also take spatial patches $\mathbf{P} \in \mathbb{R}^{(h\times w\times K)}$ with the patch height and width dimensions being h and w respectively, are flattened and the embedded features are calculated similarly with each group being of size $\mathbf{p}_Q \in \mathbb{R}^{(Nhw\times 1)}$. In this way, not only the spectral, but the spatial context for the pixel are known in the learning process. For simplicity, the pixel example is used in the following sections. Later, the embedded information is forwarded to a sequence of classic transformer encoders which generate the output reconstruction latent space $\mathbf{L}_E \in \mathbb{R}^{(D_E\times K)}$ of D_E dimensions. The use of only a small percentage of the spectral groups makes the learning of features a low and efficient computation process. The complete scheme of the RE is portrayed in Figure 3.

C. Reconstruction Decoder (RD)

The RD block's goal is to use the latent space features extracted by the RE to reconstruct the masked spectral-spatial information of the original HS image. Once the RE has generated the reconstruction latent space $\mathbf{L}_E$, a fully connected layer is used as a bridge to the RD input size

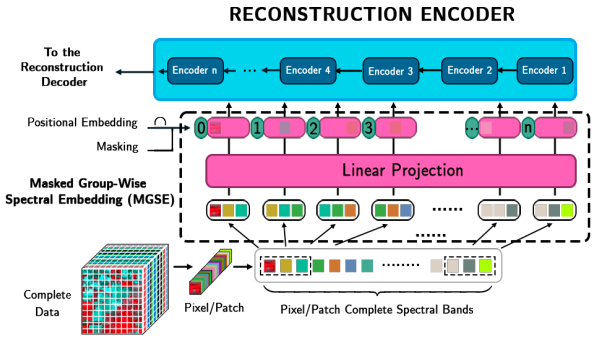


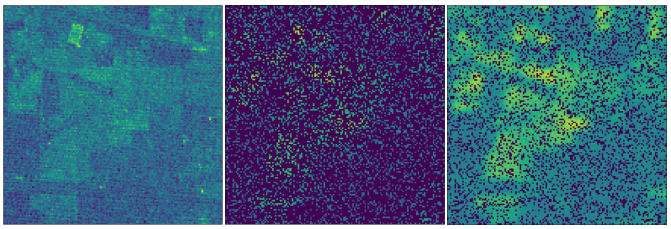
Fig. 3. Detailed diagram of the Reconstruction Encoder.

latent space $\mathbf{L}_D \in \mathbb{R}^{(D_D\times K)}$ of D_D dimensions. Then, in the position of the missing masked information, masked tokens are placed. After the masked tokens incorporation, a position embedding is added to the whole new set $\mathbf{L}_M \in \mathbb{R}^{(D_D\times N)}$, therefore including the latent space representations $\mathbf{L}_D$, the masked tokens and the absolute location information of both. Next, the complete set $\mathbf{L}_M$ is used as input by a sequence of transformer decoders. Finally, a fully connected layer is used as a reconstruction layer to predict only the masked pixels groups. We include a visual example of the masked reconstruction results in Figure 4.

D. Classification Encoder (CE)

The last block of the MAEST, the CE is designed for the HS image classification task. The CE takes advantage of the robust spectral feature extraction learned during the HS image reconstruction, and after a short fine-tuning of the inherited parameters is able to accurately classify HS image categories. In this case, the input of the CE is the complete and unmasked training data. In contrast to the RE, the original GSE method is used to create the spectral groups of pixels p_G without masking. From the $\mathbf{p}_G$ groups and the $\mathbf{W}$ projections, the embedded features $\mathbf{E}$ are obtained as in Equation (4). Then, a learnable classification extra token is included in the embedded features, now $\mathbf{E}_C \in \mathbb{R}^{(D\times M+1)}$. This extra classification token is used to assign a category to the complete pixel spectrum. Besides, a positional embedding is included in each of the embedded features and these are forwarded to the sequential modules of encoders in the same way as the RE.

However, in the CE, a SC mechanism defined as Cross-Layer Adaptative Fusion (CAF) [31] is used. This technique



(a) (b) (c)

Fig. 4. False color visualization of the reconstruction done by the patch MAEST for the first band for the Indian Pines dataset: (a) Original, (b) Masked, (c) Reconstructed.

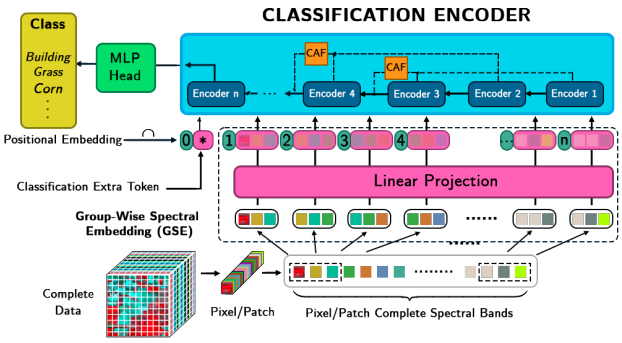


Fig. 5. Detailed diagram of the Classification Encoder.

was inspired by the improvement seen in networks like ResNet [62] and UNet [63] in computer vision tasks using short-range and long-range SC respectively. In the case of CAF, a middle-range SC is used to avoid having a huge difference between features (long-range) and a high computational cost (short-range). Also, in HS image classification, it is usual to design shallow network architectures due to the limited labeled training and test data, making long-range SC of several layers unachievable. Specifically, only one encoder is skipped in each connection. The CAF process can be expressed as

$$\hat{z}^{(i)} \leftarrow \ddot{\mathbf{w}} \begin{bmatrix} z^{(i)} \\ z^{(i-2)} \end{bmatrix}, \tag{6}$$

where $z^{(i)} \in \mathbb{R}^{(1\times D_z)}$ and $z^{(i-2)} \in \mathbb{R}^{(1\times D_z)}$ are the (i)th and $(i-2)$th feature layers of D_z dimensions, respectively, used as input, and $\ddot{\mathbf{w}} \in \mathbb{R}^{(1\times 2)}$ are the learnable adaptive fusion weights used to obtain the new cross-layer representations $\hat{z}$. Finally, an MLP head is used as a classification layer. A detailed diagram of the CE is shown in Figure 5.

IV. Experiments

In this section, extensive experimentation is conducted to measure the effectiveness of the MAEST pre-training for HS classification compared to baseline and state-of-the-art methods. First, three popular HS datasets used through the experimentation are described in Section IV-A, followed by the experimental setting and implementation details in Section IV-B. Then, we evaluate its quantitative and qualitative Section IV-C results for the HS classification task. Finally, we analyze different configurations and training situations for our model Section IV-D.

