From Features to Explanations: A Comparative Analysis of Feature-Based and Deep Learning Methods for Remote Sensing Applications

Bokai Hou, Jinhong Cai, Wanting Xue, Yijie Zhao, Yuntian Zhang
The University of New South Wales

***Abstract*—Aerial scene classification plays a critical role in various real-world applications, including urban planning, environmental monitoring, and disaster response. This study aims to develop and compare multiple computer vision approaches for the automatic classification of aerial scenes in large-scale remote sensing images. The research systematically evaluates both machine learning-based methods -exploring the effectiveness of different feature descriptors such as Local Binary Patterns (LBP) and Scale-Invariant Feature Transform (SIFT), combined with classifiers like k-Nearest Neighbors (kNN) and Support Vector Machines (SVM) - and deep learning-based methods, including architectures such as ResNet, EfficientNet and SENet. The result shows that the deep learning model is significantly superior to traditional methods in feature expression and classification accuracy. Among them, EfficientNet achieved a classification accuracy of 98% with its Compound Scaling structure, showing excellent applicability in remote sensing scene classification tasks. The study also addresses the challenges posed by imbalanced class distributions and applies data augmentation strategies to enhance model robustness. Additionally, Explainable AI (XAI) techniques, particularly the Gradient-weighted Class Activation Mapping (Grad-CAM) method, are employed to visualize and interpret the classification decisions.**

***Index Terms*—deep learning, remote sensing, classification, CNN, machine learning, interpretability**

I. Introduction

Remote sensing data acquired through satellite and unmanned aerial vehicle (UAV) imagery has become an indispensable tool for land use mapping, environmental monitoring, and a variety of geospatial applications. Effective scene classification can help enable timely and accurate resource management, policy making, and scientific analysis in a wide range of fields [1]. Traditional approaches to aerial scene classification rely on classical machine learning (ML) methods. While these pipelines may be effective, their performance depends heavily on the selection and tuning of feature descriptors, which may fail to capture the complex spatial patterns [2].

In recent years, deep learning—particularly convolutional neural networks (CNNs)—has revolutionized scene classification by enabling automated, data-driven feature extraction and hierarchical representation learning [3]. CNN-based methods have demonstrated significant improvements in classification accuracy on publicly available remote sensing datasets [4].

However, effective deployment of these methods in real-world scenarios faces additional obstacles. Class imbalance remains a persistent challenge. Data augmentation strategies, ranging from standard geometric transformations to advanced methods guided by class activation maps (CAMs), are increasingly being used to address these issues and improve generalization [5]. Another key aspect is model interpretability. As deep learning models become increasingly complex, understanding their decisions is critical for trust, validation, and compliance. The application of explanation techniques such as Grad-CAM enables visualization of the spatial evidence behind model predictions, supporting more transparent and reliable scene classification systems [6].

In light of this, our study presents a comprehensive comparative analysis of classical machine learning such as Local Binary Patterns (LBP) and Scale-Invariant Feature Transform (SIFT), combined with classifiers like k-Nearest Neighbors (kNN) and Support Vector Machines (SVM)) and deep learning methods such as ResNet, EfficientNet, and SENet for automatic aerial scene classification from satellite and UAV remote sensing imagery. We also address class imbalance via data augmentation, and integrate Grad-CAM-based interpretability into the evaluation framework. By rigorously benchmarking and explaining these methods, our work aims to improve the accuracy and transparency of remote sensing image classification systems and provide a reference for best practices and future developments in the field.

The dataset we used in this task is SkyView Aerial Landscape dataset [31]. It contains 15 landscape categories, with each category containing 800 high-quality images, and we randomly choose 20% of the 800 images as the test set, and use the remaining 80% from all categories as the training set.

(a) City (b) Agriculture (c) Mountain

Fig. 1: Samples of the dataset

II. Literature Review

A. Deep Learning for Remote Sensing

Research on deep learning for remote sensing scene classification has focused on hybrid and fusion architectures.

Early pipelines typically used handcrafted descriptors such as local binary patterns (LBP), color histograms , and texture features, followed by classifiers such as SVM or random forests (RF) [2]. However, subsequent studies have repeatedly demonstrated that deep learning models, especially CNNs, have superior accuracy for aerial scene classification, especially when CNN features are extracted or combined with classical representations [1] [4] [7] [8]. Specifically, a hybrid pipeline is adopted: a deep CNN (e.g. ResNet-50 or EfficientNet) for feature extraction, optionally augmented by an attention mechanism [4], followed by dimensionality reduction (PCA) and a classical classifier (usually SVM) for final prediction. Feature fusion strategies have shown significant performance gains [5] [9]. Empirical evaluations on public datasets support these conclusions. Notably, Carranza-García et al. [8]recommend a consistent cross-validation protocol when comparing classical and deep learning methods and show that CNN-based methods generally achieve higher performance on different land use and land cover datasets.

Due to the typical class imbalance in remote sensing datasets (some scene categories are heavily overrepresented), data augmentation is widely used to improve model generalization ability and support minority categories. Standard augmentation techniques such as rotation and flipping have been applied in many studies [1] [10]. Zhang and Cao [11]proposed a supervised data augmentation method that uses CAM to guide the augmentation process and ensure that the synthesized images retain semantically relevant regions that are important for scene recognition.

B. Interpretable AI

It is also important to understand how deep learning makes decisions and predictions. The goal of Explainable Artificial Intelligence (XAI) is to develop a series of techniques and methods to provide the ability to explain black box models.

The interpretability of deep aerial scene classifiers has been primarily addressed using Grad-CAM. Grad-CAM generates a heatmap visualization that highlights image regions that have an impact on the model's decision process [6] [10] [11]. This technique is a standard tool for improving transparency and user trust, not only in practical applications, but also for post hoc model explanations in research settings.

Dutta et al. [6]extended the interpretability paradigm and introduced Causal Gradient CAM (CG-CAM), which combines Grad-CAM with a Generative Flow Network (GFlowNet) to describe not only which regions are salient, but also the causal relationships between feature maps in CNNs and classification results. CG-CAM visualizes these relationships as a directed acyclic graph, representing a new advance in the interpretability of aerial scene classification models.

