1. Introduction （Yiheng Zhang）

Aerial scene classification task is a basic task in the field of computer vision, which aims to automatically assign semantic labels to aerial or satellite images. This has a wide range of applications in many fields, such as urban design, agricultural planning, etc. With the advancement of technology, the resolution of aerial images is getting higher and higher, which also means that the accuracy of the classification system is also increasing.

At present, this task faces many challenges. For example, the scene structures of the same class in different regions vary greatly, and it is difficult to determine the common features between them. Or, the differences between different scenes are small and detailed, making it difficult to distinguish different scenes. In addition, various factors such as season, lighting and shooting angle make image classification more complicated.

To complete the task，many CV methods have been developed, from traditional machine learning approaches using hand-crafted features to modern deep learning architectures. Traditional CV methods provide a certain degree of interpretability and computational efficiency for image classification through the combination of feature selection and classifiers. Deep learning models can learn hierarchical features directly from raw image data, thereby capturing more complex spatial and semantic patterns.

In our project, we implemented and evaluated two types of methods, traditional models and deep learning models, on the SkyView dataset, which contains 12 000 images evenly distributed across 15 landscape categories.The traditional models include four combinations of two feature selectors (HOG/LBP) and two classifiers (SVM/KNN). The focus is on deep learning models, including ResNet-18, DenseNet, EfficientNet, and Vision Transformer (ViT), which represent different architectural paradigms from residual learning to attention-based modeling. Through a unified experimental framework, we compare the performance of traditional models and deep models under consistent training settings and data augmentation strategies to evaluate their effectiveness in aerial scene classification.

1. Literature Review

Background of Aerial Scene Classification  
 Aerial scene classification from remote sensing imagery is fundamental for urban planning, environmental monitoring, and disaster response. Challenges include high image resolution, diverse textures, and fine-grained category distinctions. In this regard, many scholars have provided relevant methods and insights into this type of research.

1. Traditional Machine Learning Methods

HOG + SVM/KNN: The Histogram of Oriented Gradients (HOG) descriptor, introduced by Dalal & Triggs (2005), captures edge and shape information via gradient orientation histograms. Combined with Support Vector Machines (SVM) or k-Nearest Neighbors (KNN), it achieved strong performance in early scene classification tasks【1】.

LBP + SVM/KNN: Local Binary Patterns (LBP), proposed by Ojala et al. (2002), encode local texture by thresholding each pixel against its neighbors. LBP features are lightweight and illumination‑invariant, often paired with SVM or KNN for small‑sample texture and scene classification【2】.

2. Convolutional Neural Networks (CNNs)

ResNet: He et al. (2016) introduced Residual Networks (ResNet), which employ identity skip‑connections to mitigate vanishing gradients and degradation in deep networks. ResNet‑18, with 18 layers, is a popular lightweight baseline for transfer learning on remote sensing datasets【3】.

DenseNet: Huang et al. (2017) proposed Densely Connected Convolutional Networks (DenseNet), where each layer connects to every subsequent layer, promoting feature reuse and efficient gradient flow. DenseNets outperform equivalent‑depth ResNets with fewer parameters【4】.

EfficientNet: Tan and Le (2019) developed EfficientNet, which uniformly scales network width, depth, and input resolution by a compound coefficient. Models from B0 to B7 achieve state‑of‑the‑art accuracy on ImageNet with superior FLOPs efficiency【5】.

3. Vision Transformers (ViT)

ViT: Dosovitskiy et al. (2021) applied the Transformer architecture to vision by dividing images into fixed‑size patches and treating them as a sequence. Self‑attention captures global context dependencies. ViT, after large‑scale pretraining, can match or surpass CNNs but demands substantial pretraining data【6】.

Summary  
 This project compares representative methods from traditional hand‑crafted features to modern CNNs and Vision Transformers, evaluating their performance under various augmentation strategies on the SkyView aerial scene dataset.