A. Datasets Description

To study the performance of the proposed MAEST, we use three well-known HS datasets, the Indian Pines dataset, the Pavia University dataset and the Houston2013 dataset.

1) Indian Pines Dataset: The Indian Pines dataset is composed of 145×145 pixels sampled with a resolution of 20 m ground sampling distance (GSD) and 220 spectral bands from 400 to 2500 nm wavelength. The dataset was gathered by the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) HS sensor in 1992 over the Purdue University Agronomy farm northwest of West Lafayette and the surrounding area. After removing the water absorption bands, the number of spectral bands is reduced to 200, specifically the bands [1–103], [109–149], and [164–219] are retained. The Indian Pines dataset is labeled into 16 main classes. We include details of the training and testing data samples for every class in Table I. A visualization of the training and testing data is also included in Figure 6.

TABLE I
Land-cover classes of the Indian Pines dataset, with the standard training and testing sets for each class

Class No.	Class Name	Training	Testing
1	Corn Notill	50	1484
2	Corn Mintill	50	784
3	Corn	50	184
4	Grass Pasture	50	447
5	Grass Trees	50	697
6	Hay Windrowed	50	439
7	Soybean Notill	50	918
8	Soybean Mintill	50	2418
9	Soybean Clean	50	564
10	wheat	50	162
11	Woods	50	1244
12	Building Grass Trees Drives	50	330
13	Stone Steel Towers	50	45
14	Alfalfa	15	39
15	Grass Pasture Mowed	15	11
16	Oats	15	5
	Total	695	9671

TABLE II
Land-cover classes of the Pavia University dataset, with the standard training and testing sets for each class

Class No.	Class Name	Training	Testing
1	Asphalt	548	6304
2	Meadows	540	18146
3	Gravel	592	1815
4	Trees	524	2912
5	Metal Sheets	265	1113
6	Bare Soil	532	4572
7	Bitumen	375	981
8	Bricks	514	3364
9	Shadows	231	795
	Total	3921	40002

2) Pavia University Dataset: The Pavia University dataset contains an image of 610×340 pixels with 1.3m GSD and 103 spectral bands from 430 to 860nm. This image was obtained by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over Pavia University and its surroundings in Pavia, Italy. The Pavia University dataset is classified into 9 different land cover classes. We include details of the training and testing data samples for every class in Table II. A visualization of the training and testing data is also included in Figure 7.

3) Houston2013 Dataset: The Houston2013 dataset consists of a 349×1905 pixels HS image with 144 bands ranging from 364 to 1046 nm. This image contains the area of the campus of the Houston University and its surroundings in Texas, USA, acquired by ITRES CASI-1500 sensor. For our experimentation, we used a cloudless version of the

TABLE III
LAND-COVER CLASSES OF THE HOUSTON2013 DATASET, WITH THE STANDARD TRAINING AND TESTING SETS FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Healthy Grass	198	1053
2	Stressed Grass	190	1064
3	Synthetic Grass	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot1	192	1041
13	Parking Lot2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	Total	2832	12197

Houston2013 dataset [31]. In this version, the image has been processed to remove occlusions and restore lost data. The Houston2013 dataset has 15 different land-cover and land-use classes. We include details of the training and testing data samples for every class in Table III. A visualization of the training and testing data is also included in Figure 8.

B. Experimental Settings

1) Implementation: The implementation of the MAEST model and HS classification were designed and executed using Pytorch in Python 3.6 on a Ubuntu 16.04 x64 machine with Intel(R) Core(TM) i7-6850 K processor with 110 GB RAM with an NVIDIA GeForce GTX 1080 Ti 11 GB GPU. Due to computational limitations, we executed the masked self-supervised reconstruction and the supervised classification separately. Specifically, for the classification training, we used a batch size of 64 with the Adam optimizer. A learning rate of $5e^{-4}$ decaying with a γ factor of 0.9 every $1/10$ of the epochs, which were set to 1000. Nevertheless, both the proposed MAEST and the SpectralFormer converge in much fewer epochs. For the MAEST 300 epochs in the Indian Pines dataset, 400 epochs in Pavia University and as fast as 260 for the Houston dataset. In the case of the SpectralFormer, 300 epochs as well for the Indian Pines dataset, and 500 epochs for the other two datasets.

2) Compared Methods: In the experimentation, we compare the proposed MAEST to many representative classification standard methods, classic backbones and state-of-the-art models. We divided these classifiers into four main groups: conventional classifiers, classic backbone networks, transformer-based models and the proposed MAEST. Below the standard parameter selection for each of the classifiers is included.

1) Conventional Classifiers: In this group we included the KNN [36] SVM [37] and RF [38]. For the KNN we used a number of nearest neighbors of 10, which is the main factor. The SVM was implemented through the libsvm toolbox3, using the radial basis function kernel. This kernel is defined by two main parameters σ and λ. These parameters are optimized using a fivefold cross-validation in training ranging from $\sigma = [2{-}3, 2{-}2, ..., 24]$ and $\lambda = [10{-}2, 10{-}1, ..., 104]$. For the RF main parameter, 200 decision trees were selected.

2) Classic Backbone Networks: In the second group models based on convolutional neural networks are included, 1D-CNN [39], 2D-CNN [41], a RNN [42], and miniGCN [27]. The 1D-CNN consists of a set of 128 1D convolutional filters, a batch normalization (BN) layer, and a ReLU activation layer with a softmax for the classification purpose. The 2D-CNN contains three blocks made of a 2D convolutional layer, a BN layer, max-pooling and finally a ReLU activation layer. The parameters for the convolutional layers are a kernel size of (3×3) for the two first layers and (1×1) for the third. The number of filters for each layer is 32, 64 and 128. Finally, a softmax is added as well. In the RNN case, two recurrent layers with gated recurrent units of 128 neurons were used. And for the miniGCN a block containing a BN layer, a 128 neuron graph convolutional layer and a ReLU layer were used.

3) Transformers: In this third group we included the original ViT [30] and the SpectralFormer [31]. For the original ViT network architecture five encoder blocks were used. Each of them containing 4 SA heads, an MLP with 8 hidden dimensions and a GELU activation layer. A 10% dropout layer is used after the positional embedding and the MLPs. The SF is used with 5 encoders using the same parameters with CAF skipping 1 encoder. For the patch-wise SF a patch size of 7×7 is used along with a weight decay of $5e^{-3}$.