III. Comprehensive Method Development

A. LBP with KNN Classifier

1) Local Binary Pattern Feature Extraction: We took LBP as the feature descriptor. LBP is able to capture the local texture information in grayscale images and describe the spatial structure of local image texture.

It starts with counting the number of times each x-digit binary number occurs in the cell and achieved a 256-bin histogram (also known as the LBP feature vector), then combine the histograms of all cells of the given image and finally gives the image-level LBP feature descriptor [12], which provides a robust feature vector for typical classification tasks. Mathematically, the LBP code at a pixel (x_c, y_c) is defined as [13]:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \tag{1}$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

In my implementation, I have tested different combination pairs of the radius and number of points parameters, such as (1, 8), (2, 16), (3, 24) and (4, 36). By comparing, we choose the uniform LBP variant with Radius = 2 and Number of points = 16 since we found that this pair is the most efficient while ensuring the reasonable accuracy.

After we get the training and test datasets, we need to convert each image to grayscale using OpenCV, and then all LBP features of those images will be captured and converted into a normalized histogram. This results in a fixed-length vector for each image, suitable for input to our chosen classifier.

2) K-Nearest Neighbours Classifier: We took KNN as the classifier. KNN is a classifier that decides the class label for a sample based on the K nearest samples in the data set. It works as to recalculate the distances for each test example and use K nearest training samples to decide the class label of test samples, and assign to the corresponding class that has the most members in the neighbourhood.

To calculate the K-Nearest Neighbor value, we have

$$f(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} \gamma_i \in A_{VE}\{\gamma_i \mid x_i \in N_k(x)\} \tag{3}$$

where $N_k(x)$ consists of the k closest points to x in the sample.

When refering to the binary classification, it can be calculated as:

$$\gamma(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} \gamma_i \quad \in [0, 1] \tag{4}$$

$$f(x) = \begin{cases} 1 & \text{if } \gamma(x) > 0.5 \\ \\ 0 & \text{if } \gamma(x) < 0.51001[14] \end{cases} \tag{5}$$

3) GridSearch CV: To improve our overall performance, we applied a hyperparameter search using GridSearchCV in combination with 5-fold cross-validation. GridSearchCV evaluated various combinations of the following parameters:

```
{param_grid = 'n_neighbors':
[7, 9, 11, 13], 'weights':
['uniform', 'distance'], 'metric':
['euclidean', 'manhattan',
'chebyshev']}
```

We aim to use GridSearchCV to filter the parameter combination that maximizes our classification performance while ensuring the model is not overfitting. GridSearchCV internally splits the training set into five folds, trains and evaluates each combination and finally chooses the parameter combination that achieved the highest cross-validated performance.

After that, we obtained the optimal parameter combination in this task, which is:

```
{'metric': 'manhattan',
'n_neighbors': 11, 'weights':
'distance'}
```

This outcome shows that the 'manhattan' distance is the better choice for histogram-based LBP features in this task compared with 'euclidean' or 'chebyshev' distances, and the model performs best when we choose the neighborhood size to 11.

Then we use this parameter combination to retrain our final kNN model on the full training dataset, in the same time we remain to provide the 5-fold cross-validation and test evaluation.

4) Step-by-Step Workflow: The whole workflow of LBP-KNN is shown below:

1) Load images from class Google Colab Drive, resize them and convert to grayscale.
2) Apply Local Binary Pattern (LBP) with radius = 2, points = 16.
3) Extract the LBP features of the input datasets..
4) Build training and test datasets using extracted LBP features.
5) Use GridSearchCV with 5-fold cross-validation to find the best kNN parameter combination.
6) Train kNN classifier with best parameter combination on the full training set.
7) Evaluate on a test set using accuracy, precision, recall, F1-score, classification report, and confusion matrix.

B. SIFT with SVM Classifier

This subsection provides a conventional image classification pipeline built on a scale-invariant feature transform (SIFT), bag-of-words (BoW) encoding, and support vector machine (SVM), as indicated in [15] [16] [17].

1) Module Overview: SIFT Descriptor: SIFT generates local keypoints and creates 128-dimensional descriptors, which are invariant to scale and rotation. It forms a scale-space using:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{6}$$

where G is the Gaussian kernel and I is the image. Keypoints are found as extrema in the Difference-of-Gaussian (DoG) pyramid.

BoW Encoding:MiniBatchKMeans clusters all descriptors into K visual words. Each descriptor $\mathbf{x}_i \in \mathbb{R}^{128}$ is assigned to the nearest cluster center μ_j by:

$$\text{argmin}_j \|\mathbf{x}_i - \mu_j\|_2^2 \tag{7}$$

The resulting histogram vector $\mathbf{h} \in \mathbb{R}^K$ counts occurrences and is ℓ_2-normalized.

SVM Classification: The BoW histogram is used to train an RBF-kernel SVM:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2) + b\right) \tag{8}$$

Input features are standardized before classification.

2) Step-by-Step Workflow:

1) Resize images to 256×256 and convert to grayscale.
2) Extract dense SIFT descriptors.
3) Apply MiniBatchKMeans clustering to form visual vocabulary.
4) Encode each image into a normalized BoW histogram.
5) Train RBF-SVM using BoW features.
6) Evaluate using accuracy, precision, recall, F1-score, and confusion matrix.

C. ResNet Architecture

ResNet, or Residual Network, first introduced by He *et al.* in 2015 [18], brought a breakthrough in deep convolutional architectures by addressing the degradation problem encountered in very deep networks. Unlike traditional CNNs, which suffer from vanishing gradients as depth increases, ResNet uses residual connections—identity mappings that bypass one or more layers and are added directly to the output of stacked convolutional layers. These connections facilitate more effective backpropagation, allowing gradients to flow more smoothly and enabling deeper models to be trained without sacrificing accuracy.