1. Methods
2. Traditional(HOG/LBP+SVM/KNN) (Yiheng Zhang)

In the traditional pipeline for aerial image classification, we employ handcrafted feature extraction followed by shallow learning classifiers. Specifically, we compare the effectiveness of two widely used texture descriptors — Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP) — each combined with two classical classifiers: Support Vector Machine (SVM) and k-Nearest Neighbors (KNN). This results in four configurations: HOG+SVM, HOG+KNN, LBP+SVM, and LBP+KNN.

### **1. Feature Extraction**

**Histogram of Oriented Gradients (HOG)** extracts edge and shape information by computing the distribution of gradient orientations across localized cells in the grayscale image. In our implementation, images are resized and converted to grayscale, and HOG features are extracted using a cell size of 16×16 pixels and a 2×2 block size, producing a flattened feature vector for each image.

**Local Binary Patterns (LBP)** encode texture by thresholding the local neighborhood around each pixel. We use uniform LBP encoding with 8 neighbors (P=8) and a radius of 1 (R=1). The resulting LBP map is summarized using normalized histograms, which serve as compact texture descriptors.

### **2. Classifiers& Parameter Tuning**

Two classical machine learning classifiers are used:

* **Support Vector Machine (SVM)**: we tune the regularization parameter `C` using grid search over {0.01, 0.1, 1, 10} and compare two kernel types {`linear`, `rbf`},
* **k-Nearest Neighbors (KNN)**: We use Euclidean distance and test values of `k ∈ {3, 5, 7}`.

### **3. Training and Evaluation**

The dataset is split into 80% training and 20% testing subsets. All four traditional model combinations are trained and evaluated separately. Performance is assessed using accuracy, precision, recall, and macro F1-score. Additionally, confusion matrices are plotted to analyze class-wise performance and misclassification patterns.

1. Resnet

We chose ResNet‑18 as our backbone for three main reasons:

1. Residual Learning: ResNet’s identity skip‑connections enable training of deeper networks by learning residual functions 𝐹(𝑥) = 𝐻(𝑥) − 𝑥, which alleviates vanishing gradients and degradation problems.
2. Lightweight yet Effective: With only 18 layers, ResNet‑18 strikes a balance between expressive capacity and computational cost, making it suitable for our moderate‑sized dataset and limited hardware.
3. Proven Transfer Performance: Pretrained ResNet‑18 on ImageNet has demonstrated strong feature generalization on various downstream tasks, including aerial scene classification.

Highlighting advantages over other popular architectures:

1. Parameter & Computation Efficiency:

Vs. DenseNet: ResNet has a simpler connectivity pattern (one shortcut per block) and thus uses fewer parameters and less memory, yielding faster inference on edge devices.

Vs. EfficientNet: Although EfficientNet achieves high accuracy at low FLOPs, ResNet‑18’s uniform block structure is easier to implement, debug, and optimize across different frameworks.

1. Data‑Efficiency & Robustness

Vs. Vision Transformer (ViT): ResNet‑18 requires far less pretraining data and hyperparameter tuning. It converges reliably on moderate‑sized datasets like SkyView without extensive pretraining.

1. Ecosystem & Transfer Learning

ResNet‑18 has become a de facto standard “lightweight” CNN in PyTorch/TensorFlow, offering numerous pretrained weights, fine‑tuning recipes, and community support—accelerating development and reproducibility.

Our implementation loads torchvision.models.resnet18(pretrained=True) and replaces the final fully‑connected layer. Each ResNet block consists of two 3 × 3 convolutions, batch normalization, and a shortcut connection. We use global average pooling before the classifier to reduce parameters and enhance robustness.

Training Strategy:

1. Optimizer & Regularization: We use AdamW for decoupled weight decay, shown to improve generalization .
2. Learning‑Rate Scheduling: ReduceLROnPlateau reduces the lr when validation loss plateaus, preventing over‑stepping.
3. Early Stopping: If validation loss does not improve for 5 consecutive epochs, training stops early to avoid overfitting.

1. Densenet

In this project, we chose DenseNet121 as one of the representative deep learning methods. And in order to explore whether the attention mechanism can improve the performance of aerial image classification. We further constructed an extended model DenseNet121 + SE Block based on the channel attention mechanism.

