ESIF: Frequency and Texture Aware Multi-Domain Feature Fusion for Enhanced Remote Sensing Scene Classification

***Abstract*—Remote Sensing (RS) scene classification, a pivotal task in Earth observation, entails the categorization of satellite or aerial imagery into distinct land-use and land-cover classes. It is fraught with challenges due to high intra-class variability and low inter-class distinctions. Historically, state-of-the-art methods in this field have struggled to achieve satisfactory results without a significant trade-off in computational efficiency. These methods often require substantial computational resources to process the complex data characteristics of RS imagery, leading to inefficiencies that limit their practical application in real-time or on resource-constrained platforms. Delving into these complexities, we propose the Efficient Spectral Inception Former (ESIF) architecture, which introduces a novel paradigm in RS scene classification by integrating multi-domain feature fusion, including spatial, texture, and spectral (frequency) domains. Our proposed approach leverages the strengths of Convolutional Neural Networks (CNNs) for spatial information extraction, a novel Texture Feature Alignment Block (TFAB) for nuanced texture differentiation, the Efficient SpectroFormer Block (ESFB) utilizing spectral analysis for enhanced pattern recognition, the Cross-Domain Fusion Block (CDFB) and finally, the Inception Transformer Block (iFB) balances high and low-frequency information. ESIF achieves state-of-the-art accuracy in all six tested benchmarks with 86.55% on Optimal-31, 95.71% on UC-Merced, 94.1% on RSSCN7, 95% on SIRI-WHU, 94.52% on WHU-RS19 and 93.5% on AID datasets while compared with previous state-of-the-art architectures. Furthermore, ESIF exemplifies an optimal accuracy-efficiency trade-off, accentuating its viability for deployment in real-world applications.**

***Index Terms*—Convolutional neural networks (CNNs), Frequency analysis, Large Kernel Attention (LKA), Multi domain, Remote sensing (RS), Scene classification, Self-Attention, Texture Analysis.**

I. Introduction

REMOTE sensing techniques, as employed in Earth observation, represent a pivotal area of research that involves the acquisition of signals emanating from various physical phenomena through instruments mounted on spaceborne and airborne platforms. These methodologies are invaluable for a broad spectrum of applications, ranging from the accurate measurement and estimation of geo-bio-physical parameters to the identification of materials based on the analysis of the signals [1] [2]. Scene classification from RS imagery are among the most significant tasks in this field, offering essential insights for various applications such as urban planning, agriculture, environmental conservation, climate change research etc. [2].

In recent advancements within the field of RS Scene Classification, deep learning-based vision algorithms have markedly increased in prevalence and effectiveness. Predominantly, these models are stratified into three primary categories: Convolutional Neural Network or CNN-based [3] [4] [5] [6] [7], Transformer-based [8] [9] [10] and Hybrid (CNN+Transformer) [11] [12] models. CNNs are renowned for their capacity to extract local information from the input, synthesizing numerous local inferences to produce the final output [13]. Conversely, Transformer models are engineered to capture global information from the onset, though they often lack the nuanced local contexts inherent to the input, a gap that Hybrid models aim to bridge by amalgamating the strengths of both CNNs and Transformers to offer a comprehensive representation of both local and global information [14] [15].

However, a significant challenge arises when applying these general-purpose vision models, primarily developed for conventional images captured at eye level, to the domain of RS images. RS imagery, typically acquired via satellites, aircraft, or drones, exhibits fundamental distinctions from standard photography, primarily due to the divergent viewpoint (Top-view versus Eye-Level) [16]. This disparity in perspectives necessitates a different set of features to accurately represent the same object or class across these two modalities. Consequently, models pretrained on eye-level images do not seamlessly transition to the RS domain, often leading to misclassification due to the discordance in feature representation between eye-level and top-view imagery [17] [18]. Pretrained models trained on eye-level images are not always quite suitable for RS images and can be easily prone to misclassification. This discrepancy underscores the necessity for developing or adapting models specifically tailored for RS images, taking into account their unique characteristics and challenges.

In real-world applications, RS datasets often exhibit a marked contrast in size when compared to datasets utilized in other image classification tasks. This discrepancy primarily arises from the logistical challenges and financial constraints associated with collecting RS imagery, rendering the assembly of large-scale datasets a formidable endeavor [16]. Unlike the extensive ImageNet database, there exists no comparable, large-scale dataset tailored for RS that could facilitate pretraining [19]. Typically, RS datasets feature 100 or fewer samples per class, significantly impeding a model's capacity to learn generalization. This challenge is exacerbated as the number of classes within a dataset increases. A case in point is the Optimal-31 [20] dataset, characterized by its 31 classes and a mere 60 samples per class, posing a unique set of challenges for classification models. High intra-class variation, the appearance of 'beach' or 'forest' can vary widely depending on location, season, and lighting conditions. High inter-class similarity, 'airport' and 'runway' or 'parking_lot' and "harbor" might share similar features, such as large, flat areas, making them harder to distinguish. Limited Training Samples, the diversity within some classes may not be fully captured by 60 samples, leading to a model that generalizes poorly on those classes. Complexity of Features, distinguishing between ”chaparral” and ”meadow” might require understanding subtle differences in vegetation patterns. These causes in some of the classes being significantly harder to classify than others. From Table IV we can see that even though the overall accuracy (OA) of a model reaches around 80%, some classes have less than 50%, even in some cases less than 30% individual accuracy (IA).

Deep-learning vision algorithms typically necessitate substantial volumes of training data to discern the complex features inherent to each class, thereby attaining a high overall accuracy (OA). This requirement, however, poses a significant challenge for RS scene classification tasks due to the scarcity of such extensive datasets in the field. While models endowed with substantial computational resources may achieve commendable OA, their performance often skews towards the more readily classifiable classes, leaving the more challenging categories relatively underserved [21]. The pursuit of high OA frequently necessitates trade-offs against efficiency and processing speed [6] [9], thereby constraining the practical applicability of these models in real-world scenarios. RS scene datasets which only has RGB or spatial image information lacks the additional cues which multi-band or hyperspectral RS images have. To construct an efficient algorithm which can accurately classify the challenging classes, efficient multi-domain analysis such as frequency and texture information could be beneficial.

Thus, we introduce the *ESIF: Efficient Spectral Inception Former* architecture, a pioneering approach that processes input data across three domains—Spatial, Texture, and Frequency (Spectral)—in parallel branches. This methodology enables simultaneous analysis of the same input, leveraging the strengths of each domain to enhance RS image understanding and classification. Overall the main contributions are as follows:

- We propose the Texture Feature Alignment Block (TFAB), which utilizes three gray level co-occurrence matrix (GLCM) [22] features, crucial for capturing the nuances of image textures, enabling the model to distinguish between subtle variations in visual patterns effectively reducing the inter-class similarity through texture information.
- Efficient SpectroFormer Block (ESFB) is constructed with Spectral and Large Kernel Attention (LKA) blocks to capture frequency information through Fast Fourier Transform (FFT) and refine it with attention mechanism, which alleviates the high-intra class variation problem.
- Cross-Domain Fusion Block (CDFB) mechanism is deployed to effectively synthesize the outputs from the spatial, texture, and spectral branches, followed by the incorporation of the iFormer Block in the later stages to balance the high and low frequency components.

The rest of the article is organized as follows: Section II details our methodology and overall building of the network architecture, Section III discusses the experimental results and Section IV is the conclusion of our work.