4) MAEST: For the RE and RD we used the same parameters as the ViT, and for the CE the SF parameters, including CAF and patch-wise sizes. The masking percentage for the RE was fixed to a random 75% of the samples for the self-supervised reconstruction. Due to computational limitations, first the masked reconstruction is performed, and later the classification using the learned weights from the reconstruction task.

C. Classification Results

1) Quantitative Results: The classification performances of every presented method and the proposed MAEST are shown in three tables. The Indian Pines dataset results in Table IV, the Pavia University dataset in Table V and in Table VI the Houston2013 dataset results. In the tables three general different metrics are evaluated: the overall accuracy (OA), the average accuracy (AA) and the Kappa coefficient (κ). The accuracy results for each class are shown as well, with the best results shown in bold font for all the metrics. In general, there is a trend in all the tables where the worst results are in the conventional classifiers, followed by the classic backbone networks, then the transformer-based models and finally the best results are obtained by the proposed MAEST. However, some differences between the dataset results can be observed.

TABLE IV
QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA AND K AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE INDIAN PINES DATASET. BEST RESULTS ARE SHOWN IN BOLD.

Class	Conventional Classifiers			Classic Backbone Networks				Transformers			MAEST	
	KNN	SVM	RF	1D-CNN	2D-CNN	RNN	miniGCN	ViT	SF(pixel)	SF(patch)	pixel	patch
1	45.45	67.34	57.80	47.83	65.90	69.00	72.54	41.55	75.07	68.35	68.64	**78.97**
2	46.94	67.86	56.51	42.35	76.66	58.93	55.99	63.39	62.76	79.72	76.91	**92.73**
3	77.72	93.48	80.98	60.87	92.39	77.17	92.93	75.54	92.39	95.11	97.28	**98.91**
4	84.56	94.63	85.68	89.49	93.96	82.33	92.93	68.68	93.29	95.52	93.51	**96.20**
5	80.06	88.52	79.34	92.40	87.23	67.72	**94.98**	81.77	84.79	79.63	87.52	89.10
6	97.49	94.76	95.44	97.04	97.27	89.07	**98.63**	83.60	89.29	99.32	89.98	98.41
7	64.81	73.86	77.56	59.69	77.23	69.06	64.71	71.68	80.50	81.70	84.64	**84.86**
8	48.68	52.07	58.85	65.38	57.03	63.56	68.78	72.08	60.26	66.96	69.44	**73.44**
9	44.33	72.70	62.23	**93.44**	72.87	65.07	69.33	52.84	76.24	65.25	74.11	69.50
10	96.30	98.77	95.06	99.38	**100.00**	95.06	98.77	95.06	98.15	99.38	98.15	**100.00**
11	74.28	86.17	88.75	84.00	92.85	88.67	87.78	92.77	91.00	**93.73**	84.97	92.12
12	15.45	71.82	54.24	86.06	88.18	50.00	50.00	55.45	62.73	84.85	76.36	**91.51**
13	91.11	95.56	97.78	91.11	**100.00**	97.78	**100.00**	97.78	**100.00**	95.56	**100.00**	**100.00**
14	33.33	82.05	56.41	84.62	84.62	66.67	48.72	**89.74**	**89.74**	76.92	94.87	**89.74**
15	81.82	90.91	81.82	**100.00**	**100.00**	81.82	72.73	90.90	90.90	**100.00**	90.90	**100.00**
16	40.00	**100.00**	**100.00**	80.00	**100.00**	**100.00**	80.00	60.00	**100.00**	**100.00**	**100.00**	**100.00**
OA (%)	59.17	72.36	69.80	70.43	75.89	70.66	75.11	69.66	75.69	78.55	78.52	**84.15**
AA (%)	63.90	83.16	76.78	79.60	86.64	76.37	78.08	74.55	84.19	86.37	86.71	**90.97**
κ (%)	0.5395	0.6888	0.6591	0.6642	0.7281	0.6673	0.7164	0.6528	0.7250	0.7566	0.7567	**0.820**

TABLE V
QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA AND K AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE PAVIA UNIVERSITY DATASET. BEST RESULTS ARE SHOWN IN BOLD.

Class	Conventional Classifiers			Classic Backbone Networks				Transformers			MAEST	
	KNN	SVM	RF	1D-CNN	2D-CNN	RNN	miniGCN	ViT	SF(pixel)	SF(patch)	pixel	patch
1	73.86	74.22	79.81	88.90	80.98	84.01	**96.35**	80.17	76.92	79.77	85.45	79.85
2	64.31	52.79	54.90	58.81	81.70	66.95	89.43	63.24	68.59	86.32	77.24	**96.60**
3	55.10	65.45	46.34	73.11	67.99	58.46	**87.01**	65.45	62.37	72.12	68.60	67.66
4	94.95	97.42	**98.73**	82.07	97.36	97.70	94.26	93.92	97.77	97.87	97.49	96.84
5	99.19	99.46	99.01	99.46	99.64	99.10	**99.82**	99.19	96.14	99.73	99.28	99.73
6	65.16	93.48	75.94	**97.92**	97.56	83.18	43.12	90.75	94.38	79.31	90.94	82.70
7	84.30	87.87	78.70	88.07	82.47	83.08	90.96	87.77	90.11	80.83	89.09	**96.23**
8	84.10	89.39	90.22	88.14	**97.62**	89.63	77.42	89.77	92.39	95.65	91.20	95.96
9	98.36	**99.87**	97.99	99.87	95.60	96.48	87.27	99.12	99.62	90.56	99.50	94.47
OA (%)	70.53	70.82	69.67	75.50	86.05	77.13	79.79	75.93	78.60	85.79	83.70	**91.06**
AA (%)	79.68	84.44	80.18	86.26	88.99	84.29	85.07	85.49	86.48	86.91	88.76	**90.00**
κ (%)	0.6268	0.6423	0.6237	0.6948	0.8187	0.7101	0.7367	0.6980	0.7297	0.8123	0.7899	**0.8794**

TABLE VI
QUANTITATIVE PERFORMANCE OF DIFFERENT CLASSIFICATION METHODS IN TERMS OF OA, AA AND K AS WELL AS THE ACCURACIES FOR EACH CLASS ON THE HOUSTON2013 DATASET. BEST RESULTS ARE SHOWN IN BOLD.

