Self-Organising Explainable Multi-View Representation Learning for Remote Sensing Scene Classification

Xiaowei Gu[1*], Abdulrahman Kerim[1], Jinghao Zhang[2,3,4], Jungong Han[5], Qiang Shen[6], Peter M Atkinson[7,8,9] and Ce Zhang[10,11*]

[1] School of Computer Science and Electronic Engineering, University of Surrey, Guildford, GU2 7XH, UK

[2] Shanghai Satellite Network Research Institute Co., Ltd., Shanghai 201306, China

[3] Shanghai Key Laboratory of Satellite Network, Shanghai 201306, China

[4] State Key Laboratory of Satellite Network, Shanghai 201306, China

[5] Department of Computer Science, University of Sheffield, Sheffield, S10 2TN, UK

[6] Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK

[7] Lancaster Environment Centre, Lancaster University, Lancaster, LA1 4YR, UK

[8] Geography and Environment, University of Southampton, Highfield, Southampton, SO17 1BJ, UK

[9] College of Surveying and Geo-Informatics, Tongji University, No.1239, Siping Road, Shanghai 200092, China

[10] School of Geographical Sciences, University of Bristol, Bristol, BS8 1SS, UK

[11] UK Centre for Ecology & Hydrology, Lancaster, LA1 4AP, UK

Email: {xiaowei.gu, a.kerim}@surrey.ac.uk, jinghao-zhang@outlook.com, jungonghan77@gmail.com, qqs@aber.ac.uk, pma@lancaster.ac.uk, ce.zhang@bristol.ac.uk

Abstract: Remote sensing scene classification is widely considered to be a challenging task due to high intraclass variability and interclass similarity in remotely sensed imagery. While existing deep neural networks achieve promising performance, they often lack transparency and generalisation capability. To enhance interpretability without sacrificing accuracy, a novel self-organising transparent multi-view representation learning framework based on evolving fuzzy neural encoders for remote sensing scene classification is introduced in this paper. The framework leverages multiple pre-trained convolutional neural network backbones with different architectures to extract image embeddings from multiple views. The multi-view image embeddings are projected into a lower-dimensional feature space using multilayer evolving fuzzy neural networks trained in a supervised or self-supervised fashion as encoders and subsequently fused for scene classification. Extensive experiments on six benchmark datasets (Optimal-31, WHU-RS, UCMerced, AID, RSI-CB256, and PatternNet) demonstrate the framework's superior performance, achieving average accuracies of 99.81%, 98.83%, 97.86%, 98.37%, 99.84%, and 98.83%, respectively, without fine-tuning to the specific context. Ablation studies confirm the complementary contributions of the multi-view, supervised, and self-supervised components in the proposed framework. The proposed framework provides an effective solution for remote sensing scene classification, achieving high accuracy with enhanced transparency and interpretability.

Keywords: fuzzy neural encoder; remote sensing scene classification; multi-view learning; supervised and self-supervised representation learning.

1. Introduction

Remote sensing scene classification is an active research area focused on interpreting images at the scene level by assigning land-use labels based on their semantic content [1]–[3]. It is a critical task in Earth observation, playing a vital role in geospatial applications, such as environmental monitoring, urban planning, precision agriculture, and natural disaster response [4]–[7]. However, remote sensing scene classification remains challenging, since images often exhibit high intraclass diversity and interclass similarity due to spectral heterogeneity and complex geometric structures [8], [9]. As such, intelligent scene classification methods are needed urgently in the remote sensing and machine learning communities [10], [11].

Over the past decades, many methods have been developed for remote sensing scene classification. Based on the visual features utilised, existing methods can be divided into three categories: low-level, mid-level and high-level [3], [12]. Early approaches focused primarily on low-level or mid-level visual features, combined with classical machine learning classifiers, such as the support vector machine (SVM) [12], *k*-nearest neighbour (KNN) [13], and random forest (RF) classifiers [14]. Commonly used low-level visual features include the histogram of oriented gradient [15], scale-invariant feature transform [16], local binary pattern [17], etc. These handcrafted features perform well on images with simple structures, but fail to capture the characteristics of complex remote sensing scenes due to their limited descriptive capabilities [2]. Mid-level methods attempt to construct holistic

[*] Corresponding author

representations by encoding low-level visual features [18]. Popular mid-level methods include bag-of-visual-words [19], spatial pyramid matching [20], locality-constrained linear coding [21], etc.. Mid-level methods generally are more accurate than low-level methods, but often depend heavily on the underlying low-level visual features. Additionally, they lack flexibility and are difficult to adapt to different problems [3].

Due to recent advances in deep learning technology, deep neural networks (DNNs), and convolutional neural networks (CNNs) in particular, have been utilised extensively for remote sensing scene classification [22]–[24]. Compared to low-level and mid-level features, the high-level visual features extracted by CNNs more effectively capture the semantic content of remote sensing images, allowing high-level methods to achieve greater accuracy compared to traditional handcrafted feature-based approaches. To better utilise pre-pertained CNNs for extracting more discriminative representations to facilitate remote sensing scene classification, Chaib et al. [25] designed a feature fusion framework that integrates high-level features extracted from multiple fully connected layers of a CNN model and refines them using discriminant correlation analysis (DCA). Given that fine-tuning the pretrained CNNs on remote sensing images is expected to yield more accurate classification results, Lu et al. [26] proposed an end-to-end approach that utilises semantic label information to guide the learning and fusion of intermediate convolutional features. As training a CNN is challenging with limited labelled data, Liu et al. [27] developed a Siamese CNN to learn regularised discriminative feature embeddings from paired remote sensing images. To overcome the limitation of CNNs focusing on global image-level features, but ignoring local object-level features, Wang et al. [2] introduced an enhanced feature pyramid network that simultaneously extracts multi-scale and multi-level features from remote sensing images. These features are aggregated using a deep semantic embedding module, to enhance scene classification by complementary information. As conventional fine-tuning of pretrained CNNs often overlooks domain shifts between the source and target datasets, Wang et al. [22] designed an adaptive learning strategy specifically for transferring a pretrained CNN to scene classification. This approach adaptively resizes input images, autonomously identifies critical information and transferable layers, and leverages label smoothing regularisation and model prediction statistics to better capture relationships between the target and nontarget scene categories.

Although CNNs have demonstrated high accuracy for remote sensing scene classification, a major limitation mentioned previously is their inability to effectively capture local information from the remote sensing images, which is crucial for addressing interclass similarity [28]–[30]. To overcome this limitation, attention mechanisms are increasingly being employed to enable CNNs to focus on the most relevant local regions of remote sensing images. To date, many attention mechanisms have been proposed to enhance CNNs for remote sensing scene classification, resulting in increased classification accuracy. For example, Guo et al. [28] proposed to replace the fully connected layers of the backbone CNN with a global attention branch and a local attention branch. Tong et al. [18] developed a channel attention mechanism that adaptively enhances important feature channels whilst suppressing less relevant ones. Wang et al. [8] introduced a channel-spatial attention mechanism to enhance local feature representation and better preserve features extracted at different layers. Chen et al. [29] proposed a dual attention-aware module that separately applies channel attention and spatial attention to high-level global features and low-level local features. Dai et al. [30] designed a correlation attention mechanism to facilitate feature fusion at different scales.

Another significant limitation of CNNs is their inability to capture the global interactions between local elements within images, which are vital for remote sensing scene classification [31], [32]. This deficiency can hinder the CNNs' ability to comprehend the spatial relationships and contextual information that are often essential for accurately distinguishing between different remote sensing scenes. Rather than merely increasing the depth of CNNs by stacking additional convolutional layers, a more promising strategy to enhance scene classification accuracy involves establishing long-range dependencies between local elements. This can be achieved either by integrating CNNs with transformers or by directly employing Vision Transformers (ViTs) [9], [33]. Bazi et al. [34] first explored the use of ViTs [35] for remote sensing scene classification, demonstrating excellent performance. Zhang et al. [32] integrated transformers with CNNs by replacing the final bottlenecks of CNN backbones with transformer encoders. Lv et al. [31] improved the discriminative ability of ViTs by incorporating a progressive aggregation strategy and a lightweight channel attention module, effectively capturing both the structural and channel information of remote sensing images. Ma et al. [36] proposed a ViT-based network that simultaneously captures homogeneous and heterogeneous information within remote sensing images. Discriminative feature representations are then generated leveraging local knowledge and global contextual information. Huang et al. [33] introduced a lightweight transformer network composed of stacked multi-level group convolution modules to extract multi-level local features from remote sensing images, along with a lightweight transformer block to capture long-range dependencies.

Although CNNs, attention-based CNNs and transformer-based networks represent the state-of-the-art in remote sensing scene classification, a key deficiency associated with these DNNs is their inherent "black box" nature, which limits transparency and interpretability. DNNs are highly complex models with huge amounts of hyperparameters that do not have direct, interpretable relationships with the problem domain, making their internal reasoning and decision-making process difficult for humans to understand [37]. Although there exist a few *post hoc* approaches for explaining DNNs (e.g., Shapley additive explanations [38], layer-wise relevance propagation [39], saliency [40]), their explanations reflect primarily model behaviour rather than human understanding, which can often be misleading [41]. Concerns about the lack of interpretability and explainability of DNNs have largely limited their wider deployment in high-stakes real-world applications. Therefore, despite the increasing use of DNNs in remote sensing scene classification, there remains an urgent need to develop transparent approaches that are capable of extracting high-level visual features from images, maintaining comparable scene classification accuracy with enhanced interpretability.

In this paper, a multi-view evolving fuzzy neural encoder (MVEFNE) framework is proposed for remote sensing scene classification. MVEFNE leverages multiple pretrained CNN backbones of different architectures for feature extraction to obtain multiple independent views of images. Each CNN backbone connects a pair of evolving fuzzy neural encoders (EFNEs), one trained in a supervised fashion and the other trained in a self-supervised fashion. The self-supervised EFNEs are trained to capture the most relevant features within a lower-dimensional space that can be used for reconstructing the original image embeddings, and the supervised EFNEs are trained to capture the interclass dissimilarity from the image embeddings. These EFNEs are implemented based on the recently introduced multilayer evolving fuzzy neural network (MEFNN) [37], which is a meta-learning approach designed to learn multi-level distributed representations from data. Within the proposed framework, EFNEs compress the image embeddings generated by multiple CNN backbones into highly descriptive, lower-dimensional latent representations, which are subsequently fused for classification. Thanks to their IF-THEN fuzzy rule structure, EFNEs represent the learned knowledge from CNN-based embeddings in an intuitive and interpretable form, providing enhanced transparency compared with state-of-the-art DNNs [42]. The human-like fuzzy inferencing process enables explicit explanation for each decision made by the EFNEs, offering intrinsic interpretability that differs from *post hoc* explanation approaches, such as Shapley additive explanations (SHAP) [43]. Additionally, the strong capability of fuzzy systems in handling real-world uncertainties contributes further to the transferability of these EFNEs. By combining multi-view learning with both self-supervised and supervised representation learning, the proposed MVEFNE framework effectively captures richer discriminative information from remote sensing images, leading to increased scene classification accuracy with greater interpretability. Experimental studies conducted over a variety of benchmark remote sensing datasets for scene classification test the performance of the proposed MVEFNE framework.

The novelty of the proposed MVEFNE framework, distinguishing it from existing works, lies in the following two aspects:

1) It integrates EFNEs with multiple CNN backbones for multi-view latent representation learning, enabling the extraction of richer discriminative information to achieve higher classification accuracy.
2) It presents the first attempt of using MEFNNs as encoders for both self-supervised and supervised representation learning, providing enhanced transparency and interpretability in comparison with DNNs.

The main contributions of this paper are summarised as follows:

1) A novel ensemble framework integrating multiple CNN backbones and EFNEs to learn multi-view, highly informative latent representations for remote sensing scene classification, with enhanced accuracy and transparency.
2) A systematic experimental investigation across six benchmark datasets, demonstrating superior accuracy and strong generalisation without fine-tuning, supported by ablation studies validating the synergistic effects of the framework's key components.

The remainder of this paper is organised as follows. Section 2 summarises the technical details of MEFNN as the implementation basis of EFNEs. The proposed MVEFNE framework is described in detail in Section 3. Numerical examples are presented in Section 4 for performance demonstration. Section 5 provides a conclusion.

2. Preliminary: MEFNN

In this section, technical details of MEFNN are presented, serving as the implementation basis of the proposed EFNEs, the core component of the proposed MVEFNE framework. MEFNN is designed to self-organise a multilayer, human-interpretable IF-THEN fuzzy rule base to autonomously learn multi-level latent representations from data [37]. However, it is important to note that the MEFNN used in this paper differs from the original MEFNN introduced in [37] in the following two aspects:

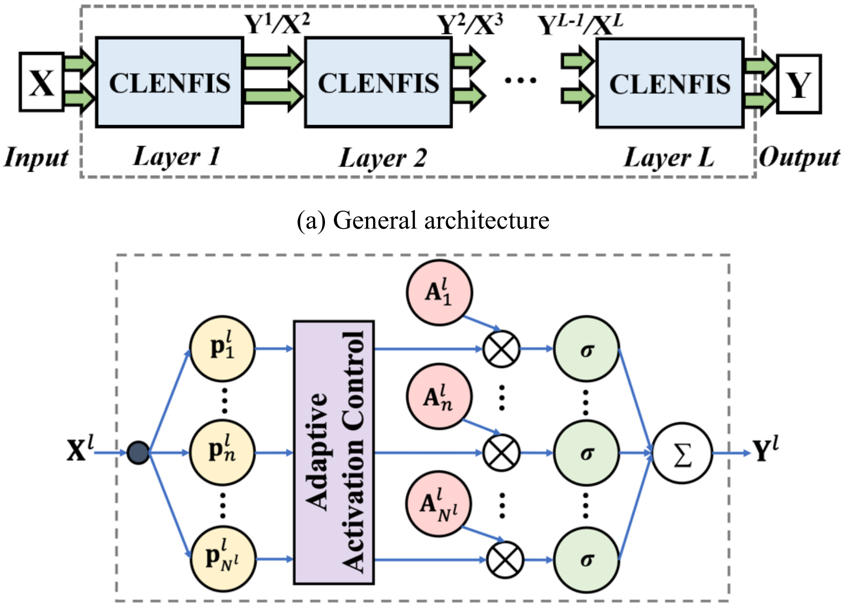
1) the consequent parameter updating is performed in a chunk-wise manner;
2) an adaptive activation control technique [44] is incorporated.

Compared with sample-wise, consequent parameter updating in the original MEFNN, the chunk-wise strategy reduces the stochastic variance of gradient estimates, leading to more stable convergence. This modification also lowers the frequency of function calls and memory access operations, thereby increasing computational efficiency. In addition, the adaptive activation control technique [44] temporarily excludes the fuzzy rules less activated by the current input data during consequent parameter updating and decision-making in each processing cycle. Such rules typically represent highly dissimilar data patterns to the current input data and, thus, contribute marginally to parameter updating and decision-making. Temporarily excluding the irrelevant rules further increases computational efficiency and prevents overfitting by avoiding unnecessary parameter updating.