II. Methodology

A. Design Concept of ESIF

In the development of the Efficient Spectral Inception Former (ESIF) architecture as showed in Fig. 1, we strategically orchestrate the processing of input data across three distinct domains: Spatial, Texture, and Frequency (Spectral), leveraging parallel branches to concurrently analyze the same input. Specifically, the input RGB image, denoted as $S \in \mathbb{R}^{H\times W\times 3}$, is simultaneously directed towards the Spatial Baseline and the Efficient SpectroFormer Block (ESFB) branches. Meanwhile, the Texture Feature Alignment Block (TFAB) branch processes the GLCM outputs: Contrast, Correlation, and Angular Second Moment (ASM) features extracted from S. Cross-Domain Fusion Block (CDFB) integrates the outputs of these three branches post the fourth stage within the Spatial Branch, ensuring a comprehensive synthesis of spatial, texture, and frequency information. Subsequent to the sixth stage of the Spatial Branch, we introduce the Inception Transformer (iFormer) Block, which further refines the spatial features. This is followed by a 1×1 convolutional layer aimed at expanding the feature map, an adaptive average pooling layer for feature concentration, and a Classifier that delineates the final output. This architecture design, embodying the simultaneous and synergistic processing across multiple domains, exemplifies our approach to harnessing the full spectrum of visual information for image understanding and classification. The detailed process of each block is explained in the next sections.

B. MBConv Based Spatial Baseline

To construct our efficient baseline, we adopt the MBConv6 block from EfficientNet [3], which is a improved version of the mobile inverted bottleneck convolution of MobileNetV2 [4]. This architecture leverages depthwise separable convolutions along with a squeeze-and-excitation (SE) mechanism for feature extraction efficiency and focusing on important channels. While being slightly lower in speed than the MobileNetV2, it is much more accurate and consistent. But, our baseline architecture varies in a lot of ways with both MobileNetV2 and EfficientNet-B0. The detailed architecture is illustrated in . In our Spatial Baseline, illustrated in Fig. 1, the first stem block consists of 3×3 convolution with stride 2 and downsamples the input image by half, while projecting from 3 to 32 channels. We use 6 stages of MBConv6. In a typical MBConv6 block, the first step involves expanding the input feature map using a 1×1 convolution. This expansion increases the number of channels, aiming to provide a richer representation for the depthwise convolution to process.

$$Y_{\text{exp}} = \text{ReLU6}\left(\text{BN}\left(\text{Conv}_{1\times 1,\, c_{in}\rightarrow c_{exp}}(X)\right)\right). \tag{1}$$

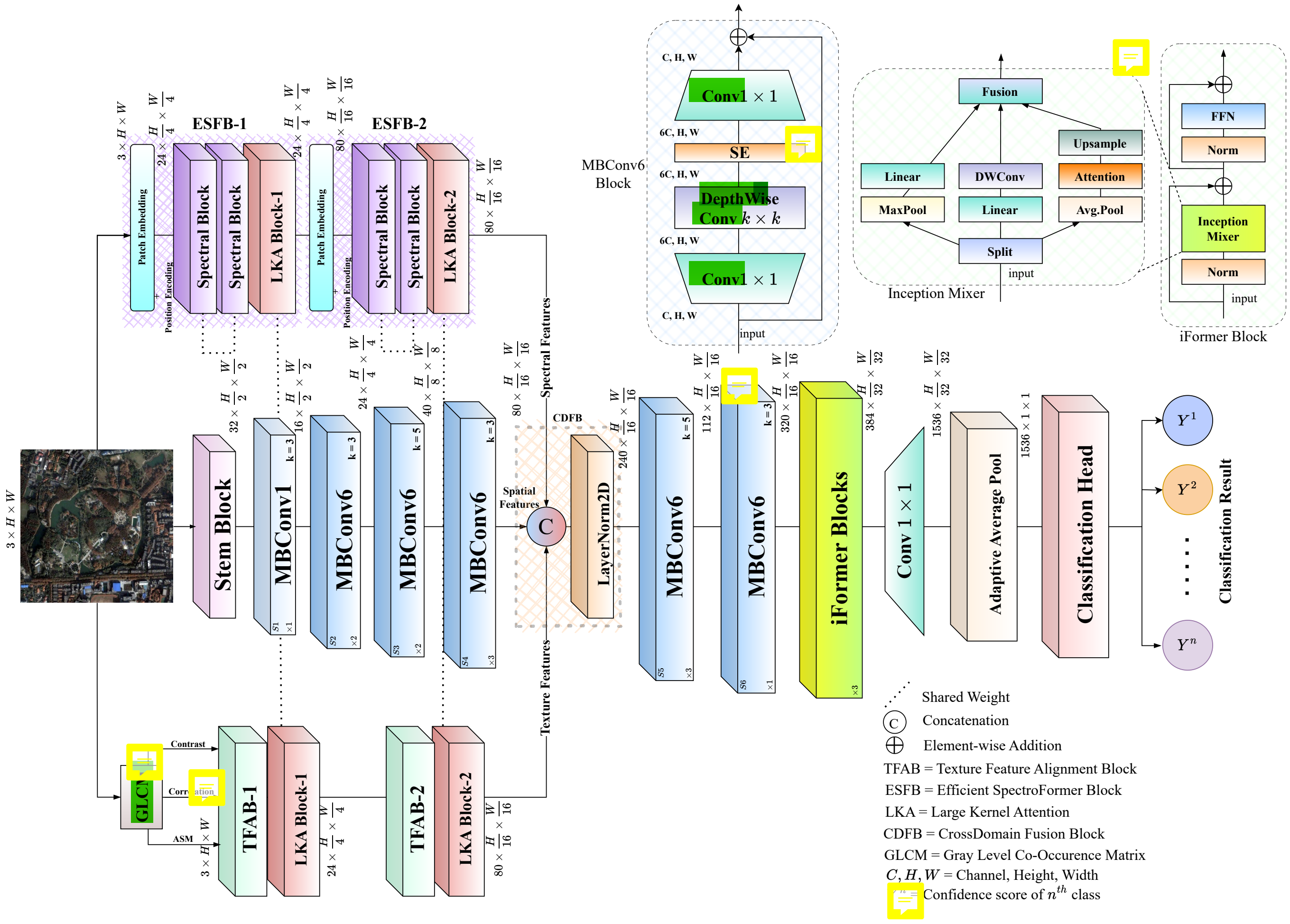

Fig. 1. Overall Architecture of the proposed network ESIF: Efficient Spectral Inception Former, comprised of three branches: Spatial Baseline for spatial feature extraction, TFAB for processing Texture information, ESFB for extraction of Spectral information, and CDFB for mulit-domain fusion, followed by iFormer Blocks for balancing

Following expansion, a $k \times k$ depthwise convolution applies spatial filtering to each channel independently, allowing for efficient extraction of spatial features.

$$Y_{\text{dw}} = \text{ReLU6}\left(\text{BN}\left(\text{DWConv}_{k\times k,\, c_{exp}}(Y_{\text{exp}})\right)\right). \quad (2)$$

The SE mechanism recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels, enhancing the representational capacity for important features.

$$Y_{\text{se}} = \text{SE}(Y_{\text{dw}}). \quad (3)$$

The expanded feature map is then projected back down to a lower-dimensional space using another 1×1 convolution, compacting the information learned from the depthwise convolution and SE block.

$$Y_{\text{proj}} = \text{BN}\left(\text{Conv}_{1\times 1,\, c_{exp}\to c_{out}}(Y_{\text{se}})\right). \quad (4)$$

If the input and output dimensions are the same (typically when stride is 1 and $c_{in} = c_{out}$), a residual connection is added from the block's input to its output, facilitating gradient flow and preserving identity features.

$$F_s = Y_{\text{proj}} + X, \quad (5)$$

where s is the stage number. MBConv6 block is designed for efficient and effective feature extraction, balancing computational efficiency with the capacity to capture essential spatial and channel-wise information. We set the output dimension of the 6 stages as $c_i = [16, 24, 40, 80, 112, 320]$. For effective feature fusion through CDFB, c for stage-2 is aligned with TFAB-1, ESFB-1 and c stage-4 is aligned with TFAB-2, ESFB-2.