Class	Conventional Classifiers			Classic Backbone Networks				Transformers			MAEST	
	KNN	SVM	RF	1D-CNN	2D-CNN	RNN	miniGCN	ViT	SF(pixel)	SF(patch)	pixel	patch
1	83.19	83.00	83.38	87.27	85.09	82.34	**98.39**	83.19	80.82	84.52	85.00	82.90
2	95.68	98.40	98.40	98.21	**99.91**	94.27	92.11	94.64	99.15	99.34	98.40	99.44
3	99.41	99.60	98.02	**100.00**	77.23	99.60	99.60	99.60	99.80	89.31	99.80	94.65
4	97.92	**98.48**	97.54	92.99	97.73	97.54	96.78	94.43	95.74	98.30	96.50	96.02
5	96.12	97.82	96.40	97.35	99.53	93.28	97.73	97.92	98.01	**100.00**	98.39	99.24
6	92.31	90.91	**97.20**	95.10	92.31	95.10	95.10	95.10	95.80	93.71	95.10	93.01
7	80.88	90.39	82.09	77.33	**92.16**	83.77	57.28	82.84	83.12	86.85	88.53	89.93
8	48.62	40.46	40.65	51.38	79.39	56.03	68.09	53.94	54.70	80.82	57.64	**82.15**
9	72.05	41.93	69.78	27.95	**86.31**	72.14	53.92	58.73	67.23	76.11	69.59	77.15
10	53.19	62.64	57.63	**90.83**	43.73	84.17	77.41	58.30	87.45	77.61	89.29	87.55
11	86.24	75.43	76.09	79.32	87.00	82.83	84.91	62.71	87.76	72.39	**88.52**	85.58
12	44.48	60.04	49.39	76.56	66.28	70.61	77.23	52.55	75.50	**81.65**	73.78	80.60
13	28.42	49.47	61.40	69.47	**90.18**	69.12	50.88	65.26	69.47	76.49	71.23	69.82
14	97.57	98.79	99.60	99.19	90.69	98.79	98.38	99.19	98.79	99.19	**100.00**	97.17
15	98.10	97.46	97.67	98.10	77.80	95.98	98.52	**98.31**	98.10	94.93	97.89	98.10
OA (%)	77.30	76.91	77.48	80.04	83.72	83.23	81.71	77.00	84.41	86.44	85.86	**88.55**
AA (%)	78.28	78.99	80.35	82.74	84.35	85.04	83.09	80.12	86.10	87.41	87.31	**88.89**
κ (%)	0.7538	0.7494	0.7564	0.7835	0.8231	0.8183	0.8018	0.7510	0.8311	0.8529	0.8467	**0.8757**

For the Indian Pines dataset case, the overall worst results are obtained by the KNN and RF, with the SVM being the only conventional classifier to obtain competitive results. Among the DL networks, a similar behavior can be seen, where good results are obtained for some of the classes, but lower accuracy for the others. These models seem to have identified properly some classes, but were not able to generalize enough for the other ones. Furthermore, even though the ViT obtained a balanced accuracy it is unable to properly model and learn the pixel spectral signatures to obtain high accuracy for many categories. However, results over an 80% of average accuracy are only seen by the SVM, 2D-CNN, both SF and MAEST versions, with the patch SF, pixel MAEST and patch MAEST obtaining the best results in that order. The improvement brought by the specialized HS techniques introduced by the SF and used in the MAEST can be seen even in the pixel versions of these methods, as those obtain overall a higher accuracy than the 2D-CNN without the use of information. A similar case happens comparing the pixel MAEST and the patch SF. Though the SF can model many local spectral discrepancies, the robust features obtained in the MAEST through the reconstruction and its generalization ability make it able to outperform in almost every class of the other methods, with also good results in the rest of the classes. These results are even more notable in this specific dataset, which contains the lowest number of samples.

In the Pavia University and Houston datasets, the gap between accuracy results for the different methods are not as huge. Nevertheless, a similar tendency can be seen, with the conventional classifiers behind the classic backbone networks, followed by the transformer-based models. We attribute this difference to the Pavia University and Houston datasets having more training data, and being more balanced than the Indian Pines dataset. The MAEST architecture has been specially designed to deal with this lack of information issues in HS imagery, so it is expected that the improvement difference drops in these cases. Still, the patch MAEST is able to obtain the best general results among the studied methods, with the pixel MAEST close showing the generalization and learning skills of the proposed architecture.

2) Qualitative Results: Additionally to the numerical results, we also show the classification maps obtained by each compared method, including a visualization of the training and testing data. The Indian Pines, Pavia University and Houston2013 datasets in Figure 6, Figure 7 and fig. 8 are shown respectively. The qualitative results follow the previously presented numerical values, with the conventional classifiers containing high levels of noise and diffuse bodies. In comparison, the DL models are able to define some edges and shapes which are better identifiable. Still, except for the 2D-CNN a high quantity of salt pepper noise distorts many shapes. Visually, it can be seen how the ViT performs in a similar way, with lots of noise distributed along the image. Both the pixel MAEST and SF show a similar pattern to the ViT, but with less noise, as was expected by their similar architectures showing the improvements from the HS image specialization techniques used. Finally, the patch SF and specially the patch MAEST show a higher skill in defining smooth zones and building shapes with reasonable sharp edges. In the three datasets, it can be seen in the zoomed detail how the MAEST is able to properly define these shapes with also more uniform categorization and texture.

TABLE VII
ANALYSIS OF THE NUMBER OF GROUPED SPECTRAL BANDS IN THE MAEST (MGSE AND GSE) IN THE INDIAN PINES DATASET IN TERMS OF OA, AA AND κ.

MAEST	Metric	Neighboring bands				
		1	3	5	7	9
Pixel	OA(%)	71.40	**78.52**	77.71	78.22	74.23
	AA(%)	78.06	**86.71**	85.15	83.77	84.35
	κ	0.6778	**0.7567**	0.7478	0.7528	0.7113
Patch	OA(%)	82.69	**84.15**	82.31	75.82	75.77
	AA(%)	90.15	**90.97**	89.46	86.67	85.54
	κ	0.8031	**0.820**	0.7991	0.7282	0.7279

D. Model Analysis

In order to evaluate in depth the MAEST performance, robustness and parameters selections we performed a number of experiments. First, we analyze the number of spectral bands used in the GSE and MGSE. Then, we discuss the difference between the pixel and patch MAEST evaluating different patch values. Next, we analyze the MAEST performance by modifying the percentage of masked data and the number of epochs during the reconstruction and the learning rate. Finally, we compare the classification results of the proposed method with the other transformer-based state-of-the-art models in extreme situations of the quantity of training data and number of epochs.