To distinguish it from the original MEFNN, the variant used in this paper is referred to as MEFNN$^+$.

2.1. General Architecture

The general architectural of MEFNN$^+$ is depicted in Fig. 1. As shown in this figure, MEFNN$^+$ is composed of multiple chunk-wise learning evolving neuro-fuzzy inference systems (CLENFIS) arranged sequentially in layers. These CLENFISs within the stacking ensemble framework are fully connected, with each one taking the outputs produced by the previous layer as inputs, and using its outputs as the inputs to the next layer.



(a) General architecture

(b) Inner architecture of the lth CLENFIS

Fig. 1. General architecture of MEFNN$^+$

The inner architecture of a particular CLENFIS (assuming the lth one) is given by Fig. 1(b). One can see from Fig. 1(b) that the lth CLENFIS comprises N^l first IF-THEN fuzzy rules in the form of Eq. (1) ($n = 1,2,\dots,N^l$).

$$\mathbf{R}_n^l\text{:}\quad \text{IF } (\mathbf{x}^l \sim \mathbf{p}_n^l) \quad \text{THEN } \left(\mathbf{y}_n^l = \sigma(\mathbf{A}_n^l \bar{\mathbf{x}}^l)\right) \tag{1}$$

where $\mathbf{R}_n^l$ denotes the nth fuzzy rule of the lth CLENFIS; $\mathbf{x}^l = \left[x_1^l, x_2^l, \dots, x_{M^l}^l\right]^T$ is the $M^l \times 1$ dimensional input vector; $\bar{\mathbf{x}}^l = [1, (\mathbf{x}^l)^T]^T$; $\boldsymbol{p}_n^l$ is the $M^l \times 1$ dimensional prototype of $\mathbf{R}_n^l$; "~" denotes similarity between the prototype $\mathbf{p}_n^l$ and the input vector $\mathbf{x}^l$; $\mathbf{A}_n^l$ is the corresponding $W^l \times (M^l + 1)$ dimensional consequent parameter matrix; $\sigma(\cdot)$ is a standard nonlinear activation function to add extra nonlinearity to the consequent part of the fuzzy rule. In this paper, following the setting of [37], a sigmoid function is used, namely, $\sigma(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$. However, alternative activation functions could be used as well. The adaptive activation control scheme is implemented by Condition 2, specified in subsection 2.2 later.

Given the input data chunk $\mathbf{X}^l$, the output chunk of the lth CLENFIS is computed as [37]:

$$\mathbf{Y}^l = f^l(\mathbf{X}^l) = \textstyle\sum_{n=1}^{N^l} \boldsymbol{\Lambda}_n^l \odot \sigma(\mathbf{A}_n^l \bar{\mathbf{X}}^l) \tag{2}$$

where $\mathbf{X}^l = [\mathbf{x}_1^l, \mathbf{x}_2^l, \dots, \mathbf{x}_G^l]$ is the $M^l \times G$ dimensionality input data matrix; $\bar{\mathbf{X}}^l = [\bar{\mathbf{x}}_1^l, \bar{\mathbf{x}}_2^l, \dots, \bar{\mathbf{x}}_G^l]$; G is the cardinality of $\mathbf{X}$; $\boldsymbol{\Lambda}_n^l$ is the $1 \times G$ dimensional normalised firing strength vector produced by $\boldsymbol{R}_n^l$ in response to $\mathbf{X}^l$ (given by Eq. (13)); "$\odot$" denotes element-wise multiplication.

For a MEFNN$^+$ consisting of L CLENFISs, its input and output relationship can be formulated as a composite function given by Eq. (3) [37]:

$$\mathbf{Y} = f^L \circ \ldots \circ f^2 \circ f^1(\mathbf{X}) \tag{3}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the respective $M \times G$ and $W \times G$ dimensional input and output matrices to MEFNN$^+$. In the next subsection, the system identification process of MEFNN$^+$ is presented.

2.2. System Identification Process

The system identification process of MEFNN$^+$ consists of two stages repeated for every input data chunk.

Stage 1. Forward Learning

Given a new input data chunk $\mathbf{X}_c = [\mathbf{x}_{c,1}, \mathbf{x}_{c,2}, \ldots, \mathbf{x}_{c,G_c}]$ and the corresponding target output chunk $\mathbf{T}_c = [\mathbf{t}_{c,1}, \mathbf{t}_{c,2}, \ldots, \mathbf{t}_{c,G_c}]$ (G_c is the number of data samples in $\mathbf{X}_c$; $\mathbf{t}_{c,j}$ is the target output of $\mathbf{x}_{c,j}$), MEFNN$^+$ will self-organise and self-evolve its multilayer fuzzy rule base in a sample-wise manner, in the same way as for MEFNN.

The multilayer system structure of MEFNN$^+$ is initialised by $\mathbf{X}_c$ if $\mathbf{X}_c$ is the very first data chunk received by MEFNN$^+$, namely, $c = 1$. With the very first input sample $\mathbf{x}^l_{c,j}$ ($j = 1$) of the input data chunk $\mathbf{X}^l_c$ ($\mathbf{X}^l_c \leftarrow \mathbf{X}_c$ if $l = 1$ and $\mathbf{X}^l_c \leftarrow \mathbf{Y}^{l-1}_c$ otherwise), the global parameters of the lth CLENFIS are initialised using Eq. (4) ($l = 1,2, \ldots, L$):

$$K^l \leftarrow 1; \quad \boldsymbol{\mu}^l \leftarrow \mathbf{x}^l_{c,j}; \quad X^l \leftarrow \left\|\mathbf{x}^l_{c,j}\right\|^2 \tag{4}$$

where K^l is the number of data samples that the lth CLENFIS has processed; $\boldsymbol{\mu}^l$ and X^l are the means of the data samples and their squared Euclidean norms, respectively.

The first fuzzy rule, denoted as $\mathbf{R}^l_{N^l}$ ($N^l \leftarrow 1$) is initialised in the form of Eq. (1) with the antecedent and consequent parameters set as:

$$\mathbf{p}^l_{N^l} \leftarrow \mathbf{x}^l_{c,j}; \quad \mathbf{A}^l_{N^l} \leftarrow \mathbf{V}^l_{N^l} \tag{5}$$

where $\mathbf{p}^l_{N^l}$ and $\mathbf{A}^l_{N^l}$ are the prototype and consequent parameter matrix of $\mathbf{R}^l_{N^l}$; $\mathbf{V}^l_{N^l}$ is a unique $W^l \times (M^l + 1)$-dimensional very sparse random projection (VSRP) matrix that is generated randomly [45].

The main reason for using a VSRP matrix as the initial consequent parameter matrix is because it better preserves the mutual dissimilarity between data samples in the original data space, compared to a purely random matrix or a zero matrix.

Parameters of the cluster associated with $\mathbf{R}^l_{N^l}$, denoted as $\mathbb{C}^l_{N^l}$, are set as:

$$S^l_{N^l} \leftarrow 1; \quad \boldsymbol{\varphi}^l_{N^l} \leftarrow \mathbf{x}^l_{c,j}; \quad \chi^l_{N^l} \leftarrow \left\|\mathbf{x}^l_{c,j}\right\|^2 \tag{6}$$

where $\boldsymbol{\varphi}^l_{N^l}$ and $\chi^l_{N^l}$ are the means of the data samples and their squared Euclidean norms associated with $\mathbb{C}^l_{N^l}$; $S^l_{N^l}$ is the number of such data samples.

Otherwise, if $\mathbf{x}^l_{c,j}$ is not the first data sample ($c > 1$ or $j > 1$), the global parameters of the lth CLENFIS are updated firstly:

$$K^l \leftarrow K^l + 1; \quad \boldsymbol{\mu}^l \leftarrow \boldsymbol{\mu}^l + \frac{\mathbf{x}^l_{c,j} - \boldsymbol{\mu}^l}{K^l}; \quad X^l \leftarrow X^l + \frac{X^l - \left\|\mathbf{x}^l_{c,j}\right\|^2}{K^l} \tag{7}$$

Then, the data density value of $\mathbf{x}^l_{c,j}$ at each data cluster is calculated using Eq. (8).

$$D_n\left(\mathbf{x}^l_{c,j}\right) = e^{-\frac{\left\|\mathbf{x}^l_{c,j} - \mathbf{p}^l_n\right\|^2}{\tau^l_n}} \tag{8}$$

where $\tau^l_n = \frac{X^l - \left\|\boldsymbol{\mu}^l\right\|^2 + \chi^l_n - \left\|\boldsymbol{\varphi}^l_n\right\|^2}{2}$.

Condition 1 (Eq. (9)) is examined to see if $\mathbf{x}^l_{c,j}$ differs significantly from existing data patterns observed from historical data and represents a novel data pattern [37]:

$$\mathit{Cond.1:} \quad \begin{array}{c} \mathit{if}\ \left(\max\limits_{\forall n}\left(D_n\left(\mathbf{x}^l_{c,j}\right)\right) < \delta_o\right) \\ \mathit{then}\ \left(\mathbf{x}^l_{c,j}\ \mathit{represents\ a\ novel\ data\ pattern}\right) \end{array} \tag{9}$$

where δ_o is a threshold to determine whether $\mathbf{x}_{c,j}^l$ is sufficiently different from existing data patterns represented by the identified prototypes, and there is $\delta_o = e^{-3}$, following [37].

If Condition 1 is satisfied by $\mathbf{x}_{c,j}^l$, a new fuzzy rule is added to the rule base of the lth CLENFIS ($N^l \leftarrow N^l + 1$) with the antecedent and consequent parameters set by Eq. (5) and the parameters of the associated cluster initialised by Eq. (6).

If Condition 1 is not met, parameters of the nearest cluster are updated using Eq. (10):

$$S_{n^*}^l \leftarrow S_{n^*}^l + 1; \quad \boldsymbol{\varphi}_{n^*}^l \leftarrow \boldsymbol{\varphi}_{n^*}^l + \frac{\mathbf{x}_{c,j}^l - \boldsymbol{\varphi}_{n^*}^l}{S_{n^*}^l}; \quad \chi_{n^*}^l \leftarrow \chi_{n^*}^l + \frac{\left\|\mathbf{x}_{c,j}^l\right\|^2 - \chi_{n^*}^l}{S_{n^*}^l} \tag{10}$$

where $n^* = \underset{\forall n}{\operatorname{argmax}}\left(D_n\left(\mathbf{x}_{c,j}^l\right)\right)$; $S_{n^*}^l$, $\boldsymbol{\varphi}_{n^*}^l$ and $\chi_{n^*}^l$ are the parameters associated with $\mathbb{C}_{n^*}^l$.

After $\mathbf{x}_{c,j}^l$ has been processed, the lth CLENFIS continues to process the remaining elements of the data chunk $\mathbf{X}_c^l$ until all the input samples have been leveraged for updating the fuzzy rule base. Once the fuzzy rule base is updated, the lth CLENFIS continues to produce the outputs in response to $\mathbf{X}_c^l$

To do so, the data density values of every input sample within $\mathbf{X}_c^l$ at the clusters associated with the IF-THEN fuzzy rules of the lth CLENFIS are re-calculated using Eq. (8) with the updated global and local parameters. Then, for each input sample (assuming the jth one), its data density values are ranked in descending order as: $D_1^*\left(\mathbf{x}_{c,j}^l\right) \geq D_2^*\left(\mathbf{x}_{c,j}^l\right) \geq \cdots \geq D_{N^l}^*\left(\mathbf{x}_{c,j}^l\right)$, and the number of fuzzy rules activated by $\mathbf{x}_{c,j}^l$ is computed using Eq. (11) ($j = 1,2, \ldots, G_c$) [44].

$$\hat{n}_j = \underset{\forall n}{\operatorname{argmin}}\left(\sum_{i=1}^{n} D_i^*\left(\mathbf{x}_{c,j}^l\right) \geq \rho_o \sum_{i=1}^{N^l} D_i^*\left(\mathbf{x}_{c,j}^l\right)\right) \tag{11}$$

where $\rho_o = 0.95$.

Based on n_j^*, Condition 2 (Eq. (12)) is utilised to identify the fuzzy rules that are activated by $\mathbf{x}_{c,j}^l$ [19]:

$$Cond. 2: \quad \begin{array}{c} \text{if } \left(D_n\left(\mathbf{x}_{c,j}^l\right) \geq D_{\hat{n}_j}^*\left(\mathbf{x}_{c,j}^l\right)\right) \\ \text{then } \left(\mathbf{R}_n^l \text{ is activated by } \mathbf{x}_{c,j}^l\right) \end{array} \tag{12}$$

The normalised firing strength vector produced by $\mathbf{R}_n^l$ in response to $\mathbf{X}^l$ is then obtained by Eq. (13):

$$\boldsymbol{\Lambda}_{n,c}^l = \left[\lambda_{n,c,1}^l, \lambda_{n,c,2}^l, \ldots, \lambda_{n,c,G_c}^l\right] \tag{13}$$

where $\lambda_{n,c,j}^l = \frac{\alpha_{n,c,j}^l D_n\left(\mathbf{x}_{c,j}^l\right)}{\sum_{i=1}^{N^l} \alpha_{i,c,j}^l D_i\left(\mathbf{x}_{c,j}^l\right)}$; $\alpha_{n,c,j}^l = \begin{cases} 1 & if \ \mathbf{R}_n^l \text{ satisfies Cond. 2} \\ 0 & \text{otherwise} \end{cases}$; $j = 1,2, \ldots, G_c$.

After all the normalised firing strength vectors have been obtained, the output of the lth CLENFIS, denoted as $\mathbf{Y}_c^l = \left\{\mathbf{y}_{c,1}^l, \mathbf{y}_{c,2}^l, \ldots, \mathbf{y}_{c,G_c}^l\right\}$ can be computed using Eq. (2), and $\mathbf{Y}_c^l$ is then passed to the $(l+1)$th CLENFIS at the next layer as the input data chunk, namely, $\mathbf{X}_c^l \leftarrow \mathbf{Y}_c^{l-1}$. The same process is repeated until the final CLENFIS is updated and the final outputs of MEFNN+, denoted as $\mathbf{Y}_c$ ($\mathbf{Y}_c \leftarrow \mathbf{Y}_c^L$) are obtained, after which the system identification process enters Stage 2 to update the consequent parameters of the multilayer fuzzy rule base in a backward manner.

Stage 2. Backward Learning

In this stage, the consequent parameters of MEFNN+ are updated layer-by-layer, starting from the final layer and moving to the first, using error backpropagation in a chunk-wise manner.