C. Texture Feature Alignment Block (TFAB)

For generating texture features from the input RGB image, we incorporate characteristics derived from the well-known Gray-Level Co-occurrence Matrix (GLCM) to augment our texture analysis. Specifically, we utilize both the contrast ratio and relevance metrics as supplementary texture descriptors, leveraging inputs from GLCM. In this configuration, we employ sub-windows of size 3×3 and set the number of gray levels to eight. We consider 3 GLCM features for our TFAB block. *Contrast* (CON) feature measures the intensity contrast between a pixel and its neighbor over the whole image. High contrast values indicate a large difference in intensity between pixel pairs, suggesting a more textured

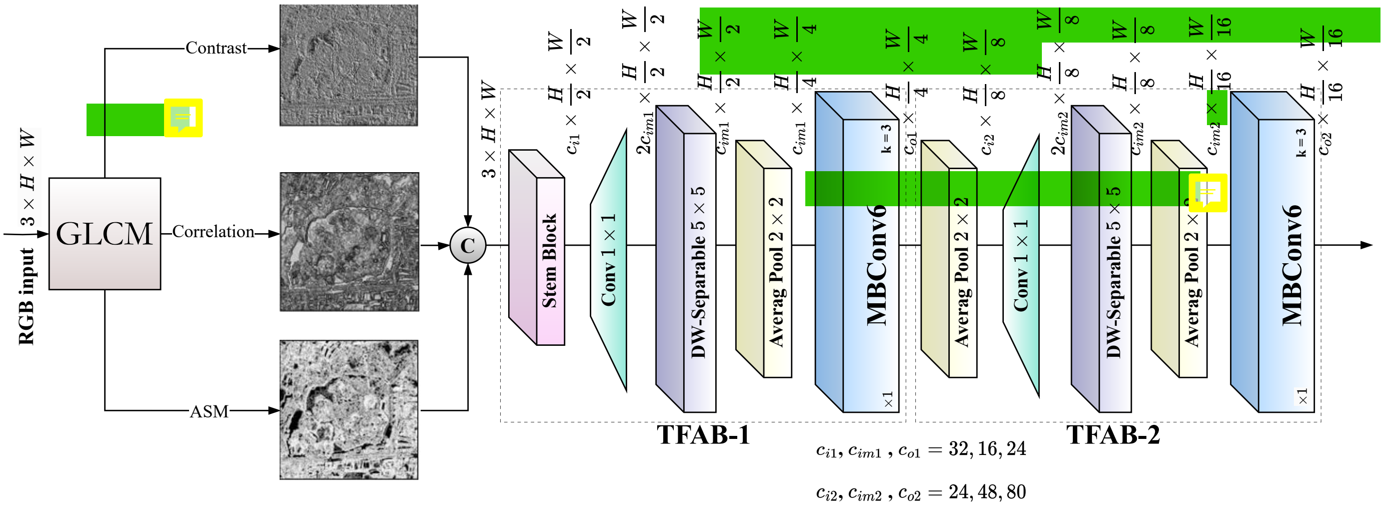

Fig. 2. Detailed Architecture of TFAB: Texture Feature Alignment Block.

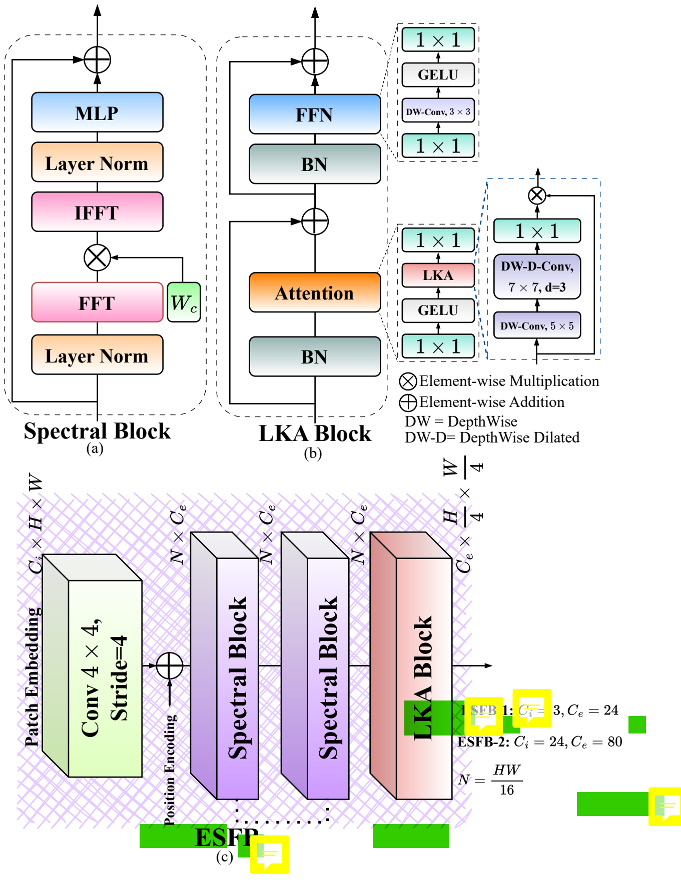

Fig. 3. Detailed Architecture of ESFB: Efficient Spectro-Former Block.

and less smooth image. Low contrast values suggest minimal intensity difference between neighboring pixels, indicating a smoother image texture. Relevance or *Correlation* (CORR) is the similarity degree of GLCM elements in directions of line and row, which denotes the relevant degree of some gray levels in images. *Angular Second Momentum* (ASM) emerges as a valuable metric for discerning the depth of textures and patterns. A higher ASM value signifies the presence of more pronounced textures and deeper patterns, while a lower value corresponds to a blurred visual representation with shallower textures.

$$CON = GRAY(\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2), \tag{6}$$

$$CORR = GRAY(\sum_{i,j=0}^{N-1} P_{i,j}\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2\sigma_j^2}}), \tag{7}$$

$$ASM = GRAY(\sum_{i,j=0}^{N-1} P_{i,j}^2), \tag{8}$$

where N is the size of GLCM and $P(i,j)$ is the probability density of the corresponding pixel, μ_i, μ_j and σ_i,σ_j refer to mean and variance of $P_x(i)$ and $P_y(j)$ respectively. GRAY indicates converting to gray-level image of one channel. Finally, three texture feature maps: $x_{CON} \in \mathbb{R}^{H\times W\times 1}$, $x_{CORR} \in \mathbb{R}^{H\times W\times 1}$, and $x_{ASM} \in \mathbb{R}^{H\times W\times 1}$ are obtained with above equations.

TFAB have two stages, TFAB-1 and TFAB-2. The detailed architecture is shown in Fig. 2, the three GLCM features are concatenated to the size $H\times W\times 3$. The stem block is a standard 3×3 convolution with stride 2 for downsampling the input to $\frac{H}{2}\times\frac{W}{2}\times 3$. We employ $c_{i(s)}$, $c_{im(s)}$ and $c_{o(s)}$ three dimensions indicating input channels, intermediate channels and output channels respectively, where s is the stage number. For a stage of TFAB it is processed by four consecutive operations, first, a 1×1 Convolution to expand the intermediate channels. Which acts as a pointwise linear transformation, mixing the input channels to produce a richer set of features and allows the network to represent a broader range of features and textures within the image.

$$Y_t^{(1)} = \mathrm{Conv}_{1\times1,\, c_{i(s)}\to 2c_{im(s)}}(X_t). \tag{9}$$

Secondly, a 5×5 depthwise separable convolution for feature extraction. By separating the convolution into a depthwise spatial component and a pointwise channel mixing component, it allows for detailed texture analysis with reduced computational cost. The depthwise component focuses on extracting spatial texture patterns from each channel independently, emphasizing the nuances of texture within the image. The subsequent pointwise convolution then combines these extracted features across channels, enhancing the model's ability to detect and represent diverse texture information.

$$Y_t = \mathrm{DWSConv}_{5\times5,\, 2c_{im(s)}\to c_{im(s)}}(Y_t^{(1)}). \tag{10}$$