1) MGSE and GSE Number of Bands Analysis: The first parameter studied is the number of spectral neighboring bands in the MGSE and GSE modules. Both of these modules, the MGSE in the reconstruction pipeline and the GSE in the classification pipeline, allow the MAEST to learn spectral features. Table VII shows the HS classification accuracy of the pixel and patch MAEST for the Indian Pines dataset with different numbers of neighboring bands. As shown in the table, the optimal number of bands for both versions of the proposed MAEST is 3 bands. This difference is further appreciable in the pixel version. With only one spectral band the MAEST is unable to take proper advantage of the spectral information. However, with a higher number of neighboring grouped bands, the accuracy drops as well. We consider this relation of lower accuracy with a higher number of grouped bands as a result of a higher spectral noise in each group of bands, which hinders learning the relationship between bands and extracting relevant features.

2) Pixel vs Patch Analysis: As previously stated, in this work we propose two versions of the MAEST: one which only uses the pixel spectral information, and another one which uses the spectral-spatial contextual information surrounding the pixel as well. We defined those as MAEST pixel and patch variations for simplicity. Even though the patch MAEST requires higher computational resources, it is able to obtain significantly better results than the non-spatial version over all

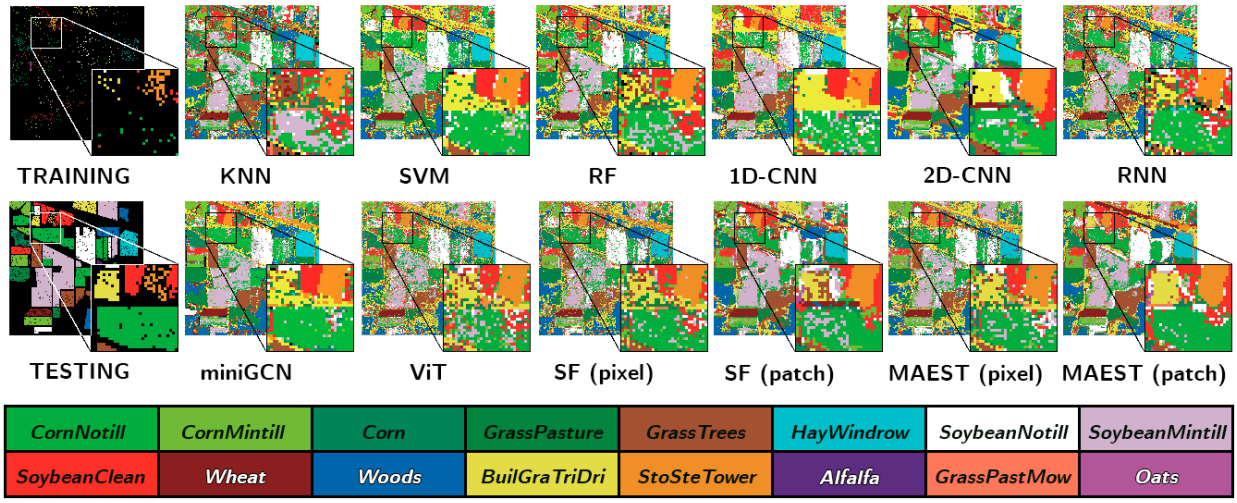


Fig. 6. Spatial distribution of the training and testing sets, and the classification maps generated by different models on the Indian Pines dataset.

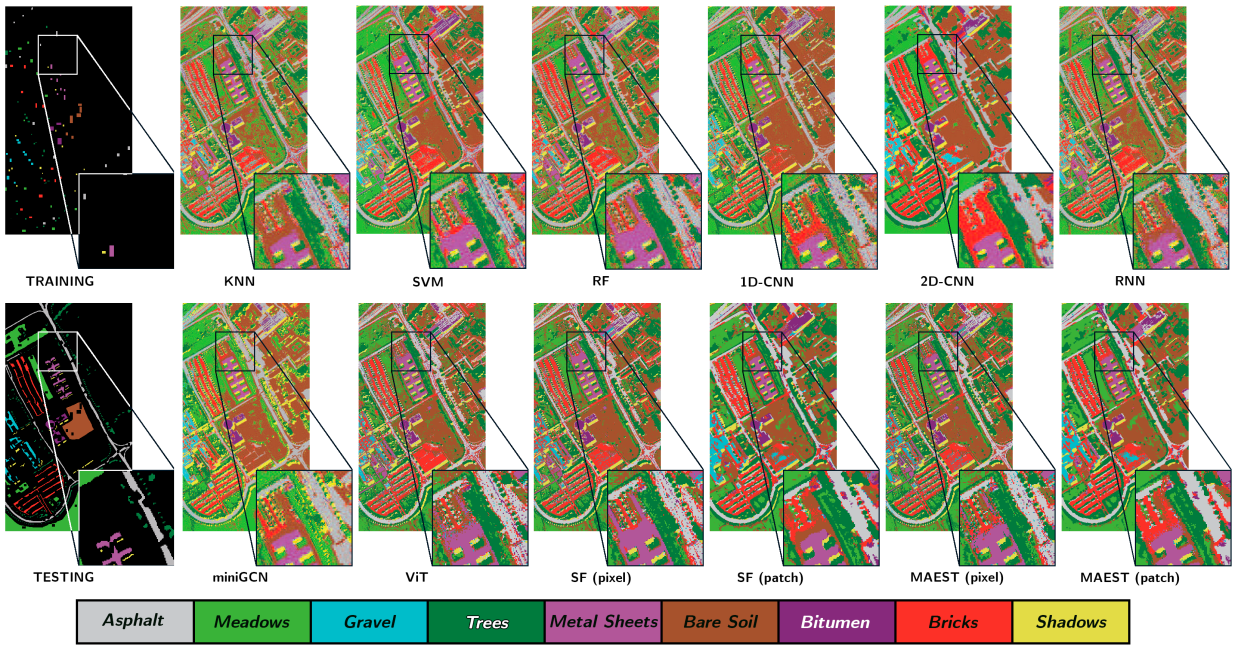


Fig. 7. Spatial distribution of the training and testing sets, and the classification maps generated by different models on the Pavia dataset.

the experimentation. Analyzing the classification results for the pixel and patch versions of both, the proposed MAEST and the SF, it is evident that the use of local contextual spatial information is able to grasp useful semantic spatial relations. The performance of the 2D-CNN compared to the other DL methods also suggest the importance of the use of contiguous information. Even in the visual results (Figure 6, Figure 7, Figure 8) is can be observed a difference along the different datasets, where the 2D-CNN and the patch MAEST and SF barely contain noise compared to 1D models and the pixel versions. For a further analysis, we evaluate the classification accuracy in Indian Pines with different patch sizes in Table VIII. In this table, we can observe how there is a difference of almost 4% of accuracy from the pixel MAEST to use a small patch size of 3×3. The accuracy obtained for the different patch size values is similar between 3×3 and 7×7, with the last one obtaining the best results. For higher values, the patch size contains too much information and the accuracy starts decreasing as expected.