TABLE I: Architectural Configuration of ResNet-18

Stage	Layer Type	Output Size	Kernel / Stride	Channels
Input	RGB Image	224×224	-	3
Conv1	7×7 Conv + BN + ReLU	112×112	7×7, 2	64
	3×3 MaxPool	56×56	3×3, 2	64
Stage 1	2 × Basic Blocks	56×56	3×3, 1	64
Stage 2	2 × Basic Blocks	28×28	3×3, 2	128
Stage 3	2 × Basic Blocks	14×14	3×3, 2	256
Stage 4	2 × Basic Blocks	7×7	3×3, 2	512
Classifier	Global AvgPool + FC	1×1	-	$512 \rightarrow N$

To accelerate convergence and enhance generalization, we initialized the network with pretrained weights from ImageNet. We modified only the final classification layer to accommodate the 15 classes in the dataset and fine-tuned the entire model for domain adaptation. We employed ResNet-18 as our backbone network due to its balance between depth and computational efficiency. The model was fine-tuned using a cross-entropy loss function and the Adam optimizer. Training was conducted for

three epochs per fold using 5-fold cross-validation, and final performance was assessed on a held-out test set.

Model training was conducted using the Adam optimizer with a learning rate of 1×10^{-3} and cross-entropy loss. We applied 5-fold cross-validation to assess the model's generalization across different data splits, with each fold trained for three epochs. The final evaluation was performed on a held-out test set. The training process exhibited a consistent decline in loss, dropping from 0.4569 in the first epoch to 0.0684 in the tenth, indicating effective learning and convergence. Notably, the loss decreased sharply in early epochs—falling to 0.2074 by the second epoch—highlighting the benefits of using pretrained weights.

D. EfficientNet Architecture

EfficientNet is a family of convolutional neural networks introduced by Tan and Le [19], which uses a compound scaling method. Unlike conventional scaling strategies that arbitrarily increase one dimension (e.g., only depth or width), EfficientNet achieves improved accuracy and efficiency through this balanced scaling.

1) ***Compound Scaling Method****:* This method uniformly scales the network's depth, width, and resolution using a single user-defined coefficient ϕ (phi). The scaling is defined as follows.

$$\begin{aligned} &\text{depth: } d = \alpha^{\phi}, \\ &\text{width: } w = \beta^{\phi}, \\ &\text{resolution: } r = \gamma^{\phi} \end{aligned} \tag{9}$$

subject to the constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad \text{and} \quad \alpha, \beta, \gamma > 0 \tag{10}$$

Here, α, β, and γ are constants determined by a small grid search, while ϕ controls the overall size of the model. This method allows the model to grow in a balanced way. It also ensures that additional computational resources are distributed proportionally across depth, width, and resolution.

2) ***Baseline****:* In this study, we introduce EfficientNet-B0. It's the baseline of the EfficientNet family, due to its great balance between performance and computational cost. The model is pre-trained on ImageNet [20].It provides strong general feature representations even when fine-tuned on relatively small datasets.

EfficientNet-B0 consists of a total of 9 stages. It starts with a standard 3×3 convolutional layer, followed by a series of MBConv blocks (Mobile Inverted Bottleneck Convolution) [23] with increasing depth and width of the feature map. Specially, the use of depthwise separable convolutions and squeeze-and-excitation (SE) modules [21] significantly improves both computational efficiency and model accuracy. The detailed architecture is shown in Table II and Fig2.

TABLE II: EfficientNet-B0 Architecture

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 + Pooling + FC	7×7	128	1

E. SENet Architecture

SENet (Squeeze-and-Excitation Networks) improves the representation ability of the model through Channel-wise Feature Recalibration [21]. Different channels in remote sensing images may correspond to differentiated features of multispectral or hyperspectral information, while traditional convolution operations treat all channels equally and may ignore the importance of key channels. The SE module dynamically adjusts channel weights to highlight important features and suppress redundant information, thereby improving classification performance.

In terms of model architecture, the backbone network uses the pre-trained SE-ResNet50 as a feature extractor, and its core improvement is the SE module. The SE module compresses spatial information through global average pooling to generate channel descriptors (Squeeze); and uses two fully connected layers (FC) to dynamically learn channel weights (Excitation). To adapt to remote sensing tasks, the last two layers (global average pooling and fully connected layers) were removed from the ResNet18 backbone network, and a channel attention module (SE Block) was inserted.

$$\mathbf{s} = \sigma\left(\mathbf{W}_2\, \delta\left(\mathbf{W}_1 \mathbf{z}\right)\right), \quad \tilde{\mathbf{x}}_c = s_c \cdot \mathbf{u}_c \tag{11}$$

In terms of training strategy, a fixed configuration of 10 cycles and a basic learning rate of 0.001 was adopted, and the batch size was set to 128 (training)/64 (testing). The performance difference between Adam and SGD optimizers with momentum (0.9) was compared. In order to analyze the impact of key components, four groups of ablation experiments were designed (A: standard enhancement + Dropout + Adam; B: no Dropout; C: strong enhancement; D: SGD optimizer), and the contribution of data enhancement strength, regularization method and optimizer selection to model performance was systematically explored.

IV. Advanced Method Development

A. Imbalanced Classification Enhancement

To solve the challenges of long-tail classification, we propose a framework combining distribution simulation and dynamic re-weighting, which systematically reduces model bias of head classes.

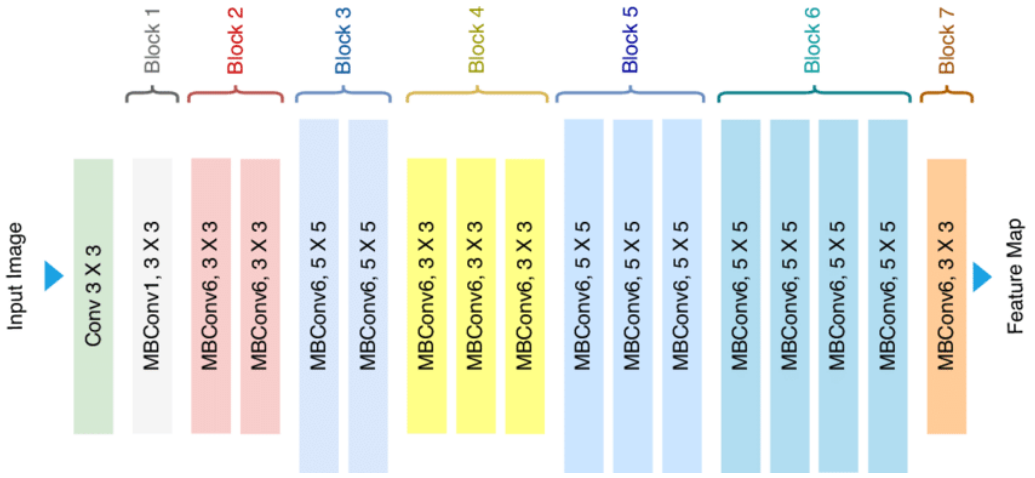


Fig. 2: The EfficientNet architecture begins with a 3×3 convolutional layer for low-level feature extraction, followed by seven sequential MBConv blocks - efficient inverted residual blocks primarily using MBConv6 with 3×3 or 5×5 kernels.