First, given the target output matrix, $\mathbf{T}_c$ and the predicted output matrix, $\mathbf{Y}_c$, the squared prediction error of MEFNN+ is calculated using Eq. (14) [37].

$$e_c = \frac{1}{2}\sum_{j=1}^{G_c}\left(\mathbf{y}_{c,j} - \mathbf{t}_{c,j}\right)^T\left(\mathbf{y}_{c,j} - \mathbf{t}_{c,j}\right) \tag{14}$$

The derivative of e_c with respect to $\mathbf{Y}_c$ can be obtained as a $(W \times G_c)$-dimensional matrix using Eq. (15):

$$\mathbf{d}_c^L = \frac{\partial e_c}{\partial \mathbf{Y}_c} = \mathbf{Y}_c - \mathbf{T}_c \tag{15}$$

where $\mathbf{Y}_c^L = \mathbf{Y}_c$.

Utilising the chain rule, the derivative of e_c with respect to the outputs of the lth layer can be obtained as ($l = 1,2, \ldots, L-1$):

$$\mathbf{d}_c^l = \frac{\partial e_c}{\partial \mathbf{Y}_c^l} = \frac{\partial e_c}{\partial \mathbf{Y}_c^L} \cdot \frac{\partial \mathbf{Y}_c^L}{\partial \mathbf{Y}_c^{L-1}} \cdot \dots \cdot \frac{\partial \mathbf{Y}_c^{l+1}}{\partial \mathbf{Y}_c^l} = \mathbf{d}_c^{l+1} \cdot \frac{\partial \mathbf{Y}_c^{l+1}}{\partial \mathbf{Y}_c^l}$$
$$= \sum_{n=1}^{N^{l+1}} \left[\mathbf{Q}_{n,c}^{l+1} + \mathbf{O}_{n,c}^{l+1}\right] \tag{16}$$

where $\mathbf{X}_c^{l+1} = \mathbf{Y}_c^l$; $\mathbf{d}_c^{l+1}$ is defined by Eq. (17); $\mathbf{Q}_{n,c}^{l+1}$ and $\mathbf{O}_{n,c}^{l+1}$ are defined by Eqs. (18) and (19), respectively.

$$\mathbf{d}_c^{l+1} = \frac{\partial e_c}{\partial \mathbf{Y}_c^{l+1}} = \frac{\partial e_c}{\partial \mathbf{Y}_c^L} \cdot \frac{\partial \mathbf{Y}_c^L}{\partial \mathbf{Y}_c^{L-1}} \cdot \dots \cdot \frac{\partial \mathbf{Y}_c^{l+2}}{\partial \mathbf{Y}_c^{l+1}} \tag{17}$$

$$\mathbf{Q}_{n,c}^{l+1} = \mathbf{\Lambda}_n^{l+1} \odot \left((\overline{\mathbf{A}}_n^{l+1})^T \cdot \mathbf{I}_{n,c}^{l+1}\right) \tag{18}$$

$$\mathbf{O}_{n,c}^{l+1} = \frac{\partial \mathbf{\Lambda}_n^{l+1}}{\partial \mathbf{X}_c^{l+1}} \cdot \left(\mathbf{Y}_{n,c}^{l+1}\right)^T \cdot \mathbf{d}_c^{l+1} \tag{19}$$

Here $\overline{\mathbf{A}}_n^{l+1}$ is the consequent parameter matrix obtained from $\mathbf{A}_n^{l+1}$ by removing the first column; $\mathbf{I}_{n,c}^{l+1} = \mathbf{d}_c^{l+1} \odot \sigma'\ (\mathbf{A}_n^{l+1}\overline{\mathbf{X}}_c^{l+1})$; $\sigma'\ (\mathbf{A}_n^{l+1}\overline{\mathbf{X}}_c^{l+1}) = \sigma(\mathbf{A}_n^{l+1}\overline{\mathbf{X}}_c^{l+1}) \odot \left(1 - \sigma(\mathbf{A}_n^{l+1}\overline{\mathbf{X}}_c^{l+1})\right)$; and $\frac{\partial \mathbf{\Lambda}_n^{l+1}}{\partial \mathbf{X}_c^{l+1}}$ is defined by Eq. (20).

$$\frac{\partial \mathbf{\Lambda}_n^{l+1}}{\partial \mathbf{X}_c^{l+1}} = \left[\frac{\partial \lambda_{n,c,1}^{l+1}}{\partial \mathbf{x}_{c,1}^{l+1}}, \frac{\partial \lambda_{n,c,2}^{l+1}}{\partial \mathbf{x}_{c,2}^{l+1}}, \dots, \frac{\partial \lambda_{n,c,G_c}^{l+1}}{\partial \mathbf{x}_{c,G_c}^{l+1}}\right] \tag{20}$$

where $\frac{\partial \lambda_{n,c,j}^{l+1}}{\partial \mathbf{x}_{c,j}^{l+1}} = \lambda_{n,c,j}^{l+1} \cdot \left(\frac{\mathbf{p}_n^{l+1} - \mathbf{x}_{c,j}^{l+1}}{\tau_n^{l+1}} - \sum_{i=1}^{N^{l+1}} \lambda_{i,c,j}^{l+1} \cdot \frac{\mathbf{p}_i^{l+1} - \mathbf{x}_{c,j}^{l+1}}{\tau_i^{l+1}}\right)$.

The derivative of e_c with respect to the consequent parameter matrix $\mathbf{A}_n^l$ can be obtained as [18].

$$\frac{\partial e_c}{\partial \mathbf{A}_n^l} = \frac{\partial e_c}{\partial \mathbf{Y}_c^L} \cdot \frac{\partial \mathbf{Y}_c^L}{\partial \mathbf{Y}_c^{L-1}} \cdot \dots \cdot \frac{\partial \mathbf{Y}_c^{l+1}}{\partial \mathbf{Y}_c^l} \cdot \frac{\partial \mathbf{Y}_c^l}{\partial \mathbf{A}_n^l} = \mathbf{d}_c^l \cdot \frac{\partial \mathbf{Y}_c^l}{\partial \mathbf{A}_n^l}$$
$$= \mathbf{\Lambda}_n^l \odot \left[\mathbf{I}_{n,c}^l \cdot (\overline{\mathbf{X}}_c^l)^T\right] \tag{21}$$

where $\mathbf{I}_{n,c}^l = \mathbf{d}_c^l \odot \sigma'\ (\mathbf{A}_n^l\overline{\mathbf{X}}_c^l)$.

After calculating the derivatives of e_c with respect to the consequent parameter matrices $\mathbf{A}_n^l$ ($\forall l = 1,2,\dots,L; n = 1,2,\dots,N^l$) of the multilayer fuzzy rule base, one-by-one, using Eqs. (16)-(21), the consequent parameter matrices are then updated using Eq. (22) [37].

$$\mathbf{A}_n^l \leftarrow \mathbf{A}_n^l - \gamma_o \cdot \frac{\partial e_c}{\partial \mathbf{A}_n^l} \tag{22}$$

where γ_o is the learning rate and, by default, $\gamma_o = 1$.

Once all the consequent parameter matrices are updated, the current processing cycle is completed. MEFNN$^+$ will continue to process the next data chunk by starting the next processing cycle ($c \leftarrow c + 1$).

For clarity, the algorithmic procedure of the system identification process is summarised by Algorithm 1 in the form of a pseudo code.

Algorithm 1. System identification process of MEFNN$^+$

```
$c \leftarrow 1$;
while ($\mathbf{X}_c$ is available) do:
########## Forward learning ##########
  for $l = 1$ to $L$ do:
    if ($l = 1$) then:
      $\mathbf{X}_c^l \leftarrow \mathbf{X}_c$;
    else:
      $\mathbf{X}_c^l \leftarrow \mathbf{Y}_c^{l-1}$;
    end if
    for $j = 1$ to $G_c$ do:
      if ($c = 1$) and ($j = 1$) then:
        initialise $K^l$, $\boldsymbol{\mu}^l$ and $X^l$ using (4);
        $N^l \leftarrow 1$;
        initialise $\mathbf{R}_{N^l}^l$ and $\mathbb{C}_{N^l}^l$ using (5) and (6);
      else:
        update $K^l$, $\boldsymbol{\mu}^l$ and $X^l$ using (7);
        for $n = 1$ to $N^l$ do:
          calculate $D_n(\mathbf{x}_{c,j}^l)$ using (8);
        end for
```

```
                if (Condition 1 is met) then:
                    $N^l \leftarrow N^l + 1$;
                    initialise $\mathbf{R}_{N^l}^l$ and $\mathbb{C}_{N^l}^l$ using (5) and (6);
                else:
                    $n^* = \underset{\forall n}{\operatorname{argmax}}\left(D_n\left(\mathbf{x}_{c,j}^l\right)\right)$;
                    update $\mathbb{C}_{n^*}^l$ using (10);
                end if
            end if
        end for
        for $n = 1$ to $N^l$ do:
            calculate $\mathbf{\Lambda}_{n,c}^l$ using (11)-(13);
        end for
        compute $\mathbf{Y}_c^l$ using (2);
    end for
    $\mathbf{Y}_c \leftarrow \mathbf{Y}_c^L$;
########### Backward learning ##########
    calculate $\mathbf{d}_c^L$ using (15)
    for $l = L$ to 1 do:
        for $n = 1$ to $N^l$ do:
            calculate $\frac{\partial e_c}{\partial \mathbf{A}_n^l}$ using (21);
            update $\mathbf{A}_n^l$ using (22);
        end for
        if $(l > 1)$ then:
            calculate $\mathbf{d}_c^{l-1}$ using (16);
        end if
    end for
    $c \leftarrow c + 1$;
end while
```

3. Proposed MVEFNE Framework for Image Classification

The proposed MVEFNE framework is a generic ensemble classification approach for remote sensing scene images, and its pipeline is depicted in Fig. 2.

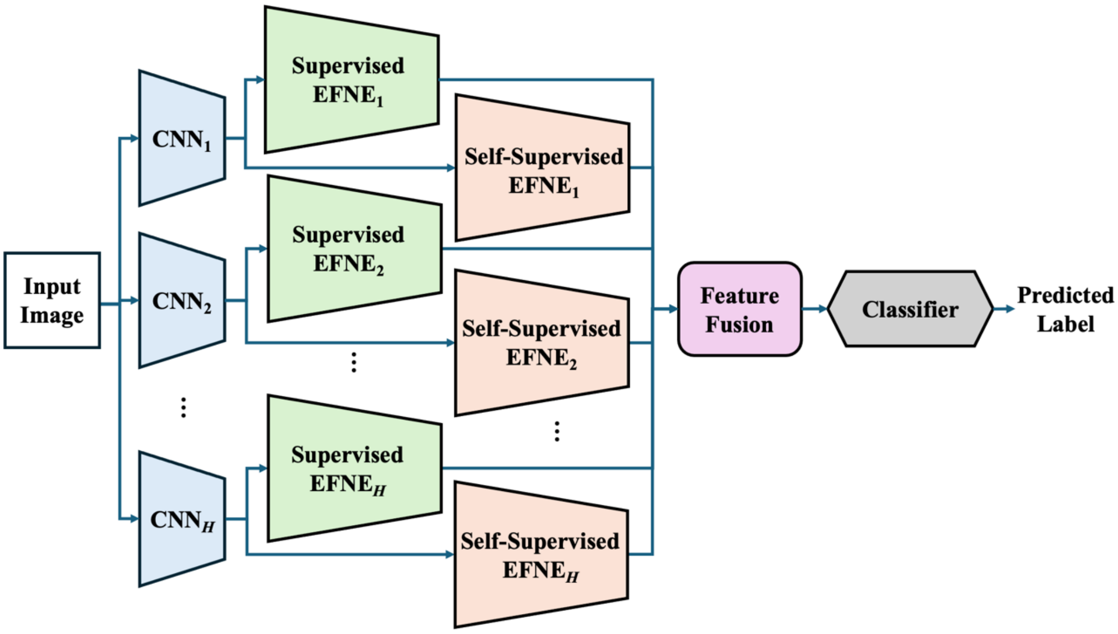


Fig. 2. Proposed MVEFNE framework

As shown in Fig. 2, the proposed MVEFNE framework consists of the following components: 1) H CNN backbones of different architectures; 2) H supervised EFNEs; 3) H self-supervised EFNEs; 4) a feature fusion module, and 5) a standard classifier.

Each input image presented to MVEFNE denoted as $\mathbf{I}$ is firstly passed to H CNN backbones of different architectures for feature extraction. Each CNN backbone (assuming the ith one) will extract a unique vectorised embedding from the image:

$$\mathbf{e}_i \leftarrow B_i(\mathbf{I}) \tag{23}$$

where $B_i(\cdot)$ denotes the ith CNN backbone; $\mathbf{e}_i$ is the image embedding of the ith CNN backbone learned from $\mathbf{I}$; $i = 1,2, \dots, H$.

Thus, a total of H embeddings will be extracted from every input image. To attain multiple views from the same image, each of the CNN backbones should have a distinct architecture. This is crucial to ensure diversity in the extracted image embeddings. Note that, the employed CNN backbones used by the MVEFNE framework can be pre-trained models on large-scale image datasets (e.g., ImageNet), without fine-tuning. This can significantly enhance the adaptability and flexibility of the proposed MVEFNE framework. Nevertheless, fine-tuning the CNN models specifically for remote sensing scene classification tasks could yield more discriminative image embeddings, thereby potentially improving the performance of MVEFNE.

The image embedding extracted by each CNN backbone (assuming the ith one) is processed subsequently by both a supervised EFNE and a self-supervised EFNE to generate more descriptive and compact representations:

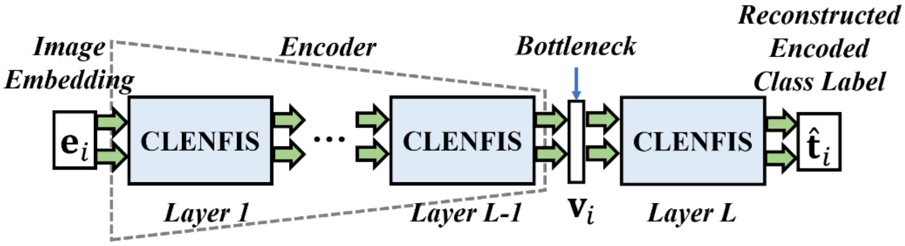
$$\mathbf{v}_i \leftarrow V_i(\mathbf{e}_i) \tag{24}$$

$$\mathbf{u}_i \leftarrow U_i(\mathbf{e}_i) \tag{25}$$

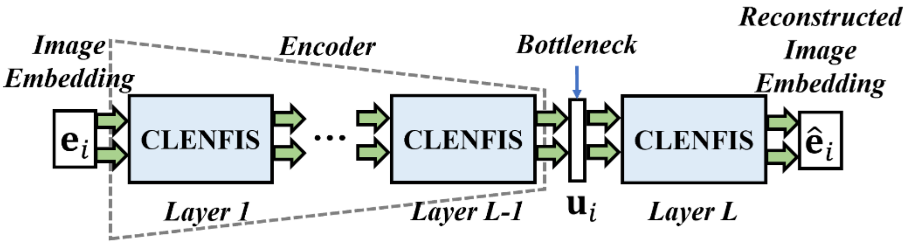
where $V_i(\cdot)$ and $U_i(\cdot)$ denote the supervised and self-supervised EFNEs connected to the ith CNN backbone, respectively; $\mathbf{v}_i$ and $\mathbf{u}_i$ are the respective latent representations learned by supervised and self-supervised EFNEs; $i = 1,2, \dots, H$.