TABLE VIII
ANALYSIS OF DIFFERENT VALUES OF PATCH SIZE FOR THE MAEST IN INDIAN PINES DATASET IN TERMS OF OA, AA AND κ.

Metric	Patch size				
	1	3	5	7	9
OA(%)	78.52	81.97	82.44	**84.15**	81.08
AA(%)	86.71	90.07	89.11	**90.97**	88.78
κ	0.7567	0.7953	0.8008	**0.820**	0.7852

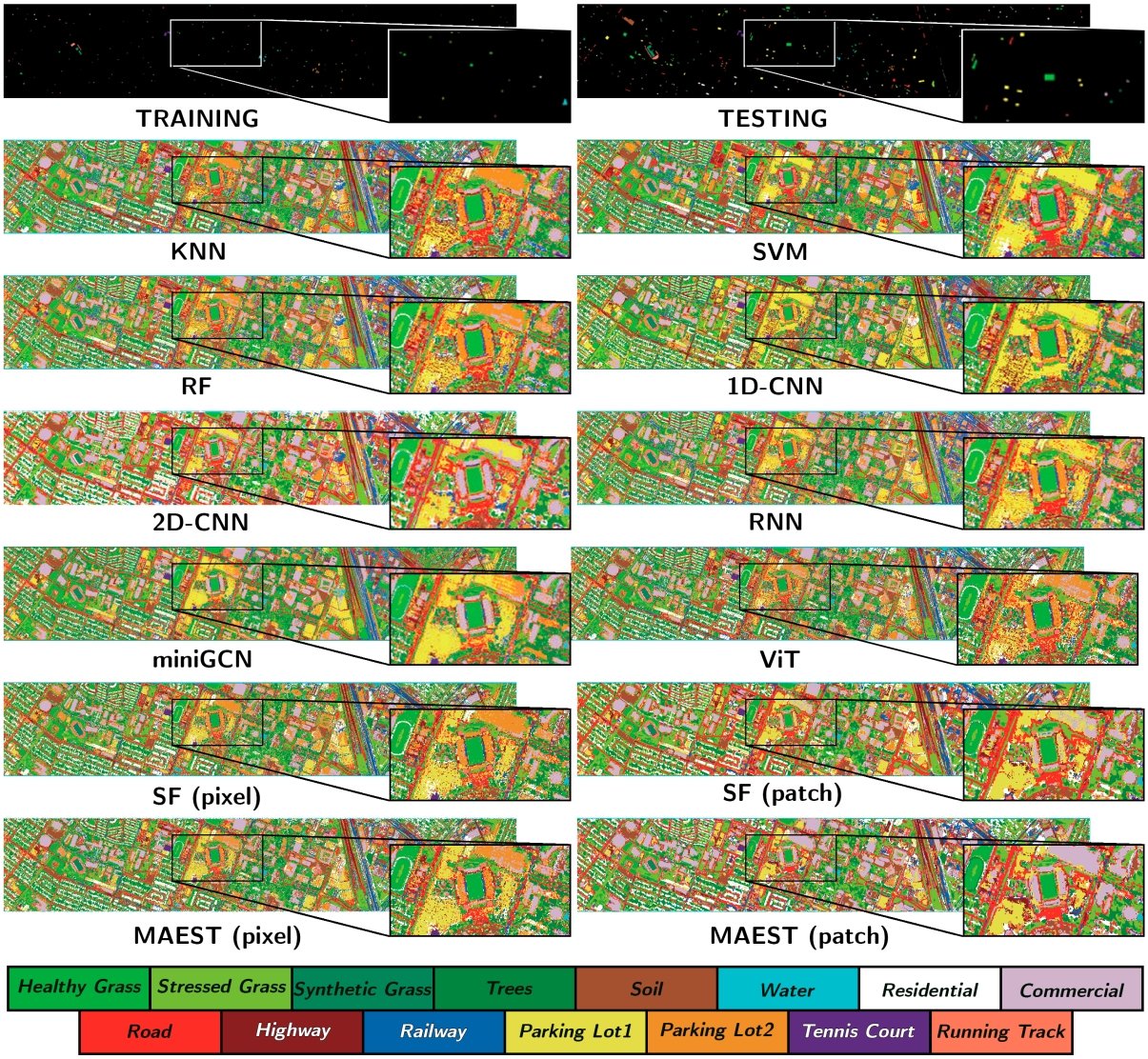


Fig. 8. Spatial distribution of the training and testing sets, and the classification maps generated by different models on the Houston dataset.

3) Reconstruction Masking Analysis: Table IX includes the different AA classification results for the Indian Pines dataset, depending on the percentage of masked data in the reconstruction pipeline. The table shows how increasing the masking ratio up to a pretty high value of 75% of masked data achieves the best classification results. This high masking results are aligned with the work of He *et al.* [55], in contrast with other masked transformer-based models as BERT [44], whose typical masking ratio is 15% and other related computer vision transformers as [30] [54] (from 20% to 50% masking ratio). Nonetheless, it is interesting noting that the major jump in accuracy happens from not using masking at all to a 25%, and it increases much slower from there up to around a 4% accuracy improvement. Moreover, the patch MAEST without masking is able to obtain better results than the patch SF and the other tested methods, proving the usefulness of using the reconstruction pipeline even without masking. We link this behavior to the significant amount of spectral noise found in HS images. A high percentage of masking makes the MAEST able to learn general feature representations robust to this spectral noise.

TABLE IX
ACCURACY ON INDIAN PINES DATASET WITH DIFFERENT PERCENTAGES OF MASKED DATA IN THE RECONSTRUCTION PIPELINE.

Masking %	Pixel AA	Patch AA
0%	82.30	87.37
25%	84.18	89.01
50%	85.32	90.23
75%	**86.71**	**90.97**

4) Learning Rate Analysis: For the proposed MAEST and many other DL networks, one of the most important hyper-parameters is the learning rate. Specially in the MAEST, as it contains two different pipelines which can be trained separately, a learning rate study is essential. Table X shows an AA analysis for different learning rate values in the

TABLE X
ACCURACY (AA) ON INDIAN PINES DATASET WITH DIFFERENT LEARNING RATE VALUES FOR RECONSTRUCTION (R), CLASSIFICATION (C) AND BOTH (R+C).