1) Long-tail Distribution Simulation: We construct a linearly decaying class distribution through resampling:

$$N_c = \max(800 - 50c,\ 50), \quad c \in \{0, 1, \ldots, 14\} \tag{12}$$

where N_c represents the sample count for class c, ensuring the tail class ($c = 14$) retains 50 samples as the minimum viable dataset. [22]

Adaptive Resampling Protocol:

- *Oversampling*: For classes with $N_c > |\mathcal{D}_c^{\text{orig}}|$:

$$\text{Repeats} = \left\lfloor \frac{N_c}{|\mathcal{D}_c^{\text{orig}}|} \right\rfloor \tag{13}$$

$$\text{Remainder} = N_c \mod |\mathcal{D}_c^{\text{orig}}| \tag{14}$$

- *Undersampling*: For classes with $N_c < |\mathcal{D}_c^{\text{orig}}|$, randomly select N_c samples without replacement. [23]

This approach preserves intra-class diversity while enforcing controlled imbalance ratios. [24]

2) Class Re-weighting Strategy: **Loss Function Balancing**: To offset the model's bias toward head classes, we scale each class's contribution to the loss function by the inverse of its sample count. Specifically, the loss for class c is weighted by $1/N_c$, where N_c is the number of training samples in class c (see Section IV-A1). This forces the model to pay more attention to tail classes during parameter updates.

Balanced Batch Sampling: We further ensure balanced class representation in each training batch through weighted random sampling. The probability of selecting a sample from class c is proportional to $1/N_c$, which guarantees that tail-class samples appear frequently in training. This dual strategy—weighted loss and balanced sampling—is inspired by the class-balanced learning paradigm [25].

3) Progressive Region Augment: To address the challenge of class imbalance in our dataset, we propose an Adaptive Progressive Augmentation strategy. The strategy is implemented via a custom module named `ProgressiveRegionAugment` (PRA). This method dynamically adjusts the augmentation intensity for each sample based on the relative frequency of its class.

a) Mechanism.: For each sample with class label y, we compute an augmentation coefficient α based on the class frequency f_y as follows [26]:

$$\alpha = 1 - \frac{f_y}{f_{\max}}$$

where $f_{\max}$ is the count of the most frequent class. This $\alpha \in [0, 1]$ controls the strength of augmentation operations.

b) Augmentation Stages.: PRA consists of two main stages:

- **Basic Augmentation:** All images are processed by a standard set of transformations including resizing, random cropping, and horizontal flipping.
- **Region-Aware Augmentation:** Based on the value of α, additional transformations are selectively applied:
  - *Sharpness adjustment* and *random rotation*, increasing with α.
  - *Perspective distortion* with randomly perturbed corner points when $\alpha > 0.5$.
  - *Random erasing* [27] with probability and area scale adjusted by α.

B. Explainable AI Analysis

1) Grad-CAM Implementation: To enhance the interpretability of the trained models and understand the spatial focus of their decisions, we utilized Gradient-weighted Class Activation Mapping (Grad-CAM) [28]. Grad-CAM is a widely used technique that generates visual explanations by leveraging the gradients of a target concept flowing into the final convolutional layer of a CNN. It produces coarse localization maps that highlight the important regions in an input image that contribute most to the model's decision.

Grad-CAM works by first computing the gradients of the output class score with respect to the feature maps of the

last convolutional layer. These gradients are then global-average-pooled to obtain importance weights, which are used to compute a weighted sum of the feature maps. This results in a class-discriminative heatmap that is upsampled and superimposed onto the original image, thereby visually indicating which parts of the image influenced the prediction.

2) Adversarial Testing Framework: To assess the robustness and interpretability of the model, we implemented adversarial evaluation through Grad-CAM. In this case, an image from the class *Agriculture* was utilized to create three perturbation types: Gaussian noise, Gaussian blur, and black-box occlusion. While the input was visually corrupted, the model responded the same; it predicted the correct class and provided semantically relevant attention maps across all of the variations.

V. Experimental Results

A. LBP with KNN Classifier

For model validation, we used the method 5-fold cross-validation on the training set while we also obtain the best parameters by using GridSearchCV:

```
{'metric': 'manhattan',
'n_neighbors': 11, 'weights':
'distance'}
```

Next to ensure the generalization of the model, our final model will be retrained on the full training set and tested on our test set.

The metrics used in evaluation are accuracy, precision, recall, and F1-score. We report macro-averaged results to give equal weight to each class, and finally we will provide the corresponding classification report and confusion matrix.

1) 5-fold Cross-Validation Results: The table below summarizes the 5-fold cross-validation results averaged across all folds:

TABLE III: Each fold result and average of the 5-fold cross validation

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fold 1	52.73	53.20	52.72	51.95
Fold 2	53.93	54.54	53.92	53.47
Fold 3	56.20	56.92	56.20	55.68
Fold 4	52.97	53.51	52.97	52.49
Fold 5	54.17	54.65	54.17	53.90
Average	54.00	54.56	53.99	53.50

From the result, we can see the outcome is consistent in each fold. The relatively uniform scores indicate that our model generalizes reasonably well across different splits and prevents the overfitting issue.

We can notice that the accuracy, precision, recall, and F1-score value we achieved from the test set is similar to previous 5-fold cross validation results, which indicates that our model has good generalization ability.

According to the classification report, we can find that the categories such as Forest (F1-score = 0.78), Mountain (F1-score = 0.72), and Port (F1-score = 0.70) achieved the highest F1-score, while visually ambiguous categories such as Airport (F1-score = 0.40) and River (F1-score = 0.36) were the most challenging among the 15 landscape categories.