After that, a 2×2 average pool is used for concentrating important features. By averaging the values within 2×2 patches, this step effectively distills the most significant texture information into a more compact representation. This process aids in reducing noise and focusing the model's attention on the most relevant texture features for classification tasks. And one MBConv6 Block (kernel size = 3) for aligning the texture features with the corresponding spatial domain. The MBConv6 block further processes the concentrated texture features, aligning them with the spatial domain of the image. This operation is crucial for integrating the extracted texture information with the overall spatial structure of the image, ensuring that texture features are correctly associated with their spatial context. The whole process can be defined by:

$$\mathrm{TFAB}(X_t) = \mathrm{MBConv6}_{3\times3,\, c_{im(s)}\to c_{o(s)}}\left(\mathrm{AvgPool}_{2\times2}(Y_t)\right). \tag{11}$$

The difference between TFAB-1 and TFAB-2 is the initial downsampling operation, while TFAB-1 uses the stem block, TFAB-2 utilizes a simple 2×2 average pool. $c_{i(s)}$, $c_{im(s)}$ and $c_{o(s)}$ values for TFAB-1 = $(32, 16, 24)$ and TFAB-2 = $(24, 48, 80)$. The $c_{o(s)}$ is matched with stage 2 and stage 4 of the Spatial Baseline for aligning the respective spatial features with texture features.

D. Efficient SpectroFormer Block (ESFB)

The detailed design of ESFB is presented in Fig. 3. For processing image information using transformer based models, Patch Embedding mechanism is used to patchify the input image into smaller patches. The convolutional approach to creating patch embeddings inherently extracts useful low-level features from images, such as edges and textures, providing a richer input to the Transformer block. We use a 4×4 convolution with stride 4 on the input image I to produce a set of patches $P(I)$. This effectively reducing the spatial dimensions while increasing the depth from c_i to c_e. Position encodings are added to these patches to retain spatial context lost during dimensionality reduction, essential for maintaining the positional relationship between patches in subsequent processing.

$$P(I) = \mathrm{Conv}_{4\times4,\mathrm{stride}=4,C_i\to C_e}(I) + \mathrm{PositionEncoding}. \tag{12}$$

The Spectral Block, shown in Fig. 3(a), leverages the Fourier Transform (FFT) to transform the spatial features patch embeddings to sequence features in the fourier domain, learning the spectral components through element-wise multiplication with complex weights W. The inverse Fourier Transform (IFFT) then maps these modified components back to the spatial domain. This process, learned with layer normalization (LN) and a Multi-Layer Perceptron (MLP), extracts and refines frequency-based features, facilitating detailed texture and pattern analysis.

$$S(P) = \mathrm{MLP}\left(\mathrm{LN}\left(\mathrm{IFFT}\left(\mathrm{LN}\left(\mathrm{FFT}(P) \otimes W\right)\right)\right)\right) + P. \tag{13}$$

The LKA mechanism which is shown in Fig. 3(b), focuses on capturing spatial details by applying depthwise convolutions followed by a 1×1 convolution to the input x, emphasizing local features through element-wise multiplication. This operation amplifies the model's sensitivity to spatial variations and details, crucial for understanding visual textures and structures.

$$\mathrm{LKA}(x) = (\mathrm{Conv}_{1\times1}(\text{DW-D-Conv}(\text{DW-Conv}(x)))) \otimes x. \tag{14}$$

The FFN applies a series of convolutions, including a 3×3 depthwise convolution activated by GELU, to process the spatial features further. This network refines the feature maps, ensuring that the model captures both broad and nuanced spatial information effectively.

$$\mathrm{FFN}(x) = \mathrm{Conv}_{1\times1}(\mathrm{GELU}(\text{DW-Conv}_{3\times3}(\mathrm{Conv}_{1\times1}(x)))). \tag{15}$$

The following equation integrates the LKA's output with the original input X through a residual connection, fostering the preservation of initial features while incorporating the detailed spatial analysis performed by the LKA.

$$Y = \mathrm{LKA}(\mathrm{BN}(X)) + X. \tag{16}$$

Building upon the refined features from the LKA, this step processes Y through the FFN, enhancing the feature set with further spatial refinement and ensuring a deep processing capability through an additional residual connection.

$$\mathrm{LKABlock(X)} = \mathrm{FFN}(\mathrm{BN}(Y)) + Y. \tag{17}$$

The culmination of the ESFB process [see Fig. 3(c)] involves applying two sequential Spectral Block operations on the patch P, followed by the comprehensive spatial refinement offered by the LKABlock. This ensures a rich, multi-dimensional feature representation, crucial for advanced analysis and classification tasks.

$$\mathrm{ESFB}(P) = \mathrm{LKABlock}\left(S_2\left(S_1(P)\right)\right). \tag{18}$$

Through these operations, the ESFB effectively integrates spectral, spatial, and attention-based mechanisms to process and refine input features. To align the Spectral Domain features with Spatial Branch and Texture Branch, we set the c_e of ESFB-1 to 24 and ESFB-2 to 80.

E. Cross-Domain Fusion Block (CDFB)

The CDFB adeptly merges spatial, texture, and spectral features into a unified representation, shown in Fig. 1. It leverages the strengths of each domain for enhanced scene classification. It is simply given by:

$$\mathrm{CDFB} = \mathrm{Norm}\left(\mathrm{Concat}\left(F_4, \text{TFAB-2}, \text{ESFB-2}\right)\right), \tag{19}$$

here F_4 is the output of Stage-4 MBConv6 Block. It is crucial to place CDFB after the 4th stage, as early-stage fusion does not benefit from unrefined cross-domain information. Furthermore, late-stage fusion might integrate features that are too abstract. By placing CDFB after 4th stage and aligning the dimensions of all three branches effective feature interaction is ensured.

TABLE I
DETAILED PERFORMANCE COMPARISON OF PREVIOUS STATE-OF-THE-ART CLASSIFICATION MODELS ON THE OPTIMAL-31 [20] DATASET

Model Name	Model Composition	Params. (M)	FLOPs (G)	BA	AA	Model Size(MB)	Memory Access(GB)	Training Time(h)	Inf. Speed(FPS)	AETS
MobileNetV2 (2018) [4]	CNN	2.2	0.3	79.83	77.68	**18.4**	1.47	**0.46**	**240**	74.78
EfficientNet-B0 (2019) [3]	CNN	4.0	0.4	80.10	78.75	32.7	1.57	0.65	226	76.94
Van (2022) [5]	CNN	4.1	0.9	73.11	72.75	31.1	1.43	0.87	208	68.48
EfficientNetB3 (2019) [3]	CNN	10.7	1.0	79.56	78.84	86.6	1.77	1.56	186	59.00
MobileViT-S (2022) [12]	Hybrid	5.0	1.8	68.81	68.63	40.4	1.71	1.06	192	58.52
FasterNet-T2 (2023) [7]	CNN	13.7	1.9	76.07	75.89	110.1	**1.46**	**0.46**	200	72.87
SpectFormer-H-S (2023) [10]	Transformer	20.2	3.9	80.37	78.84	171.0	1.73	1.44	163	57.33
iFormer-S (2022) [11]	Hybrid	18.9	4.5	76.88	76.88	156.2	1.74	2.16	145	55.25
MaxViT-T (2022) [9]	Transformer	30.3	5.1	77.15	77.15	244.4	2.05	1.60	129	54.48
SwinV2-S (2021) [8]	Transformer	33.2	5.8	78.22	76.07	393.1	2.08	3.21	110	26.31
DenseNet161 (2017) [6]	CNN	26.5	7.8	80.61	80.64	213.7	1.75	2.45	127	46.78
ESIF (Ours)	Hybrid	9.0	1.1	**86.55**	**85.48**	75.2	1.61	1.23	153	**85.78**

F. Inception Transformer Block (iFB)

We adpot the iFormer Block from [11], which is utilized to refine and synthesize the feature representations extracted and fused from previous stages. In the original architecture, Inception Transformer contains 4 stages, we only adopt the 4th stage to balance the local and global information before classifier. Through its inception-inspired design, the iFormer 4th Stage emphasizes the balance between high-frequency (detail-oriented) and low-frequency (global context) information, ensuring that both aspects are adequately represented and utilized in the final feature map. Positioned before the final 1×1 convolution, which expands the feature map four times, the iFormer block ensures that the expanded features are of the highest quality, containing all necessary information for the subsequent classification. In the Inception mixer, rather than directly inputting image tokens into the Multi-Head Self-Attention (MSA) mixer, the approach involves an initial division of the input feature across the channel dimension. Subsequently, these divided components are separately processed by a high-frequency mixer and a low-frequency mixer. The high-frequency mixer employs both a max-pooling operation and a convolution operation in parallel to manage the high-frequency aspects, whereas the processing of low-frequency elements is handled through a self-attention mechanism.