MAEST	Learning rate				
	10^{-3}	$5*10^{-3}$	10^{-4}	$5*10^{-4}$	10^{-5}
R	88.40	84.09	87.03	**90.97**	83.77
C	88.68	76.87	88.09	**90.97**	46.04
R+C	84.88	78.15	85.16	**90.97**	36.95

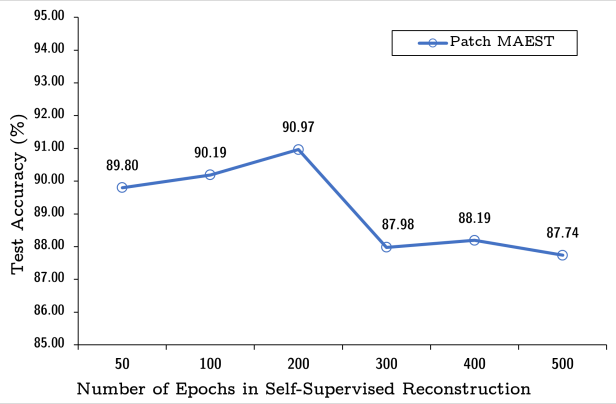


Fig. 9. Classification Accuracy for the Indian Pines dataset with different epochs in reconstruction (75% masking).

Indian Pines dataset, for the reconstruction pipeline (R), the classification pipeline (C) and in both. In the case of the separated pipelines analysis, we used a value of $5*10^{-4}$ in the other pipeline, as it is the learning rate which obtained the best results training both pipelines (R+C) with the same learning rate. For the R+C analysis, it can be seen how the best learning rate values are obtained in the 10^{-4} order, with a high difference to other values. Besides, in the separated pipelines analysis two main results can be pointed out. First, changes in the learning rate affect more to the C pipeline than the R, as the classification main task is much more complex than the masked reconstruction. Second, modifying the learning rate only in one of the pipelines generates lower changes in the AA, reinforcing the conclusion of $5*10^{-4}$ being the best learning rate value for this case.

5) Reconstruction Epochs Analysis: Even though the fact that the MAEST containing two pipelines could seem to make it converge slower than other methods, the number of epochs needed to converge the reconstruction is quite low, and once the robust representations are learned, in the classification process the performance is speeded up as well. In Figure 9 we provide the classification accuracy results for the Indian Pines dataset depending on the number of epochs in the self-supervised reconstruction. We observe that even with only 50 epochs, the improvement in accuracy is significant, rising up to 200 epochs, and then going down again stabilizing around an 88% of accuracy. We consider the reason behind this behavior is because the model over-fits and becomes unable to generalize enough for the later classification.

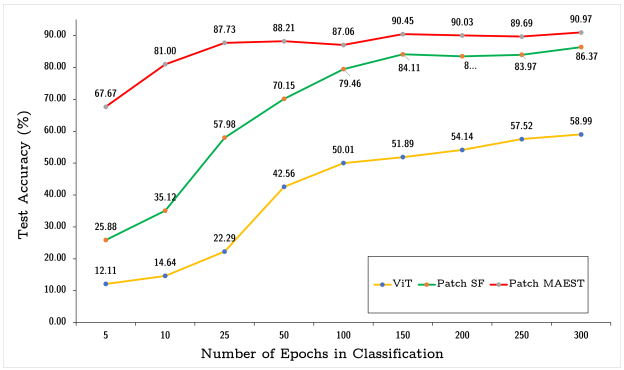


Fig. 10. Test accuracy on Indian Pines dataset for different training epochs.

6) Classification Epochs Analysis: In addition, after fixing the reconstruction epochs to 200, we compare the performance of the patch MAEST, patch SF and ViT with different numbers of classification training epochs in Figure 10. We trained the three models from 5 epochs to 300 in the Indian Pines dataset. As expected, both the SF and MAEST have a considerable difference in accuracy with the original ViT. However, it is interesting to see how the three different models behave in the first 100 epochs. It is clear how the MAEST is able to perform with more than an 80% of accuracy with only 10 training epochs, while the other methods are far below the 50%. The proposed MAEST not only converges faster, but is able to obtain over 90% accuracy in 150 epochs, obtaining a 4% accuracy improvement in almost the same total training epochs than the patch SF thanks to the previous learning from the reconstruction.

7) Classification Data Analysis: Finally, we also carried out an experiment to analyze the effect of the amount of training data we trained from the 10% of the original data, in jumps of 1/10 up to the complete training dataset of Indian Pines. The results for the ViT, pixel and patch SF, and pixel and patch MAEST are shown in Figure 11. In the graph, we observe how the ViT has overall worse results than the other transformer-based models, as was expected due to them being designed for HS image classification. It can also be noticed how the pixel SF has quite unstable results depending on the percentage of training data used, while the patch SF and both of the MAEST models steadily increase accuracy with more training data. While the patch SF and the pixel MAEST have a small fix difference in their accuracy for any percentage of data used, a huge gap can be seen between them and the patch MAEST, specially with lower data percentages. The spectral-spatial robust feature representations learned in the reconstruction pipeline allow the patch MAEST to have almost a 70% accuracy with only 1/10th of the already limited Indian Pines training set and a much steady increase in accuracy with more data.

V. CONCLUSIONS

This paper proposed a novel masked auto-encoding spectral-spatial transformer (MAEST) specially designed to classify HS remote sensing images. Unlike other existing models, MAEST takes advantage of two collaborative branches to alleviate intrinsic noise inside the network topology. Whereas the first

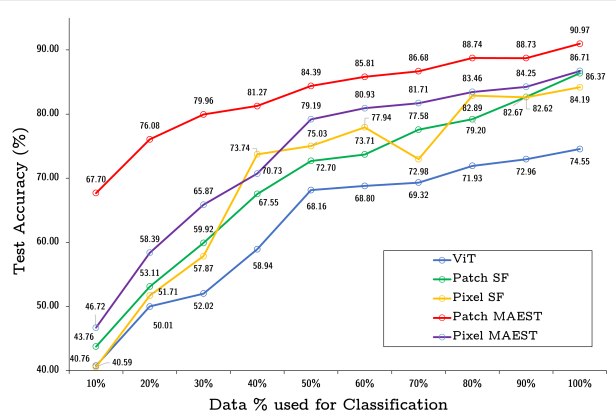


Fig. 11. Test accuracy on the Indian Pines dataset with different % of data in the classification training.

path adopts a masking auto-encoding strategy to uncover the most robust encoding features, the second transformer path pursues to learn how to classify the complete input data while exploiting these robust features. In this way, more accurate predictions and faster convergences can be achieved. The conducted experiments showed the competitive performance of the proposed approach with respect to multiple state-of-the-art classification models.