The proposed MVEFNE framework employs $2H$ EFNEs in total, with half of them trained in a supervised manner and the other half trained in a self-supervised manner. The EFNEs leveraged in MVEFNE are built upon MEFNN$^+$ by removing the final layer(s) and incorporating a bottleneck constraint. The constraint enforces the output size of the bottleneck layer (namely, the layer used for latent representation extraction) to be less than the input size of the MEFNN$^+$. As a result, the EFNEs can learn more descriptive and informative latent representations from the input image embeddings by capturing the most relevant features within a lower-dimensional space.

The architecture of the proposed EFNE is given by Fig. 3. As shown in this figure, under the supervised learning setting, a EFNE is based on a MEFNN$^+$ that is trained to map input image embeddings to the targeted one-shot encoded class labels. In contrast, under the self-supervised learning setting, a EFNE is based on a MEFNN$^+$ that is trained to reconstruct the input image embeddings by using the original inputs as the target outputs. Supervised EFNEs are more effective at capturing the distinctive features in image embeddings across different classes, whilst self-supervised EFNEs are more effective at capturing the important features of the image embeddings that are crucial for accurate reconstruction. By utilising both supervised and self-supervised EFNEs for latent representation learning, the proposed MVEFNE framework combines the strengths of both approaches, maximising the information extracted from the image embeddings. It is also worth noting that thanks to the strong capability of fuzzy systems in handling real-world uncertainties, these EFNEs exhibit good transferability. In general, once trained on a given problem, their structures and parameters can remain fixed when applied to a new problem with similar characteristics.



(a) Supervised encoder



(b) Self-supervised encoder

Fig. 3. Proposed EFNE based on MEFNN$^+$

Next, the representations extracted by the $2H$ EFNEs are passed to the feature fusion module. If the concatenation operation is used, the feature fusion module will concatenate the $2H$ representations of the image **I** into a single higher-dimensional fused representation:

$$\mathbf{z} \leftarrow \mathbf{v}_1 \oplus \mathbf{u}_1 \oplus \mathbf{v}_2 \oplus \mathbf{u}_2 \oplus \ldots \oplus \mathbf{v}_H \oplus \mathbf{u}_H \tag{26}$$

where "⊕" denotes the concatenation operation.

Alternatively, if the averaging operation is used and the $2H$ latent representations learned by the EFNEs are of the same dimensionality, the module will produce the fused representation of the image **I** as the arithmetic mean of the representations:

$$\mathbf{z} \leftarrow \frac{1}{2H} \sum_{i=1}^{H} (\mathbf{v}_i + \mathbf{u}_i) \tag{27}$$

Finally, the fused representation **z** is passed to the classifier for training or predicting the class label of the image **I**:

$$\hat{t} \leftarrow F(\mathbf{z}) \tag{28}$$

where $F(\cdot)$ denotes the classifier, and $\hat{t}$ is the predicted class label of **I**.

The proposed MVEFNE framework can utilise any standard classifier, such as logistic regression (LR), SVM [46], KNN [47], RF [48], etc.. In this paper, a linear SVM classifier is used, following common practice [49], [50]. The algorithmic procedure of the training process of the proposed MVEFNE framework is summarised by Algorithm 2 for better illustration. The decision-making process of the proposed MVEFNE framework given a particular testing image is summarised by Algorithm 3. For clarity, the representation learning pipeline of the proposed MVEFNE framework is visualised in Fig. A1 (Appendix A).

Algorithm 2. Training process of MVEFNE

```
for $i = 1$ to $H$ do:
    extract image embeddings $\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \ldots, \mathbf{e}_{i,K}$ from images $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_K$ by the $i$th CNN backbone using (23);
    obtain the $i$th supervised EFNE by training a MEFNN$^+$ to map the image embeddings $\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \ldots, \mathbf{e}_{i,K}$ to the corresponding encoded class labels, $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_K$ using Algorithm 1;
    extract the latent representations $\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \ldots, \mathbf{v}_{i,K}$ by the $i$th supervised EFNE using (24);
    obtain the $i$th self-supervised EFNE by training a MEFNN$^+$ to reconstruct the image embeddings $\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \ldots, \mathbf{e}_{i,K}$ using Algorithm 1;
    extract the latent representations $\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \ldots, \mathbf{u}_{i,K}$ by the $i$th self-supervised EFNE using (25);
end for
fuse the latent representations extracted by EFNEs and obtain $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ using (26)/(27);
train the classifier using $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ and class labels $t_1, t_2, \ldots, t_K$;
```

Algorithm 3. Decision-making process of MVEFNE

```
for $i = 1$ to $H$ do:
    extract image embedding $\mathbf{e}$ from image $\mathbf{I}$ by the $i$th CNN backbone using (23);
    extract the latent representation $\mathbf{v}_i$ by the $i$th supervised EFNE using (24);
    extract the latent representation $\mathbf{u}_i$ by the $i$th self-supervised EFNE using (25);
end for
fuse the latent representations extracted by EFNEs and obtain $\mathbf{z}$ using (26)/(27);
classify the class label of $\mathbf{I}$ using (28);
```

4. Experimental Investigation

In this section, numerical examples on a range of popular benchmark datasets for remote sensing scene classification are presented to demonstrate the performance of the proposed MVEFNE. The algorithms were implemented in Python using a laptop i7-12700H processor, 64GB RAM and RTX3050Ti GPU. Unless specifically declared otherwise, the reported numerical results were obtained as the average of 10 Monte Carlo experiments by default to allow a certain degree of randomness. The source code is available on this repository[1].

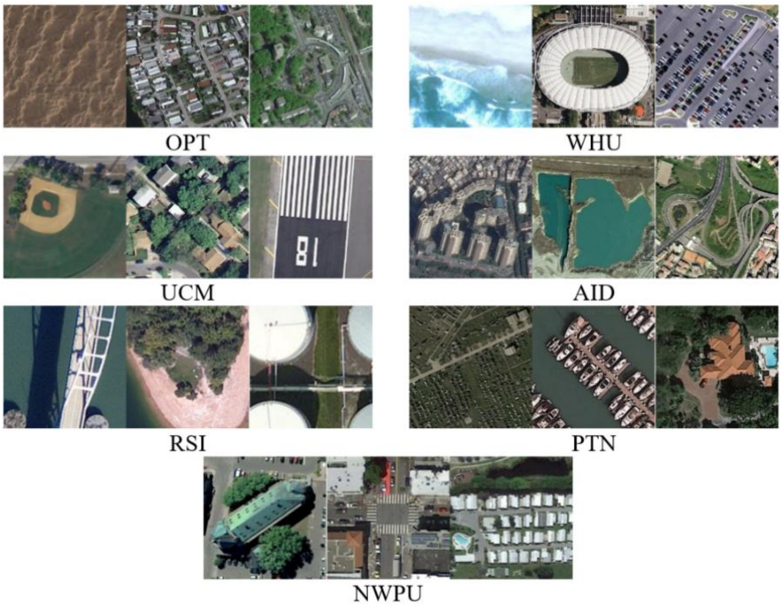
4.1. Configuration

1) Data description. The following seven benchmark datasets for remote sensing scene classification were exploited in the experimental investigation, which include: 1) Optimal-31 (OPT)[2] [51]; 2) WHU-RS19 (WHU)[3] [52]; 3) UCMerced (UCM)[4] [19]; 4) AID[5] [12]; 5) RSI-CB256 (RSI)[6] [53]; 6) PatternNet (PTN)[7] [54], and; 7) NWPU-RESISC45 (NWPU)[8] [55]. Key information of the seven benchmark datasets is summarised in Table 1 for visual clarity. Note that in this paper, the NWPU dataset was used to prime EFNEs for latent representation learning. The structures and parameters of the trained EFNEs remained fixed for all numerical experiments. The other six datasets were used for performance comparison against the state-of-the-art approaches in the literature.

Table 1. Key information of benchmark datasets for experimental investigation

Dataset	Images	Categories	Images per Category	Image Size	Resolution (m)
OPT	1860	31	60	256 × 256	0.1 to 10
WHU	1005	19	50 to 61	600 × 600	up to 0.5
UCM	2100	21	100	256 × 256	0.3
AID	10,000	30	220 to 420	600 × 600	0.3 to 3
RSI	24,747	35	198 to 1331	256 × 256	3 to 5
PTN	30,400	38	800	256 × 256	3
NWPU	31,500	45	700	256 × 256	0.2 to 30

Example images of the seven datasets are given by Fig. 4.



[1] Available at: https://github.com/Gu-X/Multi-View-Evolving-Fuzzy-Neural-Encoder-Framework
[2] Available at: https://drive.google.com/file/d/1Fk9a0DW8UyyQsR8dP2Qdakmr69NVBhq9/edit
[3] Available at: https://captain-whu.github.io/BED4RS/#
[4] Available at: http://weegee.vision.ucmerced.edu/datasets/landuse.html
[5] Available at: https://captain-whu.github.io/AID/
[6] Available at: https://github.com/lehaifeng/RSI-CB
[7] Available at: https://sites.google.com/view/zhouwx/dataset
[8] Available at: https://www.tensorflow.org/datasets/catalog/resisc45

Fig. 4. Example images from the seven benchmark datasets

B. System implementation. The proposed MVEFNE framework employs the following four pretrained CNN models on ImageNet-1K dataset as backbones (namely, $H = 4$) for feature extraction from remote sensing images, including: 1) ConvNeXtSmall; 2) ConvNeXtBase; 3) ConvNeXtLarge, and 4) ConvNeXtXLarge. These models were chosen as backbones because they are the cutting-edge CNN architectures with the highest accuracy on the ImageNet validation dataset [56]. They offer great accuracy, scalability and robustness across all major benchmarks, competing favourably with Transformers while maintaining higher computational efficiency.

During the experiments, the final prediction layers of the four models were removed, and the activations from the last 2D average pooling layers were used as image embeddings. Each image was represented by four different feature vectors extracted by the four CNN backbones with different architectures, namely, ConvNeXtSmall extracts a 768×1 dimensional feature vector, ConvNeXtBase extracts a 1024×1 dimensional feature vector, ConvNeXtLarge extracts a 1536×1 dimensional feature vector, and ConvNeXtXLarge extracts a 2048×1 dimensional feature vector. To ensure consistency in feature extraction, all images were initially resized to 248×248 pixels. From each resized image, five 224×224 segments were created by cropping the centre and four corners. These five segments were then flipped horizontally to generate five additional segments, resulting in a total of 10 segments per image. The final image embeddings were obtained by averaging the feature vectors extracted from these 10 segments using the CNN backbones [3].

The four supervised EFNEs and four self-supervised EFNEs within the MVEFNE framework were constructed based on MEFNN$^+$ with a two-layer architecture ($L = 2$) following the default setting given by [37], where the output size of the first layer is set as $W^1 = 256$. The eight EFNEs were trained on the NWPU dataset for 100 epochs and the chunk size of the input data chunks was set as 4 uniformly. The NWPU dataset was chosen for priming the EFNEs because it is one of the largest available remote sensing datasets and contains a wide variety of land-use categories. The supervised EFNEs learn to map input image embeddings to their corresponding one-hot encoded class labels, while the self-supervised EFNEs learn to reconstruct the input image embeddings. After training, the structures and parameters of the EFNEs were fixed, and their last layer removed. This allowed the trained EFNEs to compress the image embeddings extracted by the CNN backbones into 256×1 dimensional latent representations in a more compact form with enhanced descriptive capabilities. Note that, in running the experiments, the EFNEs pre-trained on NWPU were used for latent representation extraction without further fine-tuning on each problem.

The feature fusion module combines the latent representations learned by the eight EFNEs through concatenation, which preserves the full set of discriminative features from each EFNE. Consequently, for each image, the feature fusion module produces a 2048×1 dimensional fused representation. However, as concatenation significantly increases dimensionality, one may consider alternative fusion strategies such as averaging, which maintains a lower-dimensional representation, but may reduce discriminative power.

Finally, MVEFNE leverages a standard linear SVM classifier [46] to perform classification. In each experiment, the SVM classifier was firstly trained with the representations learned from training images and then used for predicting the class labels of testing images based on the representations extracted from them.

4.2. Performance Demonstration and Comparison

In this subsection, the classification performance of the proposed MVEFNE framework was evaluated on OPT, WHU, UCM, AID, RSI and PTN datasets and compared against state-of-the-art methods under the commonly used experimental protocols. Specifically, the train/test split for OPT was set to 8:2. Two train/test splits were considered for WHU, namely, 4:6 and 6:4. For UCM, the train/test splits were 5:5 and 8:2, and the train/test splits were set to 2:8 and 5:5 for AID. The train/test split for RSI was set to 5:5. Three different train/test splits were considered for PTN, which include 2:8, 5:5 and 8:2. In the numerical examples presented in this paper, the overall classification accuracy on the testing images was used as the primary performance measure [57].

A. Results on the OPT dataset. OPT contains 31 scene categories, but with a relatively smaller number of images in each category. The combination of high category diversity and limited images per category makes classification of this dataset particularly challenging. The experimental results obtained by MVEFNE are reported in Table 2. The results of the state-of-the-art methods obtained from the literature [32], [57]–[61] are reported in the same table for benchmark comparison. As shown in Table 2, during the experiments, MVEFNE achieves the highest classification accuracy of 99.81% on the testing images of the OPT dataset, surpassing the runner-up HGTNet [57] by more than 3%.

Table 2. Performance comparison on OPT dataset

Method	OPT	Publication
	8:2	Year
MVEFNE	**0.9981±0.0021**	-
Fine-tune AlexNet [62]	0.8122±0.0019	2017
Fine-tune GoogLeNet [62]	0.8257±0.0012	
Fine-tune VGGNet [62]	0.8745±0.0045	
ARCNet-AlexNet [51]	0.8575±0.0035	2019
ARCNet-VGG16 [51]	0.9270±0.0035	
ARCNet-ResNet34 [51]	0.9128±0.0045	
EfficientNet-B0 [63]	0.9397±0.0012	
EfficientNet-B3 [63]	0.9451±0.0075	
GBNet [64]	0.9328±0.0027	2020
Ghost-ResNet50 [65]	0.9473±0.0058	
EfficientNetB3-Basic [58]	0.9476±0.0026	2021
EfficientNetB3-Attn-2 [58]	0.9586±0.0022	
ViT-Base [35]	0.8973±0.0012	
ViT-Large [35]	0.9114±0.0026	
DeiT-Base [66]	0.9309±0.0035	
TRS [32]	0.9597±0.0013	
IDCCP-ResNet50-512 [67]	0.9489±0.0022	
MopNet-GCN-ResNet50 [68]	0.9534±0.0031	2022
MopNet-GAT-ResNet50 [68]	0.9606±0.0031	
RANet [59]	0.9461±0.0018	
MLCG-ResNet50 [33]	0.9527±0.0036	2023
LTNet [33]	0.9570±0.0029	
HGTNet [57]	0.9633±0.0015	2024
DSDNet-ViT-B [60]	0.9702±0.0020	2025
ACDR-CRAFF [61]	0.9575±0.0026	

B. Results on the WHU dataset. WHU consists of 1005 images distributed evenly in 19 scene categories. Images of this dataset are highly varied in terms of illumination, scale, clarity, etc. and, therefore, present a difficult benchmark problem. The result obtained by MVEFNE on the WHU dataset is reported in Table 3 and compared against a number of state-of-the-art methods in the literature, where the results of the comparative methods were obtained directly from [30], [58], [59], [64], [69]. As shown in Table 3, MVEFNE achieves the highest classification accuracy on the WHU dataset when using a 4:6 train/test split. However, its performance ranks fifth when the split is set to 6:4. This example, along with the results in Table 2, demonstrates that MVEFNE is particularly effective in scenarios where only a limited number of training samples per category are available.