We also produce the corresponding confusion matrix:

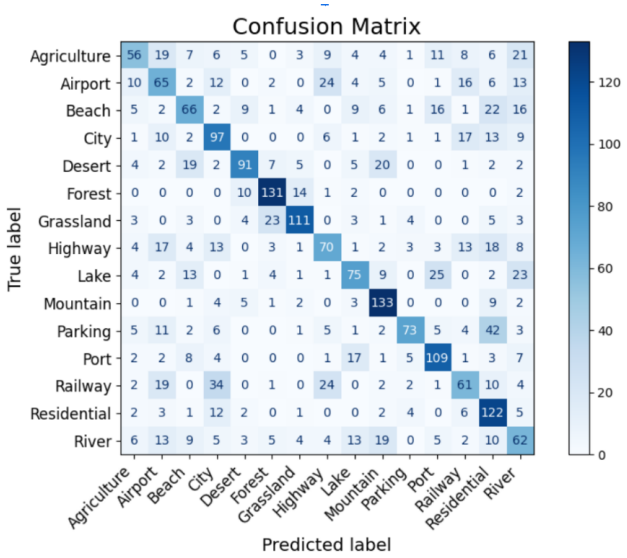


Fig. 3: Confusion matrix of LBP-KNN classifier on test set

The confusion matrix reveals common results, which our model can classify the category Mountain, Forest and Port well. Meanwhile it might misclassify some patterns, where there is confusion between Parking and Residential, and between Railway and City.

2) ablation studies: In the actual testing stage, we compare different combination pairs of the radius and number of points parameters, these are the results comparison of (1 ,8), (2, 16) and (3, 24).

TABLE IV: Performance comparison of different (radius, points) parameter combinations on the test set

(R, P)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
(1, 8)	52.17	52.53	52.17	51.78
(2, 16)	55.08	56.15	55.08	54.65
(3, 24)	54.62	54.95	54.62	54.20

When the radius and number of points parameters is (1, 8), the time cost is 5 minutes, when (2, 16) costs 7 minutes and when (3, 24) costs 9 minutes, and refering to the performance, we can notice that (2, 16) achieves the best performance among these threes. So by comparing, we choose the uniform LBP variant with Radius = 2 and Number of points = 16 since we found that this pair has the best performance while ensures the reasonable efficiency.

B. SIFT with SVM Classifier

1) Hyperparameter Tuning and Result Interpretation:

a) Vocabulary Size (k): We evaluated $k = \{20, 50, 100, 150\}$ for visual word vocabulary size. The

best performance was obtained at $k = 100$, offering a balance between overfitting and underfitting.

- **Too small** k: Poor feature discrimination due to under-representation of visual diversity.
- **Too large** k: Results in sparse histograms and amplifies noise, reducing generalization.

b) Mini-batch Size in KMeans: MiniBatchKMeans was tested with batch sizes of 512, 1024, and 2048. A batch size of 1024 achieved the most stable convergence and good clustering quality.

- **Smaller batches**: Noisy updates and poor centroid convergence.
- **Larger batches**: Better gradient estimation but increases memory consumption and processing time.

c) SVM Parameters: Our comparative analysis of support vector classifiers revealed significant performance differences:

- `SVC(kernel='rbf')` demonstrated 10% higher accuracy than `LinearSVC`
- Optimal regularization parameter $C^* = 3$ via validation curve analysis
- **RBF Kernel:** Projects data into a higher-dimensional space, enabling non-linear decision boundaries.
- **Parameter** C**:** Controls the trade-off between maximizing the margin and minimizing classification error.

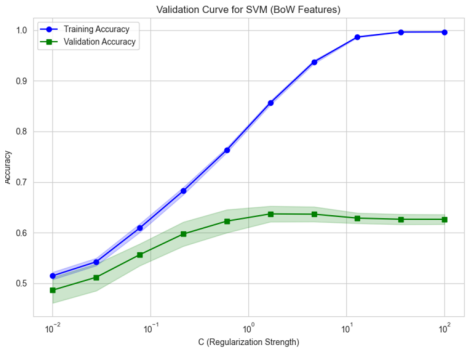


Fig. 4: Validation Curve for SVM with BoW features

2) Result Analysis: The model achieved 67.12% test accuracy, with precision, recall, and F1-score all around 67%. Structured categories (e.g., *City*, *Residential*) performed better due to rich edge features. Low-texture classes (e.g., *Beach*, *Lake*) suffered from feature ambiguity. The confusion matrix (Fig. 5) reveals typical class-wise performance.

TABLE V: Final Evaluation of SIFT+SVM on Test Set

Metric	Score (%)
Accuracy	66.85
Precision	67.01
Recall	66.92
F1-score	66.79

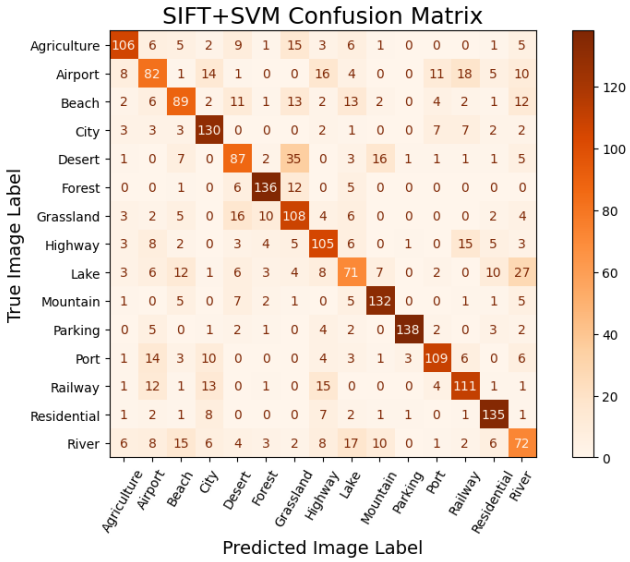


Fig. 5: SIFT+SVM Confusion Matrix and Classification Metrics

C. ResNet Architecture

1) Test Set Performance: Our final ResNet model achieved the following results on the test dataset:

- Accuracy: 95.08%
- Precision: 95.25%
- Recall: 95.08%
- F1-score: 95.08%

Classification results across categories are shown in Table VI.