1. DenseNet121 (basic model)

We use the pre-trained DenseNet121 in torchvision.models. And modify its output layer from nn.Linear(1024, 1000) to nn.Linear(1024, 15) to adapt to the 15-category classification task of the SkyView dataset. The image passes through four dense blocks and three transition layers in sequence. Then it is processed by global average pooling and finally input to the fully connected layer to complete the classification.

The training configuration of the model is as follows:

Loss function: CrossEntropyLoss;

Optimizer: Adam;

Learning rate schedule: StepLR;

Data enhancement strategy: We train the model under three settings: minimal , default , and extensive;

Evaluation indicators: Accuracy, Precision, Recall, and F1-score are evaluated in the validation set, and the confusion matrix is ​​plotted to analyze the confusion between categories.

Model selection and preservation: Compare the best model parameters during training and save the model with the best validation results.

2. DenseNet121 + SE Block (attention enhancement)

In order to further improve the model's responsiveness to key channels, we added Squeeze-and-Excitation Block (SE Block) after each dense block of DenseNet to form a DenseNet121 + SE model. SE Block is a lightweight channel attention mechanism. Its core idea is to readjust the importance of each channel by explicitly modeling the dependencies between channels, thereby enhancing the ability to focus on salient features [7].

Each SE Block contains the following structure:

Adaptive average pooling compresses the spatial dimension and extracts channel-level features;

Two-layer fully connected network (including BatchNorm and SiLU activation) generates attention weights;

Channel weights are normalized through Sigmoid activation;

The original feature map is multiplied by the attention weight to achieve feature recalibration.

We insert SE Block after dense block1, block2, block3 and block4 respectively, with input channels of 256, 512, 1024, and 1024 respectively, and the output features remain unchanged to ensure the continuity and reusability of the backbone structure.

3. Comparison with other models and selection motivation

The reason for choosing DenseNet121 is its excellent feature reuse capability and parameter efficiency. Compared with ResNet18, DenseNet's densely connected structure can more fully transmit low-level features, and still has good generalization ability at a smaller model scale, which is suitable for medium-sized datasets such as SkyView.

We further introduced SE Block to improve the model's ability to dynamically model semantic features of different regions and scales. Compared with Transformer models such as ViT, SE Block has more advantages in resource consumption and shows good structural adaptability in remote sensing images. Compared with EfficientNet, DenseNet's structure is more suitable for fine-grained insertion of attention modules, which facilitates structural expansion and performance comparison.

In summary, DenseNet121 and its attention-enhanced version are not only important representatives of deep methods in this project, but also key experimental models for analyzing the effect of attention mechanism on the performance improvement of aerial image classification.

1. Efficientnet

In this project, we use EfficientNet-B0 as one of the representatives of deep convolutional neural networks. We evaluate its performance in remote sensing image classification tasks. EfficientNet is a lightweight and efficient network architecture proposed by Tan and Le (2019). By proposing a "compound scaling" strategy, it achieves high classification accuracy while maintaining low computational cost [5].

1. Model selection motivation and remote sensing application

EfficientNet's compound scaling strategy simultaneously adjusts the depth, width, and input resolution of the network to achieve optimal performance in resource-constrained scenarios. In remote sensing image classification tasks, EfficientNet has also been widely used in recent years.

Yuan and Lu (2021) showed in their research on the NWPU-RESISC45 and AID datasets that EfficientNet achieves a better balance between accuracy and efficiency than traditional CNN architectures (such as VGG and ResNet), and is a strong candidate model for remote sensing scene recognition [8]. Zhong et al. (2021) also classified EfficientNet as an important representative of the next generation of lightweight CNN structures, suitable for classification tasks of high-resolution remote sensing images [9]. Therefore, in this project, we selected EfficientNet-B0 as one of the baselines of lightweight and efficient models, taking into account both computational efficiency and classification accuracy.