Table 3. Performance comparison on WHU dataset

Method	WHU		Publication
	4:6	6:4	Year
MVEFNE	**0.9872±0.0028**	0.9893±0.0051	-
GoogLeNet [12]	0.9312±0.0082	0.9417±0.0133	2017
CaffeNet [12]	0.9511±0.0120	0.9624±0.0056	
VGG-VD-16 [12]	0.9544±0.0060	0.9605±0.0091	
SalM^{3}LBP-CLM [70]	0.9535±0.0076	0.9638±0.0082	
Two-stream fusion [71]	0.9823±0.0056	0.9892±0.0052	2018
TEX-Net-LF [72]	0.9761±0.0052	0.9800±0.0056	
ARCNet-AlexNet [51]	0.9750±0.0049	**0.9975±0.0025**	2019
Fine-tune MobileNet V2 [69]	0.9682±0.0035	0.9814±0.0033	
SE-MDPMNet [69]	0.9846±0.0021	0.9897±0.0024	
GBNet [64]	0.9732±0.0032	0.9925±0.0050	2020
EfficientNetB3-Basic [58]	0.9728±0.0024	0.9768±0.0010	2021
EfficientNetB3-Attn-2 [58]	0.9860±0.0040	0.9868±0.0093	

RANet [59]	0.9798±0.0017	0.9897±0.0021	2022
MDRCN [30]	0.9866±0.0033	0.9949±0.0026	2024

C. Results on the UCM dataset. UCM consists of images of 21 land-use categories with 100 images in each. Some of the categories of this dataset are highly overlapping (e.g., sparse, medium, and dense residential). The classification performance of MVEFNE on the UCM dataset is reported in Table 4. Its results are also compared with a wide range of state-of-the-art methods, with their reported performances obtained directly from the literature [2], [8], [9], [18], [30], [36], [60], [61], [73]–[78]. One can see from Table 4 that MSCN [76] has the highest classification accuracies under both split ratios, namely, 99.26% and 99.95%. In contrast, the classification accuracy rates achieved by MVEFNE under the two split ratios are 97.50% and 98.21%, which are less than 2% lower than the best results.

Table 4. Performance comparison on UCM dataset

Method	UCM		Publication Year
	5:5	8:2	
MVEFNE	0.9750±0.0042	0.9821±0.0047	-
GoogLeNet [12]	0.9270±0.0060	0.9431±0.0089	2017
CaffeNet [12]	0.9398±0.0067	0.9502±0.0081	
VGG-VD-16 [12]	0.9414±0.0069	0.9521±0.0120	
SalM^{3}LBP-CLM [70]	0.9421±0.0075	0.9575±0.0080	
Two-stream fusion [71]	0.9697±0.0075	0.9802±0.0103	2018
ARCNet-AlexNet [51]	0.9681±0.0014	0.9912±0.0040	2019
GBNet [64]	0.9705±0.0019	0.9857±0.0048	2020
CAD [18]	0.9857±0.0033	0.9966±0.0027	
TRS [32]	0.9876±0.0013	0.9952±0.0017	2021
EfficientNetB3-Basic [58]	0.9763±0.0006	0.9873±0.0020	
EfficientNetB3-Attn-2 [58]	0.9790±0.0036	0.9912±0.0022	
EFPN-DSE-TDFF [2]	0.9619±0.0013	0.9914±0.0022	
CSDS [8]	0.9848±0.0021	0.9952±0.0013	
RANet [59]	0.9780±0.0019	0.9927±0.0024	2022
T-CNN [22]	-	0.9933±0.0011	
SCViT [31]	0.9890±0.0019	0.9957±0.0031	
HHTL [36]	0.9887±0.0028	0.9948±0.0025	
EMTCAL [79]	0.9867±0.0016	0.9957±0.0028	
CSCANet [6]	-	0.9976±0.0016	2023
TDFE-DAA [29]	0.9732±0.0056	0.9905±0.0072	
EMSCNet (ResNet-50) [9]	0.9870±0.0046	0.9944±0.0016	
EMSCNet (ViT-B) [9]	0.9910±0.0020	0.9921±0.0017	
LDBST [77]	0.9876±0.0038	0.9952±0.0024	
AFER [80]	0.9774±0.0117	0.9938±0.0011	2024
Scene-vector (EfficientNet-B2) [81]	-	0.9988±0.0012	
Scene-vector (EfficientNet-B7) [81]	-	0.9982±0.0018	
MDRCN [30]	0.9857±0.0019	0.9964±0.0012	
ResNet50+HFAM [4]	0.9853±0.0014	0.9943±0.0013	
STConvNext [73]	0.9667	0.9881	2025
MLCMFNet [74]	0.9889±0.0013	0.9991±0.0010	
STMSF [75]	0.9901±0.0031	0.9958±0.0023	
MSCN [76]	**0.9926±0.0007**	**0.9995±0.0010**	
DSDNet-ViT-B [60]	0.9911±0.0013	0.9975±0.0012	
ACDR-CRAFF [61]	0.9905±0.0014	0.9966±0.0013	

D. Results on the AID dataset. AID consists of 10,000 images, distributed in 30 scene categories. Compared to the OPT, WHU, and UCM datasets, AID presents a greater challenge due to its unique characteristics, including high intra-class variation, low inter-class dissimilarity, and relatively larger scale. The classification performance of MVEFNE on AID is reported in Table 5 and compared with a wide range of state-of-the-art methods. The reported results of the comparative algorithms in Table 5 were obtained directly from the literature [8], [9], [18], [28], [36], [69], [73]–[77], [80]–[83]. It can be seen from Table 5 that MVEFNE achieves the highest classification accuracy on the AID dataset, namely, 98.08% under the split ratio of 2:8 and 98.66% under the split ratio of 5:5, outperforming the runner-up Scene-vector (EfficientNet-B7) [81] by 0.90% and 0.25% respectively.

Table 5. Performance comparison on AID dataset

Method	AID		Publication
	2:8	5:5	Year
MVEFNE	**0.9808±0.0016**	**0.9866±0.0015**	-
GoogLeNet [12]	0.8344±0.0040	0.8639±0.0055	2017
CaffeNet [12]	0.8686±0.0047	0.8953±0.0031	
VGG-VD-16 [12]	0.8659±0.0029	0.8964±0.0036	
SalM^{3}LBP-CLM [70]	0.8692±0.0035	0.8976±0.0045	
TEX-Net-LF [72]	0.9087±0.0011	0.9296±0.0018	2018
Two-stream fusion [71]	0.9232±0.0041	0.9458±0.0025	
ARCNet-AlexNet [51]	0.8875±0.0040	0.9310±0.0055	2019
Fine-tune MobileNet V2 [69]	0.9413±0.0028	0.9596±0.0027	
SE-MDPMNet [69]	0.9468±0.0017	0.9714±0.0015	
GANet [28]	0.9284±0.0022	0.9602±0.0022	
LANet [28]	0.9372±0.0014	0.9652±0.0018	
GLANet [28]	0.9502±0.0028	0.9666±0.0019	
GBNet [64]	0.9220±0.0023	0.9548±0.0012	2020
CAD [18]	0.9573±0.0022	0.9716±0.0026	
TRS [32]	0.9554±0.0018	0.9848±0.0006	2021
EfficientNetB3-Basic [58]	0.9343±0.0033	0.9537±0.0041	
EfficientNetB3-Attn-2 [58]	0.9445±0.0073	0.9656±0.0012	
EFPN-DSE-TDFF [2]	0.9402±0.0021	0.9450±0.0030	
CSDS [8]	0.9429±0.0035	0.9670±0.0014	
ViT-Base [35]	0.9521±0.0016	0.9696±0.0022	
LCPP [84]	0.9096±0.0033	0.9312±0.0028	
RANet [59]	0.9271±0.0014	0.9531±0.0037	2022
T-CNN [22]	0.9455±0.0027	0.9672±0.0023	
MopNet-GCN-ResNet50 [68]	0.9553±0.0011	0.9711±0.0007	
MopNet-GAT-ResNet50 [68]	0.9516±0.0016	0.9675±0.0011	
SCViT [31]	0.9556±0.0017	0.9698±0.0016	
LGRIN [85]	0.9474±0.0023	0.9765±0.0025	
HHTL [36]	0.9562±0.0013	0.9688±0.0021	
EMTCAL [79]	0.9469±0.0014	0.9641±0.0023	
TDFE-DAA [29]	0.9215±0.0075	0.9536±0.0015	2023
EMSCNet (ResNet-50) [9]	0.9513±0.0010	0.9696±0.0010	
EMSCNet (ViT-B) [9]	0.9602±0.0018	0.9735±0.0017	
LDBST [77]	0.9510±0.0009	0.9684±0.0020	
AFER [80]	0.9494±0.0012	0.9766±0.0009	2024
Scene-vector (EfficientNet-B2) [81]	0.9615±0.0015	0.9799±0.0021	
Scene-vector (EfficientNet-B7) [81]	0.9718±0.0016	0.9841±0.0015	
MDRCN [30]	0.9364±0.0019	0.9566±0.0018	
MSHNet+VGG16 [86]	0.9437±0.0032	0.9508±0.0022	
MSHNet+swin transformer [86]	0.9521±0.0029	0.9690±0.0030	
GLFSA [87]	0.9509±0.0015	0.9731±0.0023	
ResNet50+HFAM [4]	0.9471±0.0006	0.9673±0.0004	
STConvNext [73]	0.9412	0.9625	2025
ATMformer [82]	0.9473±0.0012	0.9608±0.0010	
MLCMFNet [74]	0.9613±0.0015	0.9826±0.0013	
STMSF [75]	0.9615±0.0016	0.9751±0.0037	
MSCN [76]	0.9586±0.0016	0.9746±0.0012	
DSDNet-ViT-B [60]	0.9681±0.0007	0.9819±0.0009	
ACDR-CRAFF [61]	0.9491±0.0011	0.9656±0.0016	
LiAP-ResNet [83]	0.9388	-	

E. Results on the RSI and PTN datasets. RSI and PTN are large-scale, fine-resolution remote sensing datasets for scene classification. RSI consists of 24,747 images across 35 scene categories, with the number of images per category ranging from 198 to 1331. PTN contains 30,400 images, evenly distributed across 38 scene categories, with 800 images per category. Both datasets exhibit diverse class distributions and large-scale coverage, making them more challenging than AID. The classification performance of MVEFNE on RSI and PTN is reported in Tables 6 and 7, respectively. For benchmark comparison, the results of state-of-the-art methods on these datasets under the same experimental protocols were obtained directly from the literature [28], [83], [87] and tabulated in the respective tables. One can see from Table 6 that the classification accuracy of MVEFNE on RSI surpasses the best-performing comparative method GLFSA [87] by more than 2%. On the other hand, Table 7 shows that MVEFNE only outperforms GANet [28] on the PTN dataset under the splitting ratios of 2:8 and 5:5, and its performance also surpasses LANet [28] under the splitting ratio of 5:5. The performance gap between MVEFNE and state-of-the-art methods on the PTN and UCM datasets may be attributed to the fact that the CNN backbones and EFNEs used for feature extraction and representation learning were not fine-tuned on these datasets. As a result, the learned representations may not be the most discriminative for accurately classifying images into their respective scene categories.

Table 6. Performance comparison on RSI dataset

Method	RSI	Publication
	5:5	Year
MVEFNE	**0.9984±0.0003**	-
GoogLeNet [12]	0.8987±0.0036	2017
CaffeNet [12]	0.9137±0.0023	
VGG-VD-16 [12]	0.9244±0.0025	
TEX-Net-LF [72]	0.9531±0.0015	2018
Two-stream fusion [71]	0.9457±0.0025	
Fine-tune MobileNet V2 [69]	0.9583±0.0026	2019
SE-MDPMNet [69]	0.9635±0.0026	
LCPP [84]	0.9372±0.0037	2021
LGRIN [85]	0.9755±0.0023	2022
GLFSA [87]	0.9772±0.0025	2024
ResNet50+HFAM [4]	0.9765±0.0022	

Table 7. Performance comparison on PTN dataset

Method	PTN			Publication
	2:8	5:5	8:2	Year
MVEFNE	0.9855±0.0007	0.9890±0.0008	0.9904±0.0008	-
GANet [28]	0.9750±0.0026	0.9878±0.0028	0.9970±0.0026	2019
LANet [28]	0.9864±0.0023	0.9882±0.0024	0.9961±0.0023	
GLANet [28]	0.9946±0.0013	0.9965±0.0011	0.9970±0.0015	
AFER [80]	**0.9956±0.0003**	**0.9974±0.0002**	**0.9980±0.0001**	2024
LiAP-ResNet [83]	**-**	0.9956	0.9979	2025

F. Evaluation of Computational Complexity. The average numbers of fuzzy rules learned by the EFNEs connected to different CNN backbones are reported in Table 8. The number of parameters associated with each fuzzy rule and the corresponding number of floating-point operations (FLOPs) that each rule required for inference are also reported in the same table. Furthermore, the parameter counts and FLOPs of the four CNN backbones employed by MVEFNE, as well as those of the six DNN models used for performance comparison, are reported in Table 9.

Table 8. Complexity of EFNEs

Backbone	Embedding	EFNE		
	Dimensionality	Rules	Parameters per Rule (M)	FLOPs per Rule (M)
ConvNeXtXLarge	2048×1	43	0.5	1.1
ConvNeXtLarge	1536×1	65	0.4	0.8
ConvNeXtBase	1024×1	79	0.3	0.5

ConvNeXtSmall	768×1	23	0.2	0.4

Table 9. Complexity of DNN models

Model	Parameters (M)	FLOPs (G)
ConvNeXtXLarge	350.2	121.2
ConvNeXtLarge	197.8	68.2
ConvNeXtBase	88.6	30.4
ConvNeXtSmall	50.2	17.1
CaffeNet [12]	60.9	0.7
VGG-VD-16 [12]	138.4	15.5
EMTCAL [79]	27.8	4.3
LDBST [77]	9.3	2.6
STConvNext [73]	10.5	2.2
LiAP-ResNet [83]	16.0	1.3

It can be observed from Table 8 that both the parameter counts and FLOPs of the EFNEs vary with the dimensionality of the image embeddings extracted by their respective CNN backbones. This is because the primary role of EFNEs is to compress the image embeddings into more compact and descriptive latent representations. Consequently, higher-dimensional image embeddings lead to a greater number of parameters per fuzzy rule and higher FLOPs during inference. Moreover, thanks to the adaptive activation control scheme, the effective computational complexity of EFNEs is further reduced, as these less activated fuzzy rules are temporarily excluded during each processing cycle.