TABLE VI: Classification Report on Test Set

Category	Precision	Recall	F1-score	Support
Agriculture	0.95	0.95	0.95	160
Airport	0.96	0.86	0.91	160
Beach	0.98	0.97	0.98	160
City	0.88	0.99	0.94	160
Desert	0.98	0.93	0.96	160
Forest	0.98	0.97	0.97	160
Grassland	0.98	0.95	0.97	160
Highway	0.95	0.96	0.95	160
Lake	0.94	0.93	0.94	160
Mountain	0.86	0.99	0.92	160
Parking	0.98	0.99	0.98	160
Port	0.98	0.99	0.98	160
Railway	0.93	0.94	0.93	160
Residential	0.99	0.98	0.98	160
River	0.93	0.86	0.90	160

2) Ablation Study: We evaluated ResNet18 and ResNet50 on the SkyView aerial landscape dataset, which consists of 15 terrain categories with 800 high-resolution samples per class. Despite its simpler architecture, ResNet18 outperformed ResNet50 in this task. The final test accuracy of ResNet18 reached **95.08%**, whereas ResNet50 achieved **93.29%**. Detailed classification metrics are shown in Table VIII.

TABLE VII: 5-Fold Cross-Validation Results of ResNet

Fold	Precision	Recall	F1-score
Fold 1	0.9039	0.8906	0.8913
Fold 2	0.9068	0.8849	0.8832
Fold 3	0.8785	0.8583	0.8531
Fold 4	0.9322	0.9271	0.9276
Fold 5	0.9273	0.9198	0.9206
Average	**0.9097**	**0.8961**	**0.8952**

TABLE VIII: ResNet18 Test Performance on SkyView Dataset

Metric	Accuracy	Precision	Recall	F1-score
ResNet18	95.08%	95.25%	95.08%	95.08%
ResNet50	93.29%	93.45%	93.29%	93.29%

We conducted a 5-fold cross-validation to further assess the models' robustness. The average performance across folds is summarized in Table IX.

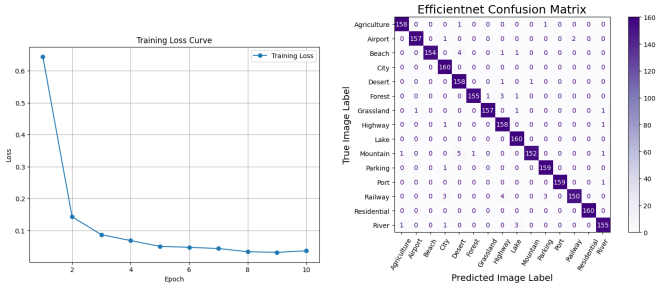
TABLE IX: 5-Fold Cross-Validation Performance Comparison

Model	Avg. Precision	Avg. Recall	Avg. F1-score
ResNet18	**90.97%**	**89.61%**	**89.52%**
ResNet50	90.63%	89.07%	89.01%

Although the differences in cross-validation scores are subtle, ResNet18 consistently shows marginally better generalization, particularly in recall and F1-score. This consistency supports the hypothesis that the lightweight architecture is more robust to varying data partitions, likely due to its reduced complexity and lower sensitivity to small-scale overfitting.

D. EfficientNet Architecture

The experimental results of using EfficientNet-B0 in our dataset. Although the model has a compact structure, its accuracy rate reaches 98%. Expanding to larger variants (B1-B7) will increase computational cost while providing limited accuracy gains. So we choose EfficientNet to explore advanced methods.



(a) Training loss curve (b) Confusion matrix

Fig. 6: Results of EfficientNet-B0

E. SENet Architecture

This part explores the performance of SENet model in remote sensing image classification task through ablation experiment. Four groups of control experiments are designed to focus on three key variables: data augmentation strength (standard and strong), Dropout regularization (with/without), and optimizer selection (Adam/SGD).

TABLE X: Ablation Experiment Design

Experiment	Data Augmentation	Dropout	Optimizer
A (Baseline)	Standard	✓	Adam
B	Standard	×	Adam
C	Strong	✓	Adam
D	Standard	✓	SGD

The experiments are based on ResNet-18 backbone network with 10 epochs. Experiment A is the baseline model, which uses standard data augmentation, enables Dropout and uses Adam optimizer. Based on Experiment A, Experiment B does not use Dropout, Experiment C uses data augmentation strategy, and Experiment D uses SGD optimizer.

The experimental results show that Experiment A has moderate overfitting. The performance of the model in Experiment B is oscillating. Although the mid-term accuracy of Experiment B is 93.92%, the final test accuracy is 85.30%, indicating that Dropout plays a key role in suppressing model overfitting. The strong data augmentation strategy of Experiment C prolongs the training time by about 60%, but the accuracy is 93.92%, which is a moderate effect.

TABLE XI: Final Test Accuracy of Ablation Experiments

Experiment	Final Test Accuracy (%)
A (Baseline)	92.13
B (No Dropout)	85.30
C (Strong Augmentation)	93.92
D (SGD Optimizer)	95.50

Experiment D is the best in terms of effect. As the training progressed, its verification accuracy continued to rise, reaching 95.50% in the 10th epoch, and the training loss decreased monotonically, with the final test accuracy also being 95.50%. The reason for the good effect of Experiment D is that the SGD optimizer was used, which enhanced the model's sensitivity to subtle feature differences by introducing moderate parameter update noise. In-depth analysis of the classification errors shows that the model is confused between categories with similar morphological or spectral features. For example, Railway was misclassified as Airport or City 13 times, mainly due to the visual similarity of linear structures. The misclassification of Lake and River was concentrated in samples with blurred boundary morphology (8 times), which is speculated to be related to the problem of capturing the dynamics of water bodies. The 5 misclassifications of Mountain and Desert were related to the terrain shadow characteristics. These error modes are inherent challenges in remote sensing image interpretation. The confusion matrix of Experiment D further demonstrates the limitations of this method. The confusion of transportation facilities, water bodies, and landforms shows that the model still does not make sufficient use of contextual information

and fails to effectively integrate the surrounding environmental features.