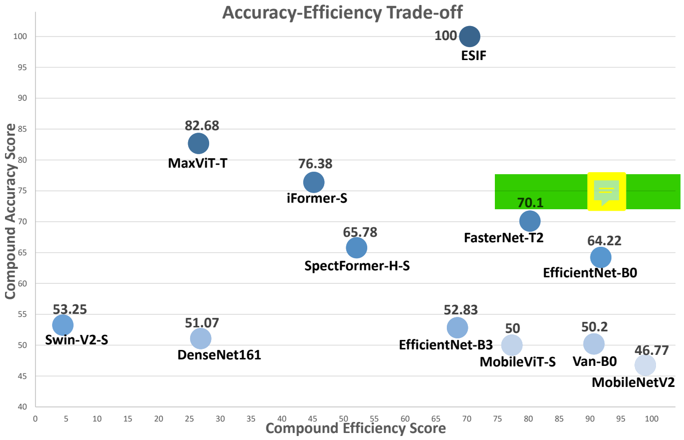

Fig. 4. Comparison of Compound Accuracy Score (CAS) VS Compound Efficiency Score (CES) of various models.

III. EXPERIMENTS

A. Datasets

In the evaluation of the proposed model, six RS scene classification datasets were utilized and each dataset presens unique challenges in terms of class diversity, image resolution, and sample size. The datasets are detailed as follows:

1) Optimal-31 [20]: The Optimal-31 dataset comprises 1,860 images distributed across 31 classes, with each class containing 60 images. The dataset poses significant challenges due to the low number of samples per class, a high number of classes, and minimal inter-class variation. Each image within the dataset has a resolution of 256×256 pixels, further complicating the classification task due to the limited spatial information available.

2) UC Merced [24]: The UC Merced dataset includes 2,100 images, distributed equally among 21 classes, each containing 100 images. With a spatial resolution of 0.3 meters per pixel, the images (256×256 pixels) are obtained from the US Geological Survey, providing a comprehensive view of various US landscapes. This dataset tests the model's performance in classifying diverse natural and man-made features.

3) RSSCN7 [25]: Derived from Google Earth for research purposes, the RSSCN7 dataset includes 2,800 images with seven classes, allocating 400 images for each class. The images are 400×400 pixels in size. The dataset is notable for its scale variation, which presents a considerable challenge in achieving consistent classification accuracy across all classes.

4) Siri-Whu [26]: This dataset consists of 2,400 images across 12 classes, with 200 images per class. The images, featuring a spatial resolution of 2 meters and dimensions of 200×200 pixels, predominantly cover urban areas within China. The urban focus and uniform class distribution facilitate focused analysis on man-made structures and their classification from satellite imagery.

5) WHU-RS19 [27]: Comprised of high-resolution RGB satellite images from Google Earth, the WHU-RS19 dataset contains 19 classes with approximately 50 samples per class, culminating in a total of 1,005 images. The dataset is characterized by its class imbalance and a uniform image resolution of 600×600 pixels, challenging the model's ability to generalize across less-represented classes.

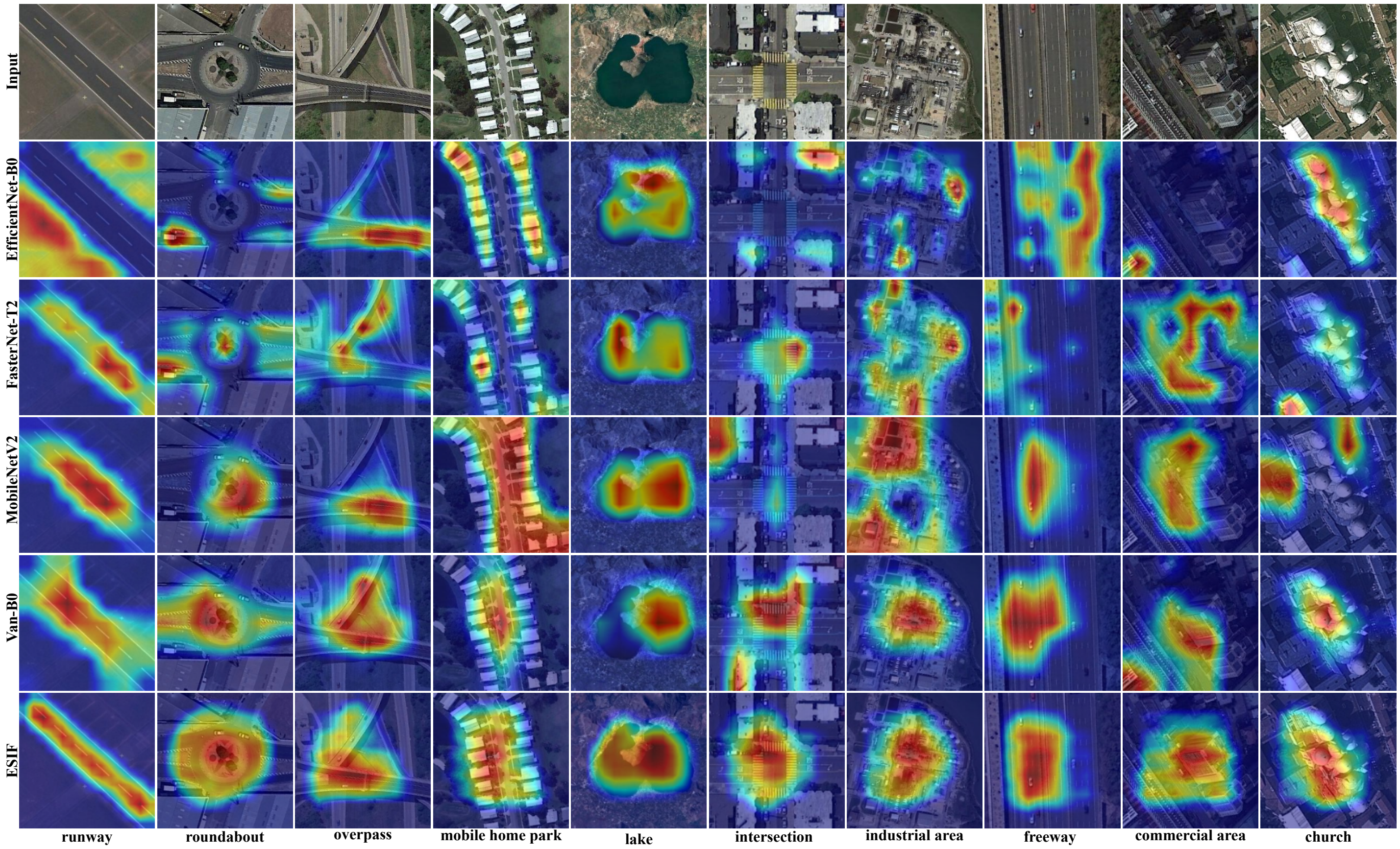

Fig. 5. Eigen-Cam [23] activation maps of EfficientNet-B0 [3], FasterNet-T2 [7], MobileNetV2 [4], Van-B0 [5] and ESIF (Ours). Only samples from the difficult classes are shown from Optimal-31 [20] dataset.