2. Model structure and training details

We loaded the pre-trained EfficientNet-B0 weights in torchvision.models and modified the original classification head nn.Linear(1280, 1000) to nn.Linear(1280, 15) to adapt to the classification task of 15 categories in the SkyView dataset. The model structure includes:

Backbone network: pre-trained EfficientNet-B0 (including MBConv and embedded SE Block);

Classification layer modification: adjust the linear layer output to 15 categories;

Input specification: the image is adjusted to 224×224 resolution to match the model default input.

The training configuration is as follows:

Optimizer: Adam;

Loss function: CrossEntropyLoss;

Learning rate scheduler: StepLR;

Data augmentation strategy: minimal, default, extensive;

Training and testing partition: 80% training, 20% testing;

Evaluation indicators: Accuracy, Precision, Recall, F1-score, and confusion matrix visualization.

The model was trained under three data augmentation strategies, and the model with the best performance on the validation set was finally evaluated on the test set.

3. Structural advantages and comparison between methods

Compared with other models, EfficientNet‑B0 provides better parameter efficiency and good training stability. Based on ResNet18, it uses MBConv modules and compound scaling strategies. This achieves a higher level of expression while maintaining a lower model complexity. Compared with DenseNet121, EfficientNet has a lighter structure and converges faster.

Compared to ViT, EfficientNet does not rely on large-scale pre-training data and is easier to train stably on medium-sized remote sensing datasets. Therefore, in this project we evaluate it as one of the representatives of lightweight deep learning models.

1. ViT

Reason for choosing ViT

Due to the excellent performance of the Vision Transformer (ViT) on ImageNet, I use the ViT to evaluate on the aerial image dataset.

（Vision Transformer (ViT) is a computer vision model based on the Transformer architecture, which was proposed by the Google team in 2020. It models image features by splitting the input image into multiple fixed-size patches and converting these patches into a one-dimensional vector sequence. This approach breaks the traditional convolutional neural network (CNN) approach that relies on local perception, allowing ViT to efficiently capture global dependencies.）

Global Information Capture: Since ViT relies on a self-attention mechanism, it can effectively capture long-distance dependencies throughout the image. For complex scenarios such as aerial landscapes, this helps to identify correlations between different areas, which is essential for analysis of land cover types, building distribution, etc.

#### Shortcoming:

ViT models often require large amounts of labeled data to be trained effectively, but this dataset is small. While ViT is good at capturing global information, it may not be as good at capturing local detail as traditional CNNs. In some cases, this can be detrimental to identifying fine-grained land cover types or small features (e.g., roads, trees). Besides, ViT requires significant computing resources to train, so in this task, I only trained the classification layer.

Model Optimization and Adaptation

To adapt to the aerial image classification task, the following improvements were made to the original ViT model:

Optimized Classification Head: Replaced the original simple linear classifier with a multi-layer classification head:

nn.Linear(768 → 512)

nn.BatchNorm1d(512)

nn.ReLU()

nn.Dropout(0.1)

nn.Linear(512 → 15 classes)

Regularization Strategies:

Applied a higher dropout rate (0.15) during training to prevent overfitting

Applied attention dropout of 0.1 on attention weights

Applied weight decay (1e-5) to further control model complexity

Class Imbalance Handling: Assigned higher loss weights to easily confused classes based on class confusion situations

Training Strategy

The following strategies were adopted during training to optimize model performance:

Optimizer Selection: Used AdamW optimizer with an initial learning rate of 0.0001

Dynamic Learning Rate: Applied ReduceLROnPlateau scheduler to reduce learning rate after validation loss plateau

Loss Function: Used weighted cross-entropy loss with label smoothing (0.1)

Batch Size: 64, balancing training stability and efficiency within GPU memory limitations

Training Epochs: Conducted 5 full epochs of training with early stopping strategy to avoid overfitting and conserve computing power.

1. Dataset Loading and Data Augmentation Methodology

Dataset Loading Mechanism

This research employs a customized AerialDataset class for the loading and processing of remote sensing landscape imagery. This class extends PyTorch's Dataset abstract class, enabling efficient management of aerial image data. In the dataset construction process, a stratified sampling strategy is implemented to partition the original dataset proportionally into training (60%), validation (20%), and test (20%) subsets, ensuring consistent class distribution across all subsets to prevent training bias.