One of the first conclusions that arises from this work is the importance of transformer-based technologies in the context of HS image classification. In contrast to conventional classifiers or classic backbone networks. Transformers are able to provide superior abilities in characterizing sequential spectral data which may eventually make the difference within the HS image domain in terms of performance. However, it is important to note that not all the tested transformers achieved the same positive results. In this regard, another important conclusion is related to the understanding of the HS field for the successful exploitation of this technology. Being noise one of the main intricacies of HS data, learning more robust feature patterns within the transformer encoder becomes an outstanding strategy as the proposed model showed. Even though the obtained results are certainly promising, there are still important challenges for future research based on extending this work to multi-modal and multi-temporal data.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[3] R. Fernandez-Beltran, A. Plaza, J. Plaza, and F. Pla, "Hyperspectral unmixing based on dual-depth sparse probabilistic latent semantic analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6344–6360, 2018.

[4] R. Fernandez-Beltran, F. Pla, and A. Plaza, "Endmember extraction from hyperspectral imagery based on probabilistic tensor moments," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 12, pp. 2120–2124, 2020.

[5] K. Bi, S. Xiao, S. Gao, C. Zhang, N. Huang, and Z. Niu, "Estimating vertical chlorophyll concentrations in maize in different health states using hyperspectral lidar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 8125–8133, 2020.

[6] K. L.-M. Ang and J. K. P. Seng, "Big data and machine learning with hyperspectral information in agriculture," *IEEE Access*, vol. 9, pp. 36 699–36 718, 2021.

[7] S. Yang, L. Hu, H. Wu, H. Ren, H. Qiao, P. Li, and W. Fan, "Integration of crop growth model and random forest for winter wheat yield estimation from uav hyperspectral imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6253–6269, 2021.

[8] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 34–44, 2013.

[9] C.-I. Chang, "Hyperspectral anomaly detection: A dual theory of hyperspectral target detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[10] G. Lee, J. Lee, J. Baek, H. Kim, and D. Cho, "Channel sampler in hyperspectral images for vehicle detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[11] S. Veraverbeke, P. Dennison, I. Gitas, G. Hulley, O. Kalashnikova, T. Katagis, L. Kuai, R. Meng, D. Roberts, and N. Stavros, "Hyperspectral remote sensing of fire: State-of-the-art and future perspectives," *Remote Sensing of Environment*, vol. 216, pp. 105–121, 2018.

[12] X. Li, H. Wang, X. Li, Z. Tang, and H. Liu, "Identifying degraded grass species in inner mongolia based on measured hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5061–5075, 2019.

[13] H. Huang, Z. Sun, S. Liu, Y. Di, J. Xu, C. Liu, R. Xu, H. Song, S. Zhan, and J. Wu, "Underwater hyperspectral imaging for in situ underwater microplastic detection," *Science of The Total Environment*, vol. 776, p. 145960, 2021.

[14] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2012.

[15] J. Jia, Y. Wang, J. Chen, R. Guo, R. Shu, and J. Wang, "Status and application of advanced airborne hyperspectral imaging technology: A review," *Infrared Physics & Technology*, vol. 104, p. 103115, 2020.

[16] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 3, pp. 1579–1597, 2017.

[17] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 60–88, 2020.

[18] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, J. Yao, A. M. Khan, M. Mazzara, S. Distefano, and J. Chanussot, "Hyperspectral image classification–traditional to deep models: A survey for future prospects," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 968–999, 2022.

[19] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.

[20] N. Audebert, B. Le Saux, and S. Lefèvre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE geoscience and remote sensing magazine*, vol. 7, no. 2, pp. 159–173, 2019.

[21] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[22] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[23] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[24] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5384–5394, 2019.

[25] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Transac-*

tions on Geoscience and Remote Sensing, vol. 56, no. 9, pp. 5046–5063, 2018.

[26] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 740–754, 2018.

[27] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2020.

[28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[29] 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.

[30] 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.

[31] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[32] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, 2017.

[33] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot, "Noise reduction in hyperspectral imagery: Overview and application," *Remote Sensing*, vol. 10, no. 3, p. 482, 2018.

[34] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," *arXiv preprint arXiv:1812.05069*, 2018.

[35] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1691–1703.

[36] W. Song, S. Li, X. Kang, and K. Huang, "Hyperspectral image classification based on knn sparse representation," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016, pp. 2411–2414.

[37] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, vol. 1. IEEE, 2003, pp. 288–290.

[38] S. Amini, S. Homayouni, and A. Safari, "Semi-supervised classification of hyperspectral image using random forest algorithm," in *2014 IEEE geoscience and remote sensing symposium*. IEEE, 2014, pp. 2866–2869.

[39] W. Lv and X. Wang, "Overview of hyperspectral image classification," *Journal of Sensors*, vol. 2020, 2020.

[40] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.

[41] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[42] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[43] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5046–5063, 2018.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[45] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[46] G. E. Hinton and R. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *Advances in neural information processing systems*, vol. 6, 1993.

[47] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.

[48] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "Daen: Deep autoencoder networks for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4309–4321, 2019.

[49] A. Vahdat and J. Kautz, "Nvae: A deep hierarchical variational autoencoder," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 667–19 679, 2020.

[50] W. Xie, B. Liu, Y. Li, J. Lei, and Q. Du, "Autoencoder and adversarial-learning-based semisupervised background estimation for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5416–5427, 2020.

[51] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4823–4833, 2019.

[52] C. Zhang, L. Zhou, Y. Zhao, S. Zhu, F. Liu, and Y. He, "Noise reduction in the spectral domain of hyperspectral images using denoising autoencoder methods," *Chemometrics and Intelligent Laboratory Systems*, vol. 203, p. 104063, 2020.

[53] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[54] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[55] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.

[56] S. Becker and G. E. Hinton, "Self-organizing neural network that discovers surfaces in random-dot stereograms," *Nature*, vol. 355, no. 6356, pp. 161–163, 1992.

[57] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[58] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[59] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, vol. 1, 2005, pp. 29–36.

[60] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[61] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[62] 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.

[63] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.