TABLE II
Evaluation on the UC-Merced, RSSCN7, SIRI-WHU, WHU-RS19 and AID Datasets

Model Name	Params. (M)	FLOPs (G)	UC-Merced [24]		RSSCN7 [25]		SIRI-WHU [26]		WHU-RS19 [27]		AID [28]
			BA	AA	BA	AA	BA	AA	BA	AA	BA
MobileNetV2 (2018) [4]	2.2	0.3	92.85	92.13	90.17	88.32	91.45	90.27	92.53	87.55	90.85
EfficientNet-B0 (2019) [3]	4.0	0.4	95.0	94.04	92.67	90.55	93.33	92.91	86.06	84.57	90.45
Van (2022) [5]	4.1	0.9	91.90	91.34	89.64	89.05	93.12	92.84	88.55	86.89	88.70
EfficientNet-B3 (2019) [3]	10.7	1.0	92.38	88.72	93.57	91.72	93.54	92.29	77.11	76.11	90.95
MobileViT-S (2022) [12]	5.0	1.8	90.47	90.39	90.71	90.65	92.29	92.29	87.56	87.56	87.65
FasterNet-T2 (2023) [7]	13.7	1.9	93.57	92.77	91.25	91.13	93.54	93.19	92.03	92.03	90.00
SpectFormer-H-S (2023) [10]	20.2	3.9	92.61	92.29	90.71	90.23	93.54	93.05	90.04	89.71	89.80
iFormer-S (2022) [11]	18.9	4.5	92.61	92.61	92.14	92.14	93.95	93.95	90.54	90.20	88.50
MaxViT-T (2022) [9]	30.3	5.4	93.33	92.77	93.21	92.97	94.37	94.16	91.04	91.04	93.05
SwinV2-S (2021) [6]	33.2	5.8	82.61	82.16	91.60	91.48	92.91	92.91	88.05	88.05	90.10
DenseNet161 (2017) [6]	26.5	7.8	95.47	94.75	86.70	85.86	92.50	92.08	93.53	92.70	93.15
ESIF (Ours)	9.0	1.1	**95.71**	**95.15**	**94.1**	**93.62**	**95.0**	**94.58**	**94.52**	**93.36**	**93.5**

6) AID [28]: As a large-scale dataset, the AID collection features 10,000 RGB images from Google Earth, each with a resolution of 600 × 600 pixels. It encompasses 30 diverse classes, with images sourced globally, exhibiting spatial resolutions ranging between 8 to 0.5 meters. This diversity and the variance in spatial resolution reinforce the dataset's utility in evaluating the robustness of classification models across a broad spectrum of aerial imagery.

For the purpose of model evaluation, each dataset was partitioned into training, validation, and test sets, comprising 60%, 20%, and 20% of the data, respectively. This split ensures a balanced approach to training and evaluating the classification model, allowing for a comprehensive assessment of its performance across different RS scenarios.

B. Implementation Details

We use the AiTLAS toolbox to train and evaluate our models. We train and evaluate each model compared in this paper from scratch using the same test split across all experiments for fair comparison. The preprocessing of input images involved resizing them to dimensions of 224 × 224 pixels. Regarding data augmentation, the study adopted the AutoAugment [29],

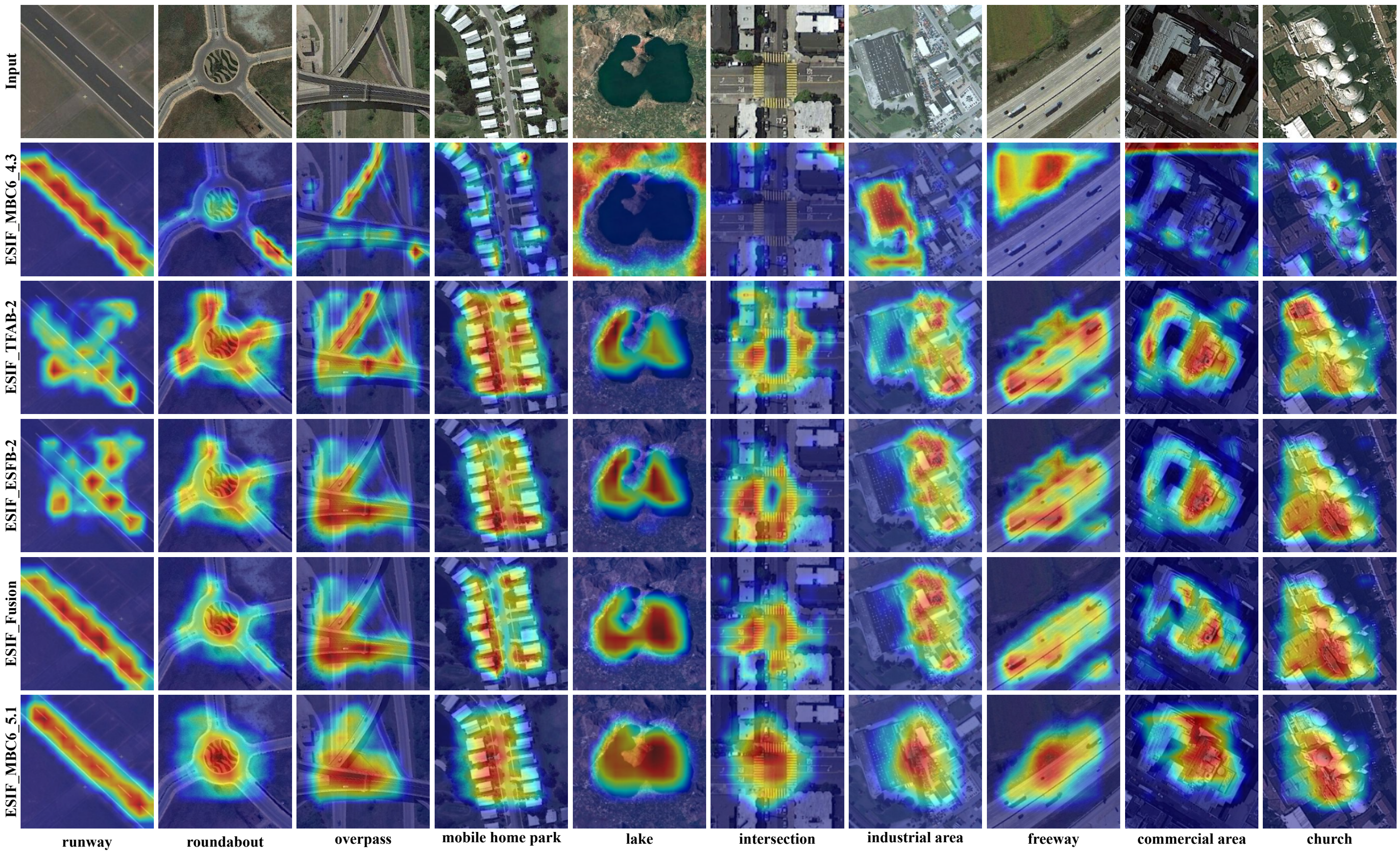

Fig. 6. Eigen-Cam [23] activation maps of ESIF_4th_Stage (Last Activation Map), ESIF_TFAB-2, ESIF-ESFB-2, ESIF-CDFB, ESIF_5th_Stage(First Activation Map, showcases the separate activation maps of each blocks and improved class activation maps by multi-modal fusion from 4th(2nd row) to 5th (last row).

TABLE III
ABLATION STUDY OF THE EFFECTIVENESS OF EACH PROPOSED BLOCKS TFAB, IFB, ESFB ON THE OPTIMAL-31 [20] DATASET.