For each sample, dynamic data loading is achieved through the \_\_getitem\_\_ method, which first reads the original image from the storage medium, then applies preset transformation and augmentation operations, and finally returns a tensorized image with its corresponding label, providing a standardized input format for model training.

Multi-level Data Augmentation Strategies

Data augmentation is a key technique for enhancing model generalization. This research designs four different intensity levels of data augmentation strategies to accommodate various training phases and model requirements:

Minimal Augmentation Strategy: Only basic transformation operations are applied, including size adjustment (224×224), random horizontal flipping, and normalization. This strategy is suitable for the initial fine-tuning phase of pre-trained models.

Default Augmentation Strategy: A medium-intensity combination of enhancements is adopted, including larger size adjustment (256×256), random cropping, random horizontal flipping, limited range random resized cropping (scale 0.6-1.0), low probability (0.3) Gaussian blur, random solarization, moderate angle (20°) random rotation, slight color jittering, and small-range affine transformation. This strategy introduces moderate variations while maintaining image semantic integrity.

Extensive Augmentation Strategy: High-intensity diversified enhancements are implemented, expanding on the default strategy with increased operation ranges and probabilities, such as greater size variation (scale 0.5-1.0), additional vertical flipping, larger angle (30°) random rotation, stronger color jittering, and random erasing operations. This strategy is applicable for model training with limited data availability.