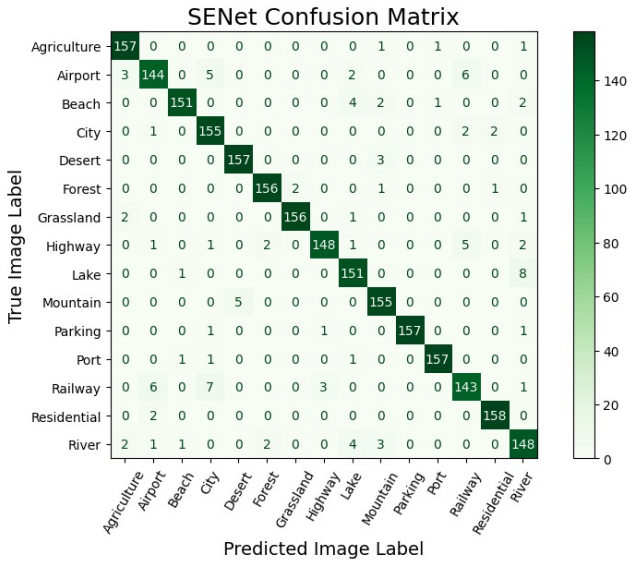
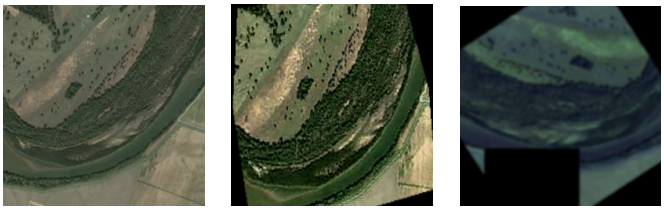


Fig. 7: SENet Confusion Matrix and Classification Metrics

F. Imbalanced Classification Enhancement

Current data augmentation methods (including PRA and standard augmentation) don't work well for long-tail geographic data. The problem is simple: normal augmentation treats all parts of an image the same way. But in real geographic images, different features have different rules. For example, you can't randomly rotate a mountain - it would look wrong. When augmentation creates these impossible "fake" images, the model learns wrong patterns instead of real geographic features. This makes the model perform worse because it gets confused between real and impossible features.



(a) Original (b) PRA Augmented (c) Class-specific

Fig. 8: Comparative visualization of river samples: (a) Original river; (b) PRA augmentation; (c) Class-specific augmentation

The accuracy decline in River-class samples is directly related to three negative effects of augmentation:

1) *Structural Discontinuity*: Random erasure may lead to occluding river meanders and misclassification as separate water lines [29]
2) *Flow Direction Violation*: Rotations exceeding 15° create samples that conflict with natural fluvial patterns
3) *Spectral Deviation*: Color jittering induces water reflectance shifts (e.g., transforming clear-water features to turbid-water features). [30]

G. Grad-CAM Implementation

In our implementation, we applied Grad-CAM to the final convolutional layer of the best-performing model, EfficientNet-B0. We selected EfficientNet because it achieved the highest accuracy among all models tested. The last convolutional layer used for Grad-CAM was `model._blocks[-1]._project_conv`, which captures high-level spatial features crucial for classification. For each class in the SkyView dataset, one representative test image was randomly selected to generate the Grad-CAM visualization.

Figure 9 illustrates the results of Grad-CAM applied to several classes. On the left are the original input images, and on the right are the corresponding Grad-CAM heatmaps overlaid on the images. The highlighted regions reveal that the model correctly focuses on semantically meaningful areas (e.g., forest canopies, roadways, shorelines) when making its predictions. This provides confidence in the reliability and interpretability of the model's decision-making process.

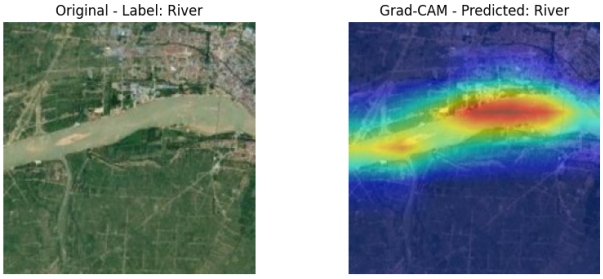


Fig. 9: Grad-CAM visualization results for River classes. Left: original input image. Right: Grad-CAM overlay indicating regions most influential to the model's classification.

H. Adversarial Testing Framework

To assess the robustness and interpretability of the model, we implemented adversarial evaluation through Grad-CAM. In this case, an image from the class *Agriculture* was utilized to create three perturbation types: Gaussian noise, Gaussian blur, and black-box occlusion. While the input was visually corrupted, the model responded the same; it predicted the correct class and provided semantically relevant attention maps across all of the variations.

Grad-CAM heatmaps indicated that the model was still attending to meaningful areas of the signal, even with degraded inputs. The occluded version particularly demonstrated that the model simply redistributed its attention to other regions that were visible and unmasked, while again predictions were correctly made. This indicates some robustness to local distortion.

The observations from these experiments indicate that the model has learned robust, generalizable features, not simply surface features that could be memorized. This adulteration process provides a framework to be expanded upon that can apply more complex attacks to more than two classes, providing understanding about model robustness to ambiguity in real-world inputs.

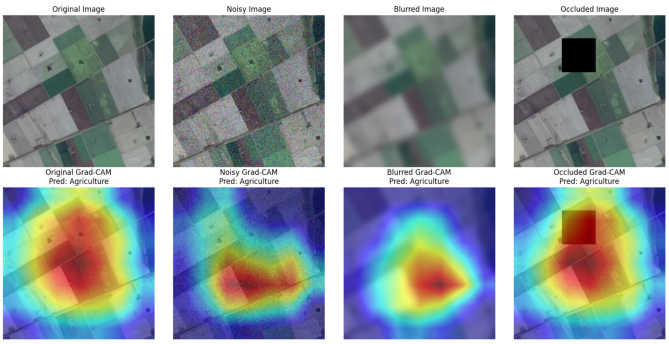


Fig. 10: Grad-CAM outputs for original and perturbed images in the *Agriculture* class

VI. Discussion

The results show the optimal solution for each model in the ablation study. The performance of the combination of traditional feature extraction methods and machine learning classifiers is relatively limited. Specifically, the accuracy of LBP with KNN is only 55.08%, while the accuracy of SIFT with SVM is improved to 67.12%. Although SIFT can capture the local spatial features of the image well, the traditional method has poor adaptability to the complex scenes of remote sensing images and is difficult to effectively deal with various practical problems such as scale, texture and lighting.