Comparison between Tables 8 and 9 reveals that the computational complexity of EFNEs is significantly lower than that of the DNNs. Therefore, the overall computational complexity of MVEFNE is determined primarily by the CNN backbones that it employs for feature extraction. To reduce complexity, one may consider replacing the currently employed CNNs with lightweight models [88] or using fewer CNNs for feature extraction.

4.3. Ablation Study and Additional Analysis

In this subsection, an ablation analysis was carried out to highlight the advantages of multi-view feature extraction and the integration of supervised and self-supervised representation learning in the proposed MVEFNE framework for remote sensing scene classification. Additional experiments were conducted to justify the choice of the concatenation operation for feature fusion and the use of the SVM classifier for classification in the proposed framework. Finally, the IF-THEN rules learned by the EFNEs are presented as examples for better illustration.

First, the classification performances of MVEFNE with different numbers of views were evaluated on the six benchmark image sets using the same experimental protocols as in the previous examples. In particular, among the four CNN backbones utilised in this study, ConvNeXtXLarge achieved the highest Top-1 accuracy on the ImageNet validation dataset, followed by ConvNeXtLarge, while ConvNeXtSmall achieved the lowest accuracy. Accordingly, the following three variants of MVEFNE were considered for this ablation analysis:

- The single-view variant employs only ConvNeXtXLarge as CNN backbones.
- The two-view variant employs both ConvNeXtXLarge and ConvNeXtLarge as CNN backbones.
- The three-view variant employs ConvNeXtXLarge, ConvNeXtLarge, and ConvNeXtBase as CNN backbones.

The results of MVEFNE and its variants are reported in Table 10. For clearer demonstration, the performance improvements achieved by adding more views on each dataset under the corresponding split ratio(s) were calculated using Eq. (29), following common practice [89]. The average percentage improvements ($\bar{\Delta}_{x\rightarrow y}$) are summarised in Table 11.

$$\Delta_{x\rightarrow y}=\frac{1-\mathrm{acc}_x}{1-\mathrm{acc}_y}\times 100\% \qquad (29)$$

where $\Delta_{x\rightarrow y}$ is the percentage improvement from x views to y views, and; acc_x denotes the classification accuracy of MVEFNE with x views ($x = 1,2,3,4$).

As shown in Tables 10 and 11, incorporating additional views during the feature extraction process consistently leads to increases in accuracy across all experiments. The most substantial gain is observed when increasing the number of views from one (ConvNeXtXLarge) to two (ConvNeXtXLarge and ConvNeXtLarge), resulting in an average improvement of 138%. Adding further views beyond two continues to yield at least a 19% improvement per additional view. Although the magnitude of improvement decreases as more views are introduced, this is attributed primarily to the relatively weaker feature representation capability of ConvNeXtBase and ConvNeXtSmall compared with ConvNeXtXLarge and ConvNeXtLarge. These results demonstrate the effectiveness of multi-view feature extraction in increasing the classification accuracy of MVEFNE.

As suggested by Tables 8 and 9, the increases in accuracy achieved by incorporating additional views come at the cost of increased computational complexity, arising primarily from feature extraction and dimensionality compression. However, these costs can be mitigated by employing lightweight feature extractors. Moreover, the training and inference processes of MVEFNE can be further accelerated through parallel computation.

Table 10. Performance of MVEFNE with varying numbers of views

Dataset	Split Ratio	MVEFNE			
		Single-View	Two-View	Three-View	Four-View
OPT	8:2	0.9911±0.0040	0.9946±0.0034	0.9941±0.0029	0.9981±0.0021
WHU	4:6	0.9745±0.0048	0.9831±0.0039	0.9866±0.0061	0.9872±0.0028
	6:4	0.9776±0.0039	0.9878±0.0041	0.9883±0.0032	0.9893±0.0051
UCM	5:5	0.9510±0.0046	0.9677±0.0057	0.9723±0.0023	0.9750±0.0042
	8:2	0.9593±0.0069	0.9721±0.0053	0.9805±0.0046	0.9821±0.0047
AID	2:8	0.9130±0.0030	0.9744±0.0015	0.9796±0.0014	0.9808±0.0016
	5:5	0.9263±0.0021	0.9812±0.0015	0.9847±0.0016	0.9866±0.0015
RSI	5:5	0.9802±0.0008	0.9970±0.0002	0.9979±0.0004	0.9984±0.0003
PTN	2:8	0.9712±0.0009	0.9799±0.0007	0.9826±0.0006	0.9855±0.0007
	5:5	0.9786±0.0010	0.9847±0.0007	0.9866±0.0006	0.9890±0.0008
	8:2	0.9809±0.0013	0.9871±0.0015	0.9880±0.0011	0.9904±0.0008

Table 11. Average percentage performance improvements of MVEFNE by adding more views

$\bar{\Delta}$	x		
	2	3	4
$1 \rightarrow x$	138.18%	195.78%	275.74%
$2 \rightarrow x$	-	19.09%	53.50%
$3 \rightarrow x$	-	-	32.98%

Second, the classification performances of MVEFNE using supervised representation learning, self-supervised representation learning and a combination of both were evaluated on the six datasets, with results presented in Table 12. One can see from this numerical example that MVEFNE leveraging only supervised EFNEs outperforms its counterpart leveraging only self-supervised EFNEs on small-scale and medium-scale datasets. This is because supervised representation learning guides the EFNEs to learn the discriminative features that help classify images into different scene categories. On the other hand, self-supervised learning forces the EFNEs to learn how to reconstruct the input feature vectors, without focusing explicitly on capturing interclass dissimilarities. As a result, when there are fewer training images, the latent representations learned by the EFNEs trained in a supervised setting enable MVEFNE to achieve higher classification accuracy. However, this advantage becomes less prominent when there are sufficient training images available. By combining the two representation learning approaches, MVEFNE achieves higher classification accuracy across all the experiments, showing the effectiveness of such integration in enhancing the classification performance of the proposed framework.

Table 12. Performance of MVEFNE using supervised, self-supervised, and combined representation learning

Dataset	Split Ratio	MVEFNE		
		Supervised	Self-Supervised	Combined
OPT	8:2	0.9970±0.0025	0.8922±0.0199	0.9981±0.0021
WHU	4:6	0.9861±0.0035	0.9706±0.0070	0.9872±0.0028
	6:4	0.9878±0.0039	0.9729±0.0039	0.9893±0.0051
UCM	5:5	0.9751±0.0026	0.9592±0.0047	0.9750±0.0042
	8:2	0.9774±0.0065	0.9620±0.0087	0.9821±0.0047
AID	2:8	0.9751±0.0014	0.9676±0.0019	0.9808±0.0016

	5:5	0.9810±0.0015	0.9793±0.0016	0.9866±0.0015
RSI	5:5	0.9974±0.0002	0.9969±0.0004	0.9984±0.0003
PTN	2:8	0.9821±0.0007	0.9820±0.0007	0.9855±0.0007
	5:5	0.9850±0.0009	0.9870±0.0007	0.9890±0.0008
	8:2	0.9860±0.0014	0.9888±0.0013	0.9904±0.0008

Then, the classification accuracy of MVEFNE using concatenation for feature fusion is compared with its counterpart using averaging for feature fusion. The classification performances of the two MVEFNE frameworks are tabulated in Table 13. It can be seen from Table 13 that concatenation can better preserve the discriminative features of the latent representations during fusion compared to the averaging operation, enabling MVEFNE to achieve higher classification accuracy across the six datasets. On the other hand, it is also important to note that concatenation can result in significantly higher dimensionality of the fused representations than averaging, which may lead to higher computational complexity and make the classifier more prone to overfitting.

Table 13. Performance of MVEFNE using averaging and concatenation for feature fusion

Dataset	Split Ratio	MVEFNE	
		Averaging	Concatenation
OPT	8:2	0.9949±0.0022	0.9981±0.0021
WHU	4:6	0.9841±0.0034	0.9872±0.0028
	6:4	0.9836±0.0056	0.9893±0.0051
UCM	5:5	0.9678±0.0046	0.9750±0.0042
	8:2	0.9695±0.0074	0.9821±0.0047
AID	2:8	0.9707±0.0014	0.9808±0.0016
	5:5	0.9775±0.0012	0.9866±0.0015
RSI	5:5	0.9957±0.0004	0.9984±0.0003
PTN	2:8	0.9792±0.0007	0.9855±0.0007
	5:5	0.9841±0.0008	0.9890±0.0008
	8:2	0.9857±0.0016	0.9904±0.0008

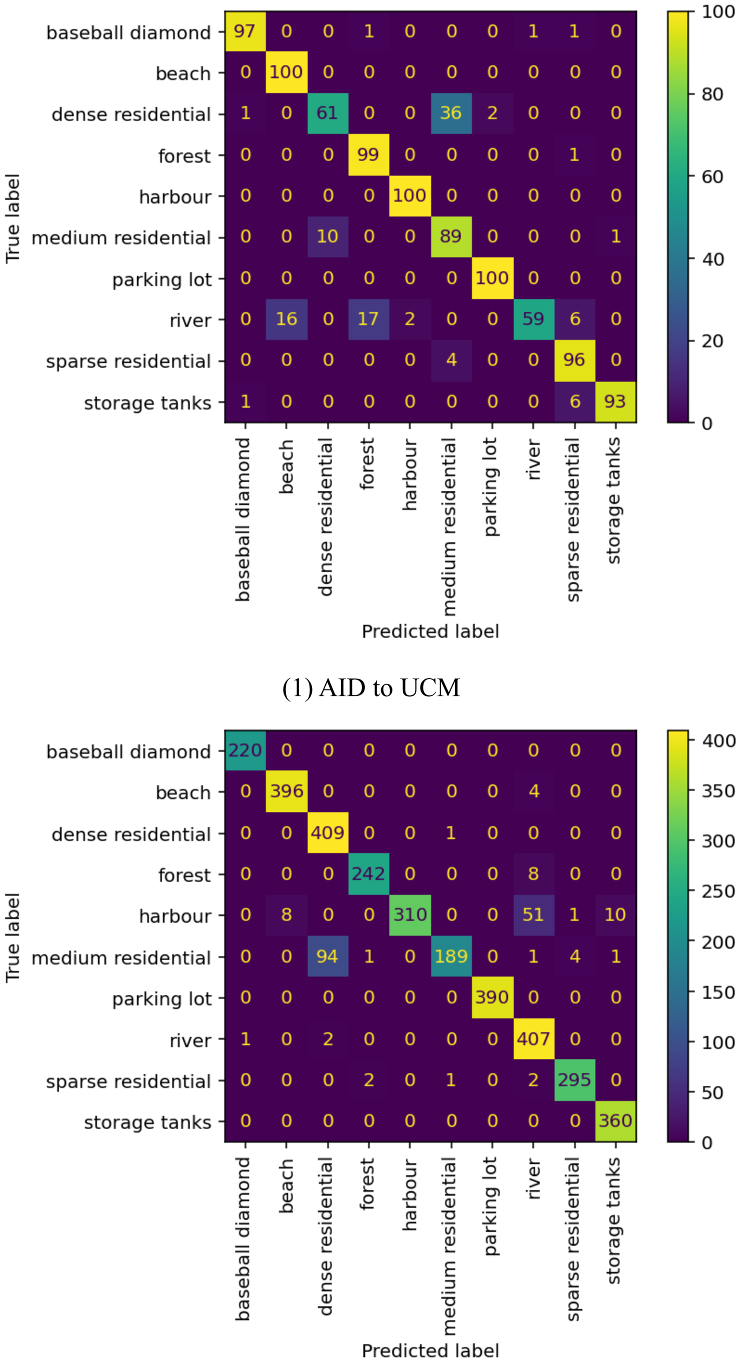
4.4. Demonstration of Generalisation Capability

In this example, the generalisation capability of the proposed MVEFNE is evaluated using the UCM and AID datasets. These two datasets share 10 common land-use categories, which include baseball diamond (baseball field), beach, dense residential, forest, harbour (port), medium residential, parking lot (parking), river, sparse residential and storage tanks. During the experiments, MVEFNE was firstly trained with images of these 10 land-use categories from AID and tested subsequently on images of the same categories from UCM. Next, the experiments were repeated with MVEFNE trained with the images from UCM and tested subsequently using images from AID. The obtained results from the cross-dataset experiments in terms of accuracy are reported in Table 14, and the corresponding confusion matrices are presented in Fig. 5.

Table 14. Classification performance of MVEFNE in cross-dataset experiments.

Training Set	Testing Set	Accuracy
AID	UCM	0.8940
UCM	AID	0.9441
AID	OPT	0.9812
OPT	AID	0.9588
OPT	UCM	0.9133
UCM	OPT	0.9669

One can see from Table 14 that MVEFNE achieves high classification accuracies in the cross-dataset experiments, exceeding 89%. Specifically, Fig. 5(a) shows that MVEFNE trained using images from the AID dataset tends to misclassify dense residential and river scene images of the UCM dataset as medium residential, beach and forest, respectively. In contrast, Fig. 5(b) shows that MVEFNE trained using the UCM dataset is more likely to misclassify harbour and medium residential scene images of the AID dataset as river and dense residential, respectively. These misclassifications are caused primarily by the substantial intraclass variability and interclass similarity between images of corresponding categories in the two datasets, as illustrated in Fig. 6. For land-use categories exhibiting more distinctive visual patterns, MVEFNE is able to correctly classify most images.



(1) AID to UCM

(2) UCM to AID

Fig. 5. Confusion matrices obtained from cross-dataset experiments between the AID and UCM datasets

Additional cross-dataset experiments were carried out by further involving the OPT dataset. Specifically, OPT and AID share 16 common land-use categories: airport, baseball diamond (baseball field), beach, bridge, church, commercial area (commercial), dense residential, desert, forest, harbour (port), industrial area (industrial), meadow, medium residential, mountain, parking lot (parking) and railway (railway station). Similarly, OPT and UCM share 15 common categories: airplane, baseball diamond, beach, chaparral, dense residential, forest, freeway, golf course, harbour, intersection, medium residential, mobile home park, overpass, parking lot and runway. The results returned from the cross-dataset experiments between the OPT and AID datasets, and between the OPT and UCM datasets, are also reported in Table 14, where similarly high accuracy across both dataset pairs is observed, with classification accuracies exceeding 91%.