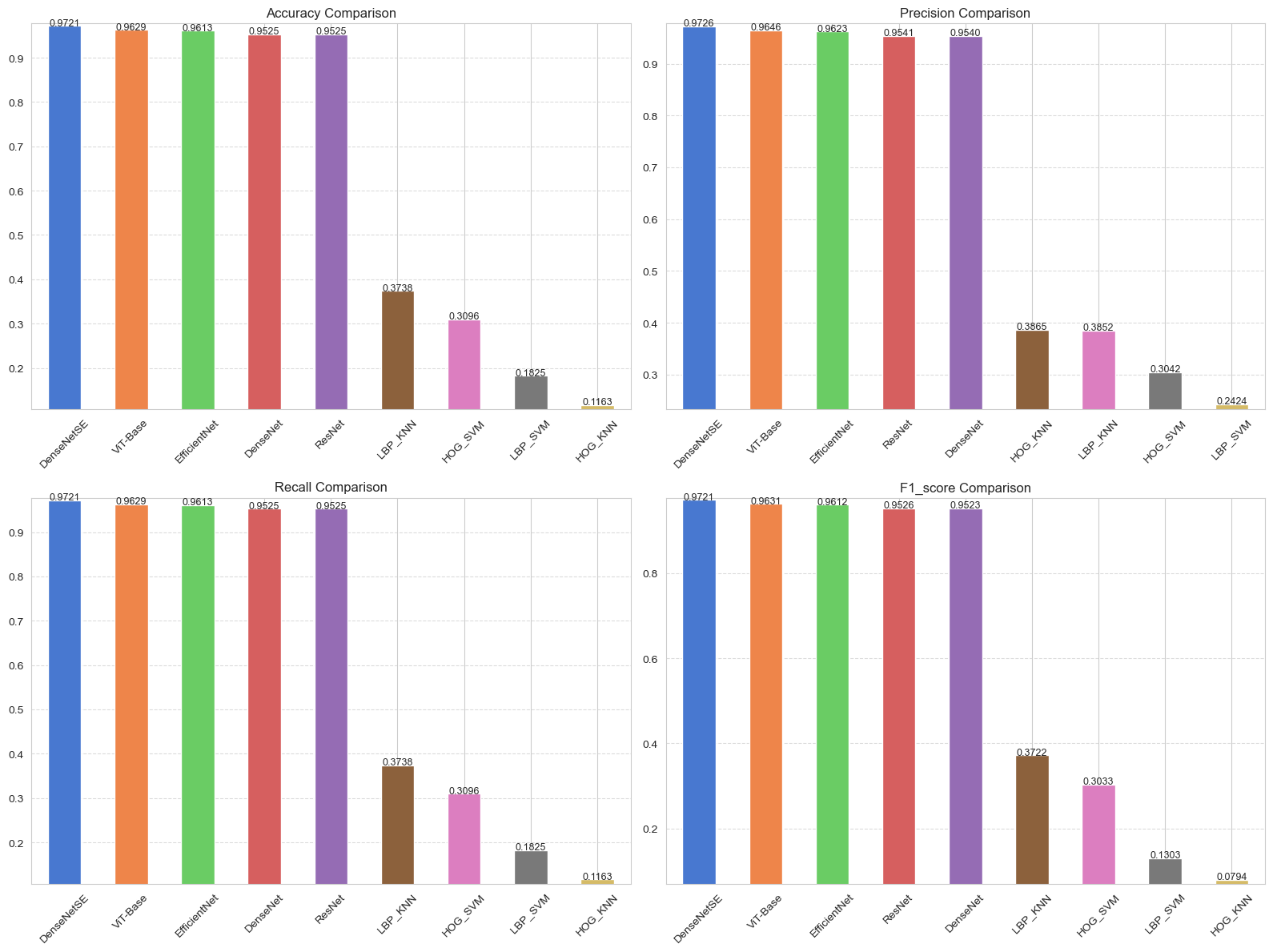
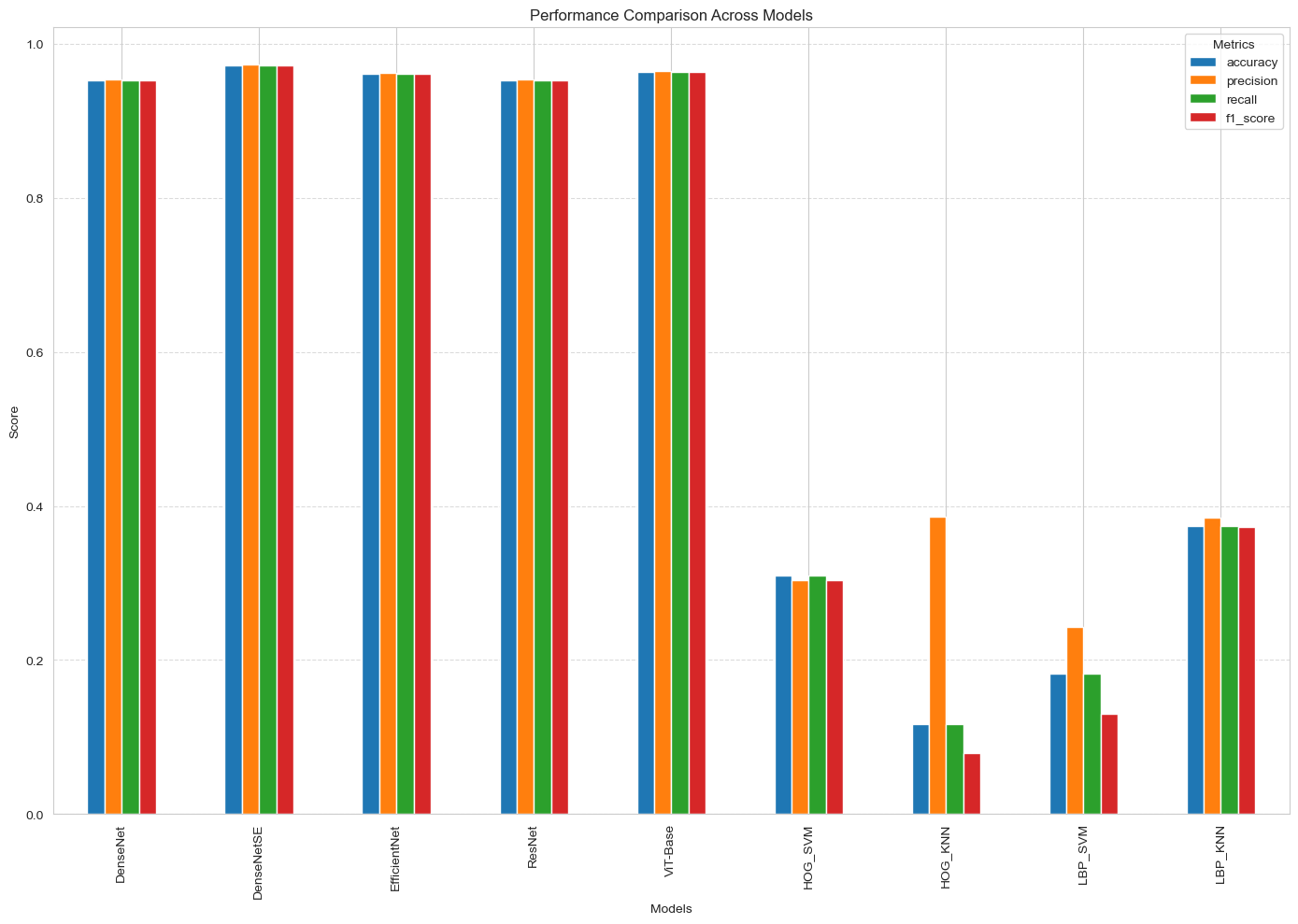
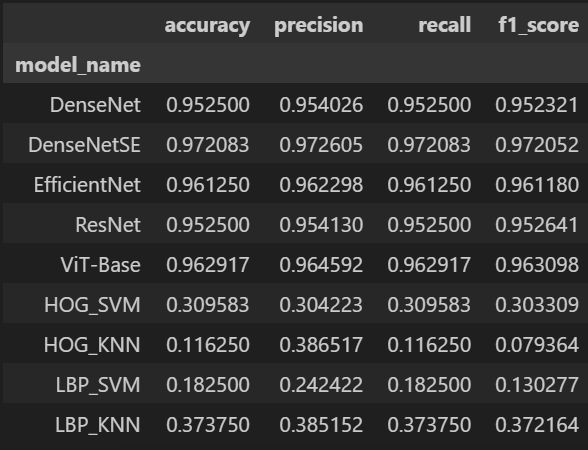
New Augmentation Strategy: An optimized enhancement combination is proposed suitable for this dataset, which integrates multiple advanced transformation techniques while maintaining consistent size (224×224), including perspective transformation, sharpness adjustment, automatic contrast, and histogram equalization, aimed at enhancing the model's recognition capability for complex scenes.