However, deep learning performance is significantly better. ResNet and SENet achieved 95.08% and 95.50% accuracy respectively. This proves that deep network structures and attention mechanisms are of great value for remote sensing image learning. The EfficientNet architecture performed best in this experiment, with an accuracy of 98.00%, and Precision and Recall were also around 98%. It fully demonstrates that Compound Scaling can achieve a better balance between model depth, width and input image resolution, significantly improving classification results. In summary, deep neural networks have demonstrated feature expression and generalization capabilities that far exceed traditional methods in remote sensing classification tasks.

TABLE XII: Classification Performance Comparison Across Different Models

Model	Accuracy	Precision	Recall	F1-score
LBP+KNN	55.08%	56.14%	55.08%	54.65%
SIFT+SVM	67.12%	67.13%	67.12%	66.96%
ResNet	95.08%	95.25%	95.08%	95.08%
EfficientNet	98.00%	98.05%	98.00%	98.00%
SENet	95.50%	95.55%	95.50%	95.50%

In addition, the Progressive Region Augment and the loss adjustment strategy based on category weight designed in this paper can improve the accuracy of tail class recognition to a certain extent, but expose the limitations of traditional image enhancement methods in the field of remote sensing. Especially for water body and landform images, the enhancement operation may destroy their geographical structure, thereby reducing the classification accuracy. This phenomenon reminds us that the remote sensing data enhancement method should follow the physical logic of the geographical scene and avoid blind deformation that ignores terrain constraints.

Finally, through the visualization analysis of Grad-CAM, this study further verifies whether the attention logic of the high-accuracy model is consistent with human cognition. It should be noted that the areas of interest of EfficientNet and SENet are highly consistent with the core features of the scene, while the features of traditional models are mostly concentrated on the local edges of the image, which makes it difficult to capture global relationships. Moreover, in the adversarial sample test, the model can still adjust the area of interest and maintain a correct prediction in the face of different types of interference,ce, showing good robustness. This shows that the deep model trained in this study has a stable spatial feature recognition mechanism, which provides an explanation for the engineering practice of remote sensing images.

VII. Conclusion

This study systematically compared the performance of traditional computer vision methods and deep learning in remote sensing image classification problems. The results show that deep learning models, such as the EfficientNet architecture, have significant advantages in feature extraction and classification accuracy. In addition, this study introduced the Grad-CAM interpretability mechanism to improve the transparency and trust of the model. To address the problem of sample class imbalance, the study proposed a combined strategy of data augmentation and loss reweighting, and analyzed its actual effect and scope of application. Future work can introduce geographic prior knowledge to improve the robustness of the model, and even link interpretability to dynamic prediction and decision support work.

References

[1] Enhancing Remote Sensing Image Classification and Interpretability: A Multi-Stage Feature Extraction Approach and Grad-CAM, 2025.
[2] Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?, 2015.
[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[4] A Hybrid Machine Learning and Deep Learning Approach for Remote Sensing Scene Classification, 2023.
[5] Dense Connectivity Based Two-Stream Deep Feature Fusion Framework for Aerial Scene Classification, 2018.
[6] Toward Causality-Based Explanation of Aerial Scene Classifiers, 2024.
[7] Deep Learning for Feature Extraction in Remote Sensing: A Case-Study of Aerial Scene Classification, 2020.
[8] A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks, 2019.
[9] Robust Object Categorization and Scene Classification over Remote Sensing Images via Features Fusion and Fully Convolutional Network, 2022.
[10] Explainable AI: Scene Classification and GradCam Visualization, 2024.
[11] A New Data Augmentation Method of Remote Sensing Dataset Based on Class Activation Map, 2021.
[12] Z. Sedaghatjoo, H. Hosseinzadeh, and B. S. Bigham, "Local Binary Pattern (LBP) Optimization for Feature Extraction," arXiv preprint, arXiv:2407.18665, 2024. https://doi.org/10.48550/arXiv.2407.18665

[13] M. A. Rahim, M. N. Hossain, T. Wahid, and M. S. Azam, "Face recognition using local binary patterns (LBP)," Global Journal of Computer Science and Technology: Graphics & Vision, vol. 13, no. 4, pp. 1–8, 2013.

[14] Kulkarni, N., & Bairagi, V. (2018). Chapter 5: Classification algorithms in diagnosis of Alzheimer's disease. In *EEG-Based Diagnosis of Alzheimer Disease: A review and novel approaches for feature extraction and classification techniques* (pp. 61–71). https://doi.org/10.1016/B978-0-12-815392-5.00005-8

[15] Gao H, Dou L, Chen W, et al. Image classification with bag-of-words model based on improved sift algorithm[C]//2013 9th Asian Control Conference (ASCC). IEEE, 2013: 1-6.

[16] Li Q, Wang X. Image classification based on SIFT and SVM[C]//2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). IEEE, 2018: 762-765.

[17] Naeem H, Bing G, Naeem M R, et al. A new approach for image detection based on refined Bag of Words algorithm[J]. Optik, 2017, 140: 823-832.

[18] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[19] Lee H B, Lee H, Na D, et al. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks[J]. arXiv preprint arXiv:1905.12917, 2019.

[20] Nitesh V C. SMOTE: synthetic minority over-sampling technique[J]. J Artif Intell Res, 2002, 16(1): 321.

[21] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141. https://doi.org/10.1016/B978-0-12-815392-5.00005-8

[22] Liu Z, Miao Z, Zhan X, et al. Large-scale long-tailed recognition in an open world[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 2537-2546.

[23] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[24] Cui Y, Jia M, Lin T Y, et al. Class-balanced loss based on effective number of samples[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 9268-9277.

[25] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[26] Ma S, Zhou Y, Gowda P H, et al. Application of the water-related spectral reflectance indices: A review[J]. Ecological indicators, 2019, 98: 68-79.

[27] Ayush K, Uzkent B, Meng C, et al. Geography-aware self-supervised learning[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 10181-10190.

[28] Selvaraju R R, Cogswell M, Das A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[C]//Proceedings of the IEEE international conference on computer vision. 2017: 618-626.

[29] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International journal of computer vision, 2015, 115: 211-252.

[30] Zhong Z, Zheng L, Kang G, et al. Random erasing data augmentation[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 13001-13008.

[31] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang 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.