TFAB	iFB	ESFB	Params (M)	FLOPs (G)	Mem. Acc.(GB)	BA	AA
✗	✗	✗	1.64	0.41	1.5	81.45	80.82
✓	✗	✗	2.00	0.64	1.6	82.79(+1.34)	82.32(+1.5)
✓	✓	✗	8.48	0.89	1.5	83.33(+0.54)	82.79(+0.47)
✓RGB	✓	✗	8.66	1.00	1.5	80.64(-2.69)	78.63(-4.16)
✓+LKA	✓	✗	8.67	0.94	1.5	83.33(+2.69)	82.79(+4.16)
✓	✓	MHSA	9.04	1.04	6.2	85.21(+1.88)	84.94(+2.15)
✓+LKA	✓	LKA	9.05	1.10	**1.6(-4.6)**	**86.55(+1.34)**	**85.48(+0.54)**

TABLE IV
CLASS-WISE ACCURACY COMPARISON FOR CHALLENGING CLASSES IN OPTIMAL-31 [20] DATASET.

Class Names	Efficient Net-B0 [3]	Mobile NetV2 [4]	Van -B0 [5]	Faster Net-T2 [7]	ESIF (Ours)
rectangular_farmland	47.61	61.53	27.27	50.00	**70.00**
commercial_area	55.55	63.15	**66.66**	60.00	50.00
church	63.63	70.58	**84.21**	50.00	70.58
freeway	66.66	69.56	51.85	52.17	**81.48**
overpass	66.66	63.63	50.00	63.15	**85.71**
mobile_home_park	69.99	66.66	46.15	63.15	**73.68**
lake	83.33	80.00	54.54	66.66	**83.33**
industrial_area	80.00	75.00	55.55	69.56	**86.95**
runway	82.75	63.63	60.86	58.33	**92.3**
roundabout	85.71	74.07	64.00	71.99	**92.85**

alongside the implementation of RandomHorizontalFlip and RandomVerticalFlip. These augmentations were applied to both RGB images and GLCM texture data. Its worth noting that exactly same augmentations should be applied to multi-modal data for effective communication between different modalities. For the training configurations, a batch size of 16 and 4 workers were utilized across all datasets with the exception of the AID dataset. Given the larger scale of AID, a batch size of 64 and 16 workers were deemed appropriate to accommodate its size. The study further incorporated the recently introduced LION (EvoLved Sign Momentum) [30] optimizer, which has demonstrated an improvement in training speed and convergence efficiency compared to other optimizers in similar experimental setups. We use the standard Cross-Entropy Loss as the loss function. The learning rate was set to $9e^{-5}$ for models based on CNNs and adjusted to $9e^{-6}$ for those based on Transformer and Hybrid architectures. This distinction arises from the observation that Transformer-based models require a lower learning rate for effective convergence, particularly when training from scratch. Notably, the FasterNet-T2 [7] model, despite being CNN-based, was trained with a learning rate of $9e^{-6}$ due to its operational similarities with Transformer architectures in practical applications. The learning rate was dynamically reduced by a factor of 0.1 in response to plateaus in validation loss improvements. We train each model for 300 epochs on all datasets except AID and 100 epochs on AID. All models were trained on an NVIDIA Tesla V-100 GPU equipped with 32 GB of memory.

C. Evaluation Metrics

The main evaluation metric in our experiments is the Accuracy or generally known as Top-1 Accuracy in classification tasks. It can be simple defined as,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{20}$$

We use the term Best Accuracy (BA) to denote the best possible result from that particular model and Average Accuracy (AA) an average of three separate instance of results for fair comparison. For calculating Efficiency of a model, we utilize six metrics, Parameters (**M**illions) referring to the total trainable parameters, FLOPs (Billions/**G**iga) or Floating Point Operations, which represents the computation complexity of a model, Memory Access is the GPU memory taken during Inference measured in GB, Training time in hours, Inference Speed is the model's ability to process number of frames per second during inference, the higher Inf.Speed is the better model, and for the other 5 metrics lower is better. The Accuracy-Efficiency Trade-off Score (AETS) for a model M, denoted as $AETS_M$, is calculated as the arithmetic mean of the Compound Accuracy Score (CAS) and the Compound Efficiency Score (CES).

$$CES_M = \frac{1}{6}\sum_{i=1}^{6}\left(\frac{S_{M,i} - \min(S_i)}{\max(S_i) - \min(S_i)} \times 100\right), \tag{21}$$

$$CAS_M = \frac{1}{6}\sum_{j=1}^{6}\left(\frac{A_{M,j} - \min(A_j)}{\max(A_j) - \min(A_j)} \times 100\right), \tag{22}$$

$$AETS_M = \frac{CES_m + CAS_M}{2}, \tag{23}$$

where, for a model M, $S_{M,i}$ is the best score for a given efficiency metric i and $A_{M,j}$ is the highest AA for a given dataset j. In this formulation, CAS_M represents the model's overall accuracy across multiple datasets, normalized and averaged to reflect comparative performance. Similarly, CES_M assesses the model's computational efficiency by averaging normalized scores across various efficiency metrics. The AETS thus encapsulates a balanced evaluation of model M's performance, combining accuracy and efficiency into a single metric to provide a comprehensive assessment of its effectiveness and resource utilization.

D. Evaluation on Optimal-31

The detailed experimental analysis on the Optimal-31 dataset is shown in Table I. We compare our proposed network ESIF with 11 previous state-of-the-art networks, with varying computational efficiency. Among them, six of them are purely CNN-based: EfficentNet-B0, EfficientNet-B3, MobileNetV2, Van-B0, DenseNet161 and FasterNet-T2. Three are pure transformer-based: SpectFormer-H-S, MaxViT-T and SwinV2-S. Two networks are CNN-Transformer Hybrids: MobileViT-S and InceptionTransformer (iFormer-S). We classifiy our network as a Hybrid since it employs Conv. based attention (LKA) as well as little amount of attention in the later stages using iFormerBlock. As previously explained, Average Accuracy (AA) is the average of three instance of training result, and Best Accuracy (BA) is the best possible accuracy attained by that model. Our proposed ESIF outperformed all the compared methods by a large margin in both the AA and BA category on Optimal-31 dataset. We achieve 86.55% BA, while the second best DenseNet161 achieves 80.91%, a 5.64% difference, similarly ESIF achieves 85.48% AA which is 4.84% higher than the second position of DenseNet161 at 80.64%. SpectFormer-H-S and EfficientNet-B0 models also perform more than 80% at BA but falls short at AA. In case of the efficiency metrics, our model is not the best in the list, MobileNetV2 model which is designed focusing on the efficiency comes out on top in most of the metrics-Parameters 2.2million, FLOPs 0.3 G , Model Size 18.4 MB, Inference Speed 240 FPS, while FasterNet-T2 is best for Memory Access 1.46 GB and Training Time 0.46h But, for the Accuracy-Efficiency Trade-Off Score AETS, our model achieves the best score of 85.78, while the trailing positions over 70 are EfficientNet-B0 with 76.94, MobileNetV2 with 74.78 and FasterNet with 72.87. This results highlights that ESIF maintains a high Accuracy-Efficiency Trade-Off, being more focused on accuracy and achieving state-of-the-art result while keeping up in the efficiency metrics as well. Fig. 5 shows the Eigen-Cam [23] activation maps of all eleven models for 10 challenging classes of Optimal-31 dataset.