All strategies utilize ImageNet standardization parameters (mean [0.485, 0.456, 0.406], standard deviation [0.229, 0.224, 0.225]), ensuring compatibility with pre-trained model feature distributions and optimizing transfer learning effects.

Validation and test sets only undergo basic size adjustment and normalization processing, avoiding the introduction of unnecessary distribution shifts and ensuring the objectivity and reliability of evaluation results.

1. Experimental Results and Discussion

We systematically evaluated nine models on an independent test set of the SkyView dataset, covering traditional methods (HOG/LBP + SVM/KNN) and modern deep models (ResNet, DenseNet, EfficientNet, Vision Transformer). All models were trained on the same training and validation set splits. The test set did not participate in any training process to ensure the fairness of the evaluation. The prediction results are as follows:



1. Performance comparison between deep models and traditional methods

From the overall performance point of view, all deep learning models are significantly better than traditional methods in four evaluation indicators (Accuracy, Precision, Recall, F1-score). The best traditional method (LBP + KNN) F1-score is only 0.3722, while the weakest deep model (such as ResNet18 or DenseNet121) F1-score is still over 0.95. This gap shows that modern convolutional structures and deep feature learning mechanisms have obvious advantages in complex textures and fine-grained semantic modeling in remote sensing images.

Traditional features (HOG, LBP) have certain distinguishing capabilities in early image recognition tasks, but their expression capabilities are limited in remote sensing images due to complex background interference, scale changes, and blurred category boundaries. In addition, among traditional classifiers, SVM has stronger generalization capabilities than KNN and can build stable decision boundaries in high-dimensional space, while KNN is extremely sensitive to the distribution and scale changes of feature space, resulting in large fluctuations in overall performance.

