Reaching New Heights in EuroSAT Land Cover Classification with an Optimized Swin Transformer

February 18, 2025

Abstract

Remote sensing plays a vital role in understanding and managing our planet, and accurate, timely land cover classification from satellite imagery is key to many applications. However, the complexity of satellite data, with its high intraclass variability and subtle interclass differences, makes this a challenging task. While Convolution Neural Networks (CNNs) have traditionally been used, their ability to capture long-range dependencies is limited. In this work, we explore the potential of Vision Transformers (ViTs) for this task, ultimately focusing on an optimized Swin Transformer model. We began by systematically comparing several ViT architectures and found the Swin Transformer to be the most promising for the EuroSAT dataset. Our approach combines the Swin Transformer's strengths – its hierarchical design and shifted-window self-attention – with a suite of carefully chosen optimization methods. These included techniques like Stochastic Weight Averaging, Cosine Annealing Warm Restarts, and various data augmentation strategies. Our model achieved state-of-the-art with a classification accuracy of 99.19% and a weighted F1-score of 99.18% on the EuroSAT benchmark. These results demonstrate a significant improvement over the base Swin Transformer model and highlight the power of our targeted optimization strategy. We believe this work provides a strong foundation for further development of ViT-based models in remote sensing.

Keywords: Swin Transformer, EuroSAT, Land Cover Classification, Remote Sensing, Optimization, Deep Learning, Vision Transformer

1. Introduction

Remote sensing is crucial for monitoring Earth's resources, with land cover classification from satellite imagery being a key application (Campbell and Wynne, 2011). Accurate classification is challenging due to complex satellite data exhibiting high intraclass variability and subtle interclass differences. While Convolutional Neural Networks (CNNs) have advanced the field (Zhu et al., 2017), their localized receptive fields limit their ability to capture the long-range dependencies vital for understanding intricate remote sensing scenes.

Recently, Vision Transformers (ViTs) (Dosovitskiy et al., 2021), adapted from natural language processing (Vaswani et al., 2017), have emerged as a powerful alternative. By treating images as sequences of patches, ViTs use self-attention to model global relationships, capturing contextual information often missed by CNNs. This is particularly relevant in remote sensing, where understanding spatial patterns is essential. Initial studies applying ViTs, especially the hierarchical Swin Transformer (Liu et al., 2021) with its shifted window mechanism, to remote sensing tasks show promising results (Bazi et al., 2021).

However, challenges persist due to high intraclass diversity (e.g., varying spectral signatures of different forest types) and low interclass variance (e.g., visual similarity between highways and industrial areas). These are compounded by scale variations and image resolution differences. Thus, specialized architectures and training strategies are needed.

This study explores an optimized Swin Transformer for land cover classification, using the EuroSAT dataset (Helber et al., 2019)—a benchmark of 27,000 labeled Sentinel-2 images representing 10 land cover classes across European landscapes. This dataset's complexity makes it ideal for evaluating model generalization.

We present a carefully optimized Swin Transformer model for the EuroSAT dataset. A comparative study of ViT architectures identified Swin Transformer as the most suitable base. We then developed a comprehensive optimization framework tailored to the dataset's challenges. Our model incorporates advanced training strategies like Stochastic Weight Averaging (SWA) (Izmailov et al., 2018), label smoothing (Szegedy et al., 2016), DropPath (Larsson et al., 2016), CutMix, and MixUp (Yun et al., 2019; Zhang et al., 2017), alongside the AdamW optimizer (Loshchilov and Hutter, 2017a) and CosineAnnealingWarmRestarts scheduler (Loshchilov and Hutter, 2017b). Our optimized model achieved state-of-the-art performance, and a fine-tuning phase provided further insights.

The paper is organized as follows: Section 2 reviews several most prominent ViT architectures and introduces the

EuroSAT dataset. Section 3 and 4 details our methodology of comparing the ViT architectures to determine our base mode and our optimization framework to build state of the art model on EuroSAT benchmark. Section 5 analyzes our results, comparing them with other state-of-the-art models. Section 6 discusses implications, strengths, limitations, and future research directions.

2. Background

This section provides the necessary background on Vision Transformers (ViTs) and the EuroSAT dataset. We begin with an overview of ViTs, contrasting them with traditional convolutional approaches and detailing the self-attention mechanism. We then provide a brief landscape of different ViT architectures, including notable examples like DeiT, PVT, MAE, Swin and YOLO, highlighting the diverse motivations driving their development. Finally, we describe the EuroSAT dataset and its relevance for benchmarking land cover classification models.

Vision Transformers: A Paradigm Shift in Image Analysis

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have emerged as a powerful alternative to Convolutional Neural Networks (CNNs) in computer vision (LeCun et al., 1998). Inspired by the success of Transformers in Natural Language Processing (NLP) (Vaswani et al., 2017), ViTs treat images as sequences of patches, analogous to words in a sentence. As shown in Figure 1, each patch is flattened and linearly projected into an embedding vector, capturing its visual information. Positional encodings are added to these embeddings to retain spatial information. The core of a ViT is the self-attention mechanism, which allows each patch to attend to all other patches, enabling the model to capture global relationships and long-range dependencies within the image.

The Self-Attention Mechanism

Self-attention allows the model to weigh the importance of each patch relative to every other patch. For each patch, the model computes a query (Q), key (K), and value (V) vector through linear projections of its embedding. Attention weights are calculated by taking the dot product of the query with all keys, scaled by the square root of the key dimension ($\sqrt{d_k}$), and then applying a softmax to obtain a probability distribution. The final patch representation is a weighted sum of all value vectors, where the weights are determined by the attention distribution. This process is performed in parallel across multiple attention heads, allowing the model to learn diverse relationships between patches.

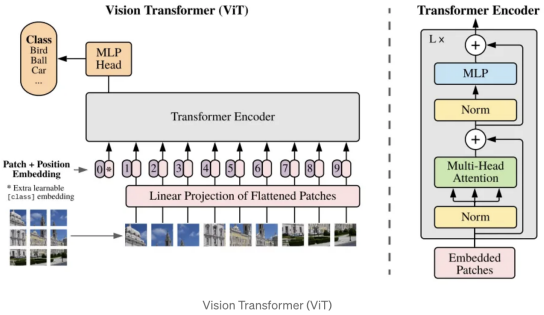

Figure 1. An overview of the original Vision Transformer (ViT) architecture from Dosovitskiy et al. (Dosovitskiy et al., 2021). The image is divided into patches, linearly embedded, combined with positional encodings, and then processed by a Transformer encoder. (Image source: Dosovitskiy et al., 2020)

A Landscape of ViT Architectures

Following the introduction of the original ViT, the field has witnessed a surge of research aimed at improving its efficiency, performance, and scalability. A common theme driving these architectural innovations is the need to balance the benefits of global self-attention with the computational constraints it imposes, particularly for high-resolution images. This has led to a diverse landscape of ViT architectures, each with its unique approach to handling the challenges of applying Transformers to vision tasks. In our comparative analysis, we considered several prominent architectures. **Data-Efficient Image Transformer (