E. Evaluation on UC-Merced, RSSCN7, SIRI-WHU, WHU-RS19 and AID

In the comprehensive evaluation presented in Table II, our Efficient Spectral Inception Former (ESIF) model consistently outperforms a broad spectrum of state-of-the-art models across several RS image datasets. On the UC-Merced dataset, known for its challenging urban and natural landscapes, ESIF achieves the highest Best Accuracy (BA) of 95.71% and Average Accuracy (AA) of 95.15%, surpassing DenseNet161 and EfficientNet-B0, which are the second and third best performers, respectively. The RSSCN7 dataset, characterized by a variety of scene categories, sees ESIF leading with a BA of 94.1% and an AA of 93.62%, with MaxViT-T and iFormer-S following closely behind. In the SIRI-WHU evaluation, focused on complex land use and cover types, ESIF secures the top position again with a BA of 95.0% and an AA of 94.58%, outshining SwinV2-S and SpectFormer-H-S. For the high-resolution satellite images in the WHU-RS19 dataset, ESIF maintains unparalleled accuracy with a BA of 94.52% and an AA of 93.36%, ahead of DenseNet161 and EfficientNet-B3. Lastly, on the AID dataset, ESIF's BA of 93.5% stands out against the competitive accuracies achieved by DenseNet161 and MaxViT-T, marking it as the superior model for aerial scene recognition. Across all datasets, ESIF not only demonstrates its exceptional capability in integrating spatial, texture, and spectral information for RS image analysis but also establishes a new benchmark in classification accuracy, significantly surpassing the second-best DenseNet161 and other contenders like EfficientNet-B0 in several instances.

This remarkable performance underlines ESIF's advanced feature processing capabilities, affirming its competitive edge and versatility for diverse RS applications. Fig. 4 demonstrates Accuracy-Efficiency Tradeoff of the models using CES and CAS.

F. Ablation Study

Table III presents an ablation study conducted to scrutinize the contributions of different blocks within the Efficient Spectral Inception Former (ESIF) architecture, specifically evaluating the impact of the Texture Feature Alignment Block (TFAB), Inception Transformer Block (iFB), and Efficient SpectroFormer Block (ESFB) on the model's overall performance. Initially, the base model without TFAB, iFB, and ESFB achieves a Best Accuracy (BA) of 81.45% and an Average Accuracy (AA) of 80.82%, serving as a foundational benchmark. The integration of TFAB alone improves the model's performance, leading to an increase of 1.34% in BA and 1.5% in AA, which highlights the significance of texture analysis in improving classification accuracy. Further addition of iFB to the architecture with TFAB elevates BA by 0.54% and AA by 0.47%, indicating the importance of balancing local and global information through the iFB. A variant using RGB inputs instead of TFAB with iFB resulted in a notable decrease in performance, highlighting the inadequacy of raw RGB inputs in comparison to specialized texture features for this task. Incorporating the Large Kernel Attention (LKA) mechanism with TFAB and iFB reverses this decline, matching the BA and AA achieved with TFAB and iFB alone, which emphasizes the effectiveness of LKA in processing spatial information. The substitution of ESFB with Multi-Head Self Attention (MHSA) further propels the model to achieve significantly higher accuracies, with a BA of 85.21% and an AA of 84.94%, illustrating the critical role of frequency domain processing in enhancing the model's capability. The final configuration, which combines TFAB with LKA, iFB, and an LKA-based ESFB, culminates in the highest performance boost, achieving a BA of 86.55% and an AA of 85.48%, alongside a substantial reduction in memory access. This configuration exemplifies the synergistic effect of these blocks, highlighting their collective importance in establishing ESIF's state-of-the-art performance. The study conclusively demonstrates that while each component—TFAB, iFB, and ESFB—individually contributes to the model's efficiency and accuracy, their integration yields the most significant improvements, validating the architectural choices underpinning ESIF. Fig. 6. shows the Eigen-Cam [23] activation maps of TFAB, ESFB, CDFB, last conv activation of 4th Stage and first conv stage of 5th Stage of ESIF to showcase the effectiveness of each proposed block.

G. Performance Analysis on the Challenging classes of Optimal-31 Dataset

Table IV offers a detailed class-wise accuracy comparison for ten challenging classes within the Optimal-31 dataset, juxtaposing the performance of our Efficient Spectral Inception Former (ESIF) model against notable counterparts such as EfficientNet-B0, MobileNetV2, Van-B0, and FasterNet-T2. This granular analysis reveals the nuanced strengths and weaknesses of each model in recognizing specific scene types, with ESIF consistently showcasing superior or highly competitive performance across a majority of the classes. For classes like 'rectangular_farmland' and 'overpass', ESIF markedly outperforms its competitors, achieving top accuracies of 70.00% and 85.71%, respectively, highlighting its adeptness at handling intricate spatial patterns and textures. Notably, while 'commercial_area' and 'church' see stronger performances from Van-B0 and itself respectively, ESIF demonstrates its robustness with a substantial 70.58% accuracy in 'church', closely mirroring MobileNetV2's performance. In instances where detailed feature extraction is paramount, such as in 'freeway' and 'runway' categories, ESIF's accuracy peaks at 81.48% and 92.3% respectively, significantly surpassing the alternatives. This underscores ESIF's exceptional ability to process and integrate complex spatial, texture, and spectral information, ensuring precise classification even in challenging scenarios. Moreover, ESIF's architecture enables it to achieve the highest accuracies in 'mobile_home_park,' 'lake', 'industrial_area', and 'roundabout', with scores of 73.68%, 83.33%, 86.95%, and 92.85%, respectively. These results not only attest to the model's comprehensive feature representation capabilities but also to its versatility across diverse environmental and architectural contexts. In contrast, certain classes like 'commercial_area' see a dip in ESIF's performance, suggesting areas where the model's processing strategy might benefit from further refinement or adaptation. Nevertheless, the overarching trend within the table solidifies ESIF's position as a formidable solution for RS image classification, especially in deciphering complex scenes within the Optimal-31 dataset. The class-wise comparison underscores ESIF's advancements in achieving state-of-the-art accuracy, marking significant progress in the field and showcasing the model's potential in navigating the intricacies of RS data.

IV. Conclusion

The Efficient Spectral Inception Former (ESIF) architecture represents a significant advancement in RS scene classification, successfully addressing the challenges posed by the unique characteristics of RS images. By integrating spatial, texture, and spectral domain analyses through TFAB, ESFB, and iFB, ESIF effectively captures the comprehensive visual information necessary for accurate classification. The model's exceptional performance is validated through rigorous testing across diverse datasets, consistently outperforming existing state-of-the-art models. The ablation study further elucidates the crucial role of each component, underscoring the importance of multi-domain feature fusion in enhancing classification accuracy. Additionally, ESIF's design considerations, such as efficient computation and the ability to handle high intra-class variability and low inter-class distinction, make it particularly suited for RS applications. Future work will explore further optimizations and applications of ESIF, potentially extending its utility to other domains requiring fine-grained image analysis. The promising results obtained thus far emphasize ESIF's potential to revolutionize RS scene classification, offering a robust tool for earth observation and similar applications.

References

[1] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and remote sensing magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[2] 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 journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

[3] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[5] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Computational Visual Media*, vol. 9, no. 4, pp. 733–752, 2023.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[7] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 021–12 031.

[8] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[9] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," in *European conference on computer vision*. Springer, 2022, pp. 459–479.

[10] B. N. Patro, V. P. Namboodiri, and V. S. Agneeswaran, "Spectformer: Frequency and attention is what you need in a vision transformer," *arXiv preprint arXiv:2304.06446*, 2023.

[11] C. Si, W. Yu, P. Zhou, Y. Zhou, X. Wang, and S. Yan, "Inception transformer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 495–23 509, 2022.

[12] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[14] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:13756489

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[16] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE geoscience and remote sensing magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[17] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 842–857.

[18] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[20] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of vhr remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1155–1167, 2018.

[21] I. Dimitrovski, I. Kitanovski, D. Kocev, and N. Simidjievski, "Current Trends in Deep Learning for Earth Observation:An Open-source Benchmark Arena for Image Classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 18–35, 2023.

[22] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[23] M. B. Muhammad and M. Yeasin, "Eigen-cam: Class activation map using principal components," in *2020 international joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[24] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 270–279. [Online]. Available: https://doi.org/10.1145/1869790.1869829

[25] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, 2015.

[26] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 2108–2123, 2015.

[27] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. MaÎtre, "Structural high-resolution satellite image indexing," Vienna, Austria, 2010.

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

[29] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[30] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, "Symbolic discovery of optimization algorithms," 2023. [Online]. Available: https://arxiv.org/abs/2302.06675