1. Internal comparison analysis of traditional methods

Among the four traditional combinations, LBP + KNN performs best, with an F1-score of 0.3722. LBP features are highly robust to local texture information, especially in scenes with rich textures and blurred edges, where they are relatively stable. HOG, on the other hand, relies more on obvious edges and structures in images, and is easily affected by background complexity in remote sensing images. Especially when paired with KNN, its performance is significantly reduced, with an F1‑score of only 0.0794.

From the perspective of the classifier, SVM has better overall performance than KNN, especially under the high-dimensional sparse features extracted by HOG, SVM can effectively construct boundaries and avoid the dimensionality disaster encountered by KNN. Although HOG + SVM does not perform as well as LBP + KNN, it still has certain recognition capabilities in some scenes with obvious structural features (such as airports and highways). LBP + SVM performs slightly worse than LBP + KNN, probably because SVM is not flexible enough in distinguishing boundaries for low-level texture features.

In summary, traditional methods can only provide limited baseline performance in this task and are not competent for the semantic classification requirements of complex remote sensing images.

1. Internal comparison analysis of deep models

Among all deep models, DenseNet121 + SE performs best, achieving the highest F1-score (0.9721) on the test set, verifying the effectiveness of the channel attention mechanism (SE Block) in enhancing the model's response to significant feature channels, especially for fine-grained classification tasks.

As a representative of lightweight neural networks, EfficientNet‑B0 achieves a good balance between accuracy (F1-score 0.9612) and efficiency, demonstrating the advantages of compound scaling structures in resource-constrained environments. Its training and inference overhead is relatively low, making it one of the preferred models for actual deployment.

ViT‑Base is the only non-convolutional structure in this project. Its global self-attention mechanism effectively models long-distance spatial dependencies, with an F1-score of 0.9631, slightly lower than DenseNet‑SE, showing the potential of the Transformer architecture in remote sensing scene classification tasks. However, ViT is more sensitive to the amount of training data and preprocessing strategies, and has higher requirements for computing resources, which limits its promotion in small-scale remote sensing tasks.

ResNet18 and DenseNet121 have similar performances, with F1-scores exceeding 0.95, indicating that both have good feature extraction capabilities and generalization performance on medium-sized datasets, and are stable and reliable backbone structures in this task.

1. Conclusions

This project systematically compares the performance of traditional methods and multiple deep learning models in aerial image classification tasks. The experimental results show that deep models are significantly better than traditional methods in terms of accuracy, precision, recall and F1-score. DenseNet121 and its attention-enhanced version perform best in fine-grained classification, while EfficientNet-B0 achieves a good balance between accuracy and efficiency. Finally, ViT shows strong global modeling capabilities, proving the advantages of deep architecture in remote sensing image analysis.

However, traditional methods are obviously insufficient in feature expression and discrimination capabilities. First, high-complexity models such as ViT are sensitive to training resources. Secondly, the effect of SE Block also has a certain marginal weakening problem. At the same time, the SkyView dataset is still limited in geographical and spatiotemporal diversity, which may limit the generalization ability of the model in a wider range of practical scenarios.

Future work can consider introducing larger-scale and multi-modal remote sensing data, enhancing model interpretability, and exploring the possibility of deploying lightweight models in edge devices to further improve the practicality and promotion of the model.

1. Reference

【1】Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

【2】Ojala, T., Pietikäinen, M., & Harwood, D. (2002). Multiresolution Gray-Scale and Rotation

Invariant Texture Classification with Local Binary Patterns. *Pattern Recognition*.

【3】He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

【4】Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

【5】Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

【6】Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR)*.

【7】Hu, J., Shen, L., & Sun, G. (2018). *Squeeze-and-Excitation Networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141.

【8】Yuan, Z., & Lu, X. (2021). *Remote Sensing Scene Classification with EfficientNet.* Remote Sensing, 13(6), 1143.

【9】Zhong, Y., et al. (2021). *A review of deep learning methods for land use/land cover classification from remote sensing images.* ISPRS Journal of Photogrammetry and Remote Sensing, 179, 41–56.