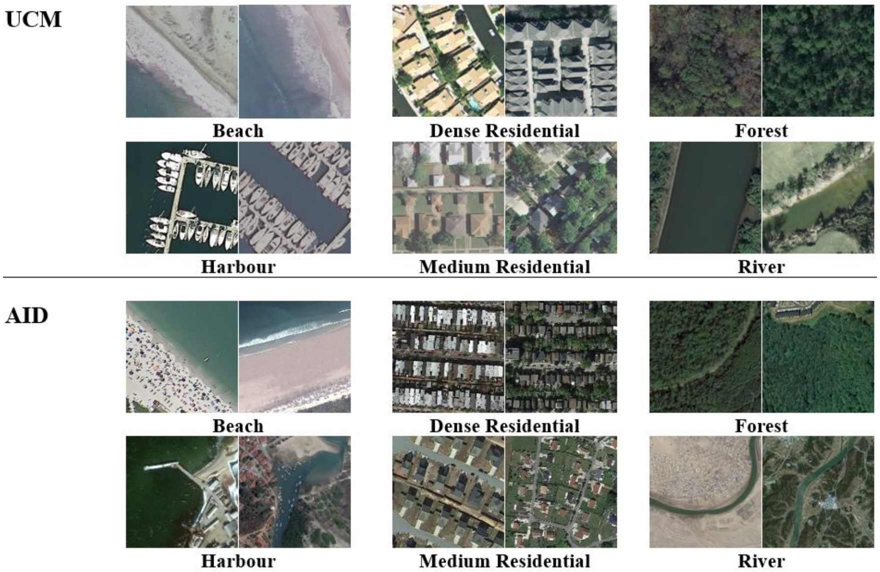


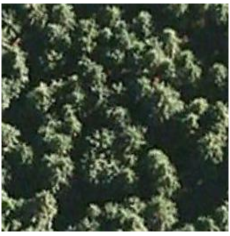
Fig. 6. Example images of the six land-use categories of UCM and AID datasets

4.5. Demonstration of Interpretability

Eight examples of IF-THEN fuzzy rules learned by supervised and self-supervised EFNEs from the embeddings of the NWPU image set, extracted using the ConvNeXtLarge backbone, are presented in Tables 15 and 16. It is worth noting that, since the learned prototypes are the embeddings of real images, the antecedent parts of the fuzzy rules are visualised using the corresponding images for clearer illustration.

It can be seen from Tables 15 and 16 that the eight rules learned by the two EFNEs capture the distinctive visual patterns exhibited by remote sensing images of eight different scene categories in the NWPU dataset. Specifically, the four fuzzy rules learned by the supervised EFNE correspond to the forest, wetland, beach and ground track field categories, respectively. The four rules learned by the self-supervised EFNE correspond to the overpass, desert, intersection and cloud categories, respectively. These examples clearly demonstrate that the antecedent part of each fuzzy rule corresponds to a distinct semantic concept represented in the feature space. The consequent part of each rule is a transformation matrix that projects the high-dimensional image embeddings into a lower-dimensional latent space.

Table 15. Examples of IF-THEN fuzzy rules learned by supervised EFNE

#	Rule
1	IF (**I**~) $THEN\left(\boldsymbol{y}_1=\sigma\left(\overbrace{\begin{bmatrix}0.0567 & -0.0044 & 0.0516 & \cdots & -0.1746 & 0.1549\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.0322 & -0.0531 & 0.0913 & \cdots & -0.3831 & 0.0440\end{bmatrix}}^{256\times1537}\overline{\boldsymbol{x}}\right)\right)$

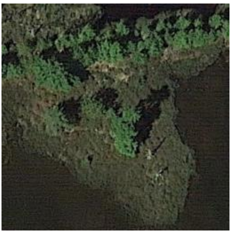
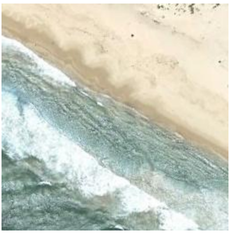
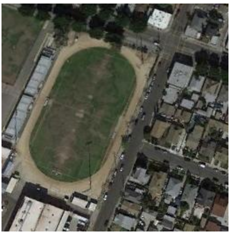
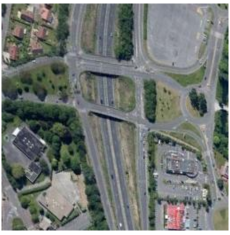
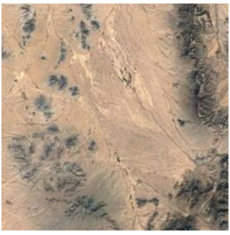
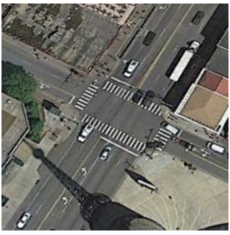
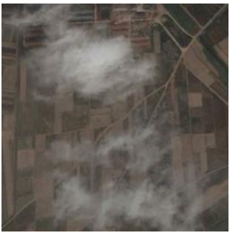
2	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_2=\sigma\left(\left[\overbrace{\begin{matrix}0.0934 & 0.0251 & 0.0928 & \cdots & -0.2760 & 0.1485\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.0435 & -0.0689 & 0.1475 & \cdots & -0.1377 & 0.0449\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$
3	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_3=\sigma\left(\left[\overbrace{\begin{matrix}-0.3278 & -0.0202 & 0.0643 & \cdots & -0.1935 & 0.1440\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.0305 & -0.0787 & 0.0912 & \cdots & -0.1219 & 0.0230\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$
4	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_4=\sigma\left(\left[\overbrace{\begin{matrix}0.1310 & 0.0400 & 0.0084 & \cdots & -0.3909 & 0.2410\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.0660 & -0.1129 & 0.1547 & \cdots & -0.1727 & -0.0308\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$

Table 16. Examples of IF-THEN fuzzy rules learned by self-supervised EFNE

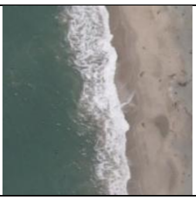
#	Rule
1	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_1=\sigma\left(\left[\overbrace{\begin{matrix}0.4034 & -0.0902 & -0.0790 & \cdots & -0.3719 & -0.3682\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.4386 & -0.0048 & 0.2436 & \cdots & -0.0180 & 0.7124\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$

2	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_2=\sigma\left(\left[\overbrace{\begin{matrix}-0.1045 & 0.0366 & 0.5691 & \cdots & -0.0079 & 0.1194\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.2337 & -0.0503 & 0.2726 & \cdots & 0.1001 & 0.8008\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$
3	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_3=\sigma\left(\left[\overbrace{\begin{matrix}-0.2543 & 0.4444 & 0.0722 & \cdots & -0.0241 & 0.5678\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0.1939 & -0.8002 & -3.4338 & \cdots & -1.2119 & -0.7518\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$
4	$IF\ (\mathbf{I}\sim$  $)$ $THEN\left(\boldsymbol{y}_4=\sigma\left(\left[\overbrace{\begin{matrix}-0.3166 & -0.2469 & 2.3040 & \cdots & -3.0853 & -0.1121\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -0.1062 & -0.6663 & -2.3563 & \cdots & -1.7632 & -0.7359\end{matrix}}^{256\times1537}\right]\overline{\boldsymbol{x}}\right)\right)$

The activations (normalised firing strengths) of the eight fuzzy rules to eight selected remote sensing images of the UCM and AID datasets are reported in Table 17. Specifically, the four images of UCM correspond to the airplane, beach, freeway and sparse residential categories, respectively. The four images of AID correspond to the bridge, farmland, pond and viaduct categories, respectively. It is worth noting that the reported normalised firing strengths do not sum to 1 because only a subset of fuzzy rules is presented here. The remaining rules in the EFNEs' rule bases also contribute to the normalisation.

As one can see from Table 17, among the four fuzzy rules learned by the supervised EFNE, the first rule produces lower activations (≤ 0.05) on all eight images, indicating that the visual pattern of its prototype is significantly different from those images. The second rule corresponds to the wetland category and, thus, it yields higher activations (≥ 0.15) on the images corresponding to the sparse residential, bridge and pond categories, reflecting their similar visual characteristics, such as water and vegetation textures. The third rule is associated with the beach category, and it exhibits the highest activation (0.27) on the beach image from the UCM dataset because of its strong visual similarity to the corresponding prototype. The fourth rule yields higher activations (≥ 0.15) on images corresponding to the freeway, sparse residential, bridge, farmland and pond categories. This is because the prototype of this rule shares similar elements such as grassland, forest, residential buildings, and road structures, with the five images. Similar observations can also be made for the activations produced by the four rules of the self-supervised EFNE. Overall, this table illustrates clearly how the learned fuzzy rules respond to different input images, offering valuable insights into how the EFNEs derive latent representations from image embeddings.

Table 14. Normalised firing strengths of the learned fuzzy rules in response to different input images.

EFNE	# Rule	UCM			
					
Supervised	1	0.0149	0.0206	0.0356	0.0440
	2	0.0618	0.0859	0.1029	0.1900
	3	0.0252	0.2700	0.0263	0.0319
	4	0.0834	0.0641	0.1798	0.1998
Self-Supervised	1	0.0620	0.0556	0.2031	0.1808
	2	0.0838	0.1273	0.0640	0.1112
	3	0.2170	0.0877	0.1588	0.1708
	4	0.0490	0.0475	0.0514	0.0768
EFNE	# Rule	AID			
Supervised	1	0.0449	0.0348	0.0327	0.0222
	2	0.1877	0.1183	0.1694	0.0646
	3	0.0308	0.0213	0.0319	0.0254
	4	0.1753	0.1842	0.2354	0.1134
Self-Supervised	1	0.1416	0.1126	0.1941	0.1823
	2	0.1130	0.1151	0.0968	0.0586
	3	0.1815	0.1760	0.1846	0.0906
	4	0.0775	0.0289	0.0638	0.2817

4.6. Discussion and Directions for Future research

To summarise, the systematic experiments carried out in this paper demonstrated the efficacy of the proposed MVEFNE for remote sensing scene classification. Numerical examples presented in Section 4.2 collectively demonstrated that, without any fine-tuning, MVEFNE achieved superior classification performance across six popular benchmark image sets, surpassing or at least on par with the best-performing methods in the literature. Under the commonly used experimental protocols, MVEFNE obtained the highest classification accuracy on OPT, WHU (under the split ratio of 4:6), AID and RSI, outperforming the runner up methods by 3.47%, 0.06%, 0.90%, 0.25% and 2.12%, respectively. The numerical examples presented in Section 4.3 further justified the effectiveness of multi-view feature extraction and the integration of supervised and self-supervised representation learning in the proposed MVEFNE framework for remote sensing scene classification. Additional experiments were conducted to justify the choice of the concatenation operation for feature fusion. Finally, the generalisation capability and interpretability of MVEFNE were highlighted by the numerical examples presented in Sections 4.4 and 4.5, respectively.

There are several considerations for future research:

First, the primary purpose of this paper was to present the concept and general principles of MVEFNE. Therefore, only one specific configuration was used across the numerical experiments. However, the proposed MVEFNE framework is highly flexible, offering numerous opportunities for further performance enhancements. One could incorporate more powerful pretrained CNN backbones for feature extraction to achieve greater classification accuracy. Alternatively, to increase the computational efficiency of MVEFNE, one may employ lightweight feature extractors or reduce the number of CNN backbones in the framework, as discussed earlier. In addition, the externally controlled parameters (e.g., L, W^1, δ_o, ρ_o) of EFNEs could be adjusted to facilitate learning more discriminative latent representations. The exploration and implementation of alternative feature fusion techniques for combining the learned latent representations could also improve the classification performance of MVEFNE.

Furthermore, different image pre-processing techniques (e.g., noise reduction and contrast enhancement), can be applied to enhance feature extraction.

Second, the EFNEs, as the core component of MVEFNE, were primed using image embeddings extracted by CNN backbones from the NWPU dataset. Each EFNE was trained independently using either a self-supervised or supervised approach, without any interaction with others. However, the lack of interaction may not be the best approach to maximising the classification performance of MVEFNE, because EFNEs could potentially benefit from exchanging information during the learning process. Therefore, a promising direction is to develop a joint representation learning scheme that facilitates information exchange between EFNEs. Additionally, fine-tuning the EFNEs for specific problems could further increase classification accuracy.

Last but not least, the proposed MVEFNE framework employed pretrained CNN backbones and EFNEs for latent representation extraction during the experiments without fine-tuning. The reduced need for training images enabled MVEFNE to achieve excellent performance on small-scale datasets. However, since MVEFNE utilises a standard SVM for classification, its performance deteriorates significantly when the number of labelled training images is insufficient to train a reliable prediction model. Hence, more advanced classifiers with enhanced generalisation capability could be leveraged to attain greater classification performance, even when there exists limited training data. Furthermore, semi-supervised learning techniques could be developed to mitigate the labelling bottleneck and increase the classification accuracy of MVEFNE in application scenarios where labelled data are scarce.

5. Conclusion

In this paper, a multi-view ensemble framework was presented, named MVEFNE, that integrates multiple pretrained CNN backbones with different architectures and EFNEs to learn informative latent representations from remote sensing images for scene classification. By leveraging transparent self-supervised and supervised MEFNNs as encoders, MVEFNE compresses the multi-view image embeddings extracted by CNN backbones into descriptive, low-dimensional latent representations using a set of human-interpretable IF-THEN rules. The compressed latent representations are further fused together to enhance the discrimination ability for scene classification. Systematic benchmark comparison demonstrated the superior performance of MVEFNE on several challenging benchmark scene classification datasets, without fine-tuning on specific problems. Notably, MVEFNE achieved average classification accuracies of 99.81%, 98.83%, 97.86%, 98.37%, 99.84% and 98.83% on the testing sets of the OPT, WHU, UCM, AID, RIS and PTN datasets, respectively, surpassing or matching multiple state-of-the-art approaches reported in the literature. An ablation study further quantitatively validated the efficacy of MVEFNE, which synergistically combines multi-view learning, supervised learning, and self-supervised learning to extract highly discriminative latent representations from remote sensing imagery, thereby facilitating accurate scene classification.

Acknowledgement

This research was supported by the Engineering and Physical Sciences Research Council under Grant EP/X027732/1.

Appendix A: Representation Learning Pipeline

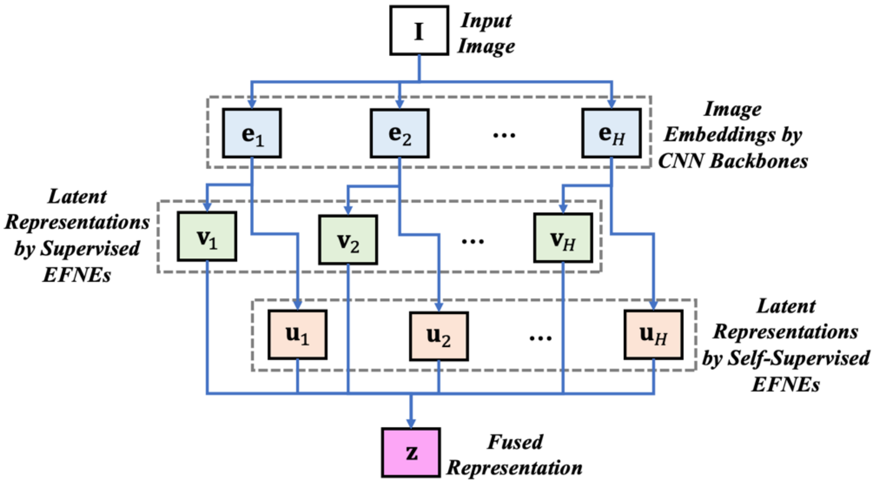


Fig. A1. Representation learning pipeline of MVEFNE

Appendix B: Classification Performance with Different Classifiers

The performances of MVEFNE leveraging different classifiers are reported in Table B1. Three alternative classifiers are considered in this numerical example:

- LR classifier.
- KNN classifier with $k = 3$.
- RF classifier consisting of 100 decision trees with maximum depth of 20.

One can see from the comparison that the linear SVM classifier enables MVEFNE to achieve higher classification accuracy across the six benchmark image sets, compared to the other three classifiers. This example justifies the use of SVM for classification in the proposed MVEFNE framework.

Table B1. Performance of MVEFNE leveraging different classifiers

Dataset	Split	MVEFNE			
	Ratio	LR	KNN	RF	SVM
OPT	8:2	0.9911±0.0027	0.9718±0.0100	0.9874±0.0042	0.9981±0.0021
WHU	4:6	0.9839±0.0046	0.9753±0.0045	0.9748±0.0071	0.9872±0.0028
	6:4	0.9866±0.0067	0.9806±0.0073	0.9761±0.0046	0.9893±0.0051
UCM	5:5	0.9764±0.0032	0.9519±0.0062	0.9570±0.0045	0.9750±0.0042
	8:2	0.9768±0.0073	0.9633±0.0088	0.9631±0.0061	0.9821±0.0047
AID	2:8	0.9815±0.0010	0.9413±0.0030	0.9546±0.0036	0.9808±0.0016
	5:5	0.9861±0.0016	0.9556±0.0028	0.9688±0.0029	0.9866±0.0015
RSI	5:5	0.9976±0.0003	0.9928±0.0006	0.9897±0.0005	0.9984±0.0003
PTN	2:8	0.9847±0.0006	0.9664±0.0010	0.9648±0.0014	0.9855±0.0007
	5:5	0.9879±0.0005	0.9751±0.0009	0.9724±0.0013	0.9890±0.0008
	8:2	0.9893±0.0014	0.9784±0.0017	0.9743±0.0016	0.9904±0.0008

References

[1] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: a survey," *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 6, p. e1264, 2018.

[2] X. Wang, S. Wang, C. Ning, and H. Zhou, "Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7918–7932, 2021.

[3] X. Gu, C. Zhang, Q. Shen, J. Han, P. P. Angelov, and P. M. Atkinson, "A self-training hierarchical prototype-based ensemble framework for remote sensing scene classification," *Inf. Fusion*, vol. 80, pp. 179–204, 2022.

[4] Q. Wan, Z. Xiao, Y. Yu, Z. Liu, K. Wang, and D. Li, "A hyperparameter-free attention module based on feature map mathematical calculation for remote-sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–18, 2024.

[5] X. Hou, Y. Bai, Y. Li, C. Shang, and Q. Shen, "High-resolution triplet network with dynamic multiscale feature for change detection on satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 103–115, 2021.

[6] Y. E. Hou, K. Yang, L. Dang, and Y. Liu, "Contextual spatial-channel attention network for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[7] X. Dong, C. Zhang, L. Fang, and Y. Yan, "A deep learning based framework for remote sensing image ground object segmentation," *Appl. Soft Comput.*, vol. 130, p. 109695, 2022.

[8] X. Wang, L. Yuan, H. Xu, and X. Wen, "CSDS: end-to-end aerial scenes classification with depthwise separable convolution and an attention mechanism," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 10484–10499, 2021.

[9] Y. Zhao, J. Liu, J. Yang, and Z. Wu, "EMSCNet: efficient multisample contrastive network for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–14, 2023.

[10] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, 2017.

[11] Y. Feng, L. Wang, J. Jin, and X. Wang, “A semi-supervised multi-source remote sensing image classification network based on adaptive pseudo-label generation,” *Appl. Soft Comput.*, vol. 175, p. 113055, 2025.

[12] G. Xia *et al.*, “AID: a benchmark dataset for performance evaluation of aerial scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, 2017.

[13] R. N. Patro, S. Subudhi, P. K. Biswal, and F. Dell’Acqua, “Dictionary-based classifiers for exploiting feature sequence information and their application to hyperspectral remotely sensed data,” *Int. J. Remote Sens.*, vol. 40, no. 13, pp. 4996–5024, 2019.

[14] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, “Support vector machine versus random forest for remote sensing image classification: a meta-analysis and systematic review,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 6308–6325, 2020.

[15] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[16] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[17] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[18] W. Tong, W. Chen, W. Han, X. Li, and L. Wang, “Channel-attention-based DenseNet network for remote sensing image scene classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 4121–4132, 2020.

[19] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.

[20] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features : spatial pyramid matching for recognizing natural scene categories,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.

[22] W. Wang, Y. Chen, and P. Ghamisi, “Transferring CNN with adaptive learning for remote sensing scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.

[23] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: a meta-analysis and review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 152, no. April, pp. 166–177, 2019.

[24] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, “Hyperspectral classification based on lightweight 3-D-CNN with transfer learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, 2019.

[25] S. Chaib, H. Liu, Y. Gu, and H. Yao, “Deep feature fusion for VHR remote sensing scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, 2017.

[26] X. Lu, H. Sun, and X. Zheng, “A feature aggregation convolutional neural network for remote sensing scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7894–7906, 2019.

[27] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, and Y. Zheng, “Siamese convolutional neural networks for remote sensing scene classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1200–1204, 2019.

[28] Y. Guo, J. Ji, X. Lu, H. Huo, T. Fang, and D. Li, “Global-local attention network for aerial scene classification,” *IEEE Access*, vol. 7, pp. 67200–67212, 2019.

[29] X. Chen, Z. Han, Y. Li, M. Ma, S. Mei, and W. Cheng, “Attention-aware deep feature embedding for remote sensing image scene classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 1171–1184, 2023.

[30] W. Dai, F. Shi, X. Wang, H. Xu, L. Yuan, and X. Wen, “A multi-scale dense residual correlation network for remote sensing scene classification,” *Sci. Rep.*, vol. 14, no. 1, p. 22197, 2024.

[31] P. Lv, W. Wu, Y. Zhong, F. Du, and L. Zhang, “SCViT: a spatial-channel feature preserving vision

transformer for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.

[32] J. Zhang, H. Zhao, and J. Li, "TRS: transformers for remote sensing scene classification," *Remote Sens.*, vol. 13, p. 4143, 2021.

[33] X. Huang, F. Liu, Y. Cui, P. Chen, L. Li, and P. Li, "Faster and better: a lightweight transformer network for remote sensing scene classification," *Remote Sens.*, vol. 15, no. 14, p. 3645, 2023.

[34] Y. Bazi, L. Bashmal, M. M. Al Rahhal, R. Al Dayil, and N. Al Ajlan, "Vision transformers for remote sensing image classification," *Remote Sens.*, vol. 13, no. 3, pp. 1–20, 2021.

[35] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[36] J. Ma, M. Li, X. Tang, X. Zhang, F. Liu, and L. Jiao, "Homo-heterogenous transformer learning framework for RS scene classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 2223–2239, 2022.

[37] X. Gu, P. Angelov, J. Han, and Q. Shen, "Multilayer evolving fuzzy neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 12, pp. 4158–4169, 2023.

[38] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4766–4775.

[39] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K. R. Müller, "Layer-wise relevance propagation: an overview," in *Explainable AI: interpreting, explaining and visualizing deep learning*, 2019, pp. 193–209.

[40] T. Szandala, "Enhancing deep neural network saliency visualizations with gradual extrapolation," *IEEE Access*, vol. 9, pp. 95155–95161, 2021.

[41] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[42] A. Barredo Arrieta *et al.*, "Explainable artificial antelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020.

[43] X. Gu, J. Han, Q. Shen, and P. P. Angelov, "Autonomous learning for fuzzy systems: a review," *Artif. Intell. Rev.*, pp. 1–47, 2022.

[44] X. Gu and Q. Shen, "A self-adaptive fuzzy learning system for streaming data prediction," *Inf. Sci. (Ny).*, vol. 579, pp. 623–647, 2021.

[45] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 287–296.

[46] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.

[47] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers-a tutorial," *ACM Computing Surveys*, vol. 54, no. 6. Association for Computing Machinery, Jul. 01, 2021.

[48] L. Breiman, "Random forests," *Mach. Learn. Proc.*, vol. 45, no. 1, pp. 5–32, 2001.

[49] W. Yang, X. Yin, and G. S. Xia, "Learning high-level features for satellite image classification with limited labeled samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4472–4482, 2015.

[50] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, 2018.

[51] Q. Wang, S. Liu, and J. Chanussot, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, 2019.

[52] G. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maitre, "Structural High-resolution Satellite image indexing," in *ISPRS, TC VII Symposium Part A: 100 Years ISPRS—Advancing Remote Sensing Science*, 2010, pp. 298–303.

[53] H. Li *et al.*, "RSI-CB: a large-scale remote sensing image classification benchmark using crowdsourced data," *Sensors*, vol. 20, no. 6, pp. 28–32, 2020.

[54] W. Zhou, S. Newsam, C. Li, and Z. Shao, “PatternNet: a benchmark dataset for performance evaluation of remote sensing image retrieval,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, 2018.

[55] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: benchmark and state of the art,” *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[56] Z. Liu, H. Mao, C. Wu, F. Christoph, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11976–11986, 2022, [Online]. Available: http://arxiv.org/abs/2201.03545.

[57] Z. Li *et al.*, “A hierarchical graph-enhanced transformer network for remote sensing scene classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 20315–20330, 2024.

[58] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, “Classification of remote sensing images using EfficientNet-B3 CNN model with attention,” *IEEE Access*, vol. 9, pp. 14078–14094, 2021.

[59] X. Wang, L. Duan, C. Ning, and H. Zhou, “Relation-attention networks for remote sensing scene classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 422–439, 2022.

[60] Y. Hu, L. Zhang, X. Luo, and X. Cao, “Diffusion self-distillation for remote sensing scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–15, 2025.

[61] W. Dai, H. Xu, F. Shi, L. Yuan, X. Wang, and X. Wen, “ACDR-CRAFF Net: a multi-scale network based on adaptive channel and coordinate relational attention network for remote sensing scene classification,” *IET Image Process.*, vol. 19, no. 1, p. e70112, 2025.

[62] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, “Remote sensing image scene classification using bag of convolutional features,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1735–1739, 2017.

[63] M. Tan and Q. V. Le, “EfficientNet: rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[64] H. Sun, S. Li, X. Zheng, and X. Lu, “Remote sensing scene classification by gated bidirectional network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 82–96, 2020.

[65] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: more features from cheap operations,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1577–1586.

[66] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *nternational conference on machine learning*, 2021, pp. 10347–10357.

[67] S. Wang, Y. Ren, G. Parr, Y. Guan, and L. Shao, “Invariant deep compressible covariance pooling for aerial scene categorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6549–6561, 2021.

[68] F. Peng, W. Lu, W. Tan, K. Qi, X. Zhang, and Q. Zhu, “Multi-output network combining GNN and CNN for remote sensing scene classification,” *Remote Sens.*, vol. 14, no. 6, pp. 1–19, 2022.

[69] B. Zhang, Y. Zhang, and S. Wang, “A lightweight and discriminative model for remote sensing scene classification with multidilation pooling module,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 8, pp. 2636–2653, 2019.

[70] X. Bian, C. Chen, L. Tian, and Q. Du, “Fusing local and global features for high-resolution scene classification,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 6, pp. 2889–2901, 2017.

[71] Y. Yu and F. Liu, “A two-stream deep fusion framework for high-resolution aerial scene classification,” *Comput. Intell. Neurosci.*, p. 8639367, 2018.

[72] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, “Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification,” *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, 2018.

[73] B. Liu, C. Zhan, C. Guo, X. Liu, and S. Ruan, “Efficient remote sensing image classification using the novel STConvNeXt convolutional network,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–18, 2025.

[74] A. Chen and M. Xu, “Remote sensing image scene classification based on mutual learning with complementary multi-features,” *IEEE Access*, vol. 13, pp. 33436–33454, 2025.

[75] Y. Duan, C. Song, Y. Zhang, P. Cheng, and S. Mei, “STMSF: swin transformer with multi-scale fusion

for remote sensing scene classification," *Remote Sens.*, vol. 17, no. 4, p. 668, 2025.

[76] J. Ma, W. Jiang, X. Tang, X. Zhang, F. Liu, and L. Jiao, "Multiscale sparse cross-attention network for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–16, 2025.

[77] F. Zheng, S. Lin, W. Zhou, and H. Huang, "A lightweight dual-branch swin transformer for remote sensing scene classification," *Remote Sens.*, vol. 15, no. 11, pp. 1–19, 2023.

[78] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.

[79] X. Tang, M. Li, J. Ma, X. Zhang, F. Liu, and L. Jiao, "EMTCAL: efficient multiscale transformer and cross-level attention learning for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[80] X. Tu, L. Tianruo Yang, S. Liu, and R. Li, "Accelerated feature extraction and refinement for improved aerial scene categorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–17, 2024.

[81] R. Datla, N. Perveen, and C. Krishna Mohan, "Learning scene-vectors for remote sensing image scene classification," *Neurocomputing*, vol. 587, p. 127679, 2024.

[82] Y. Niu, Z. Song, Q. Luo, G. Chen, M. Ma, and F. Li, "ATMformer: an adaptive token merging vision transformer for remote sensing image scene classification," *Remote Sens.*, vol. 17, no. 4, p. 660, 2025.

[83] Y. Lu, Z. Hu, W. Zhao, Z. Guan, M. Gong, and M. Zhang, "Layer-interaction adaptive pruning for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–12, 2025.

[84] X. Sun, Q. Zhu, and Q. Qin, "A multi-level convolution pyramid semantic fusion framework for high-resolution remote sensing image scene classification and annotation," *IEEE Access*, vol. 9, pp. 18195–18208, 2021.

[85] C. Xu, G. Zhu, and J. Shu, "A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[86] Y. Lu, Y. Zhu, H. Feng, and Y. Liu, "Remote sensing scene classification using multi-domain sematic high-order network," *Image Vis. Comput.*, vol. 143, Mar. 2024.

[87] C. Xu, J. Shu, Z. Wang, and J. Wang, "A scene classification model based on global-local features and attention in lie group space," *Remote Sens.*, vol. 16, no. 13, pp. 1–22, 2024.

[88] J. Zhang, L. Wu, X. Shi, and B. Wang, "A lightweight self-supervised learning segmentation model for variable and complex high-resolution remote sensing images," *Appl. Soft Comput.*, vol. 165, p. 112061, 2024.

[89] Z. H. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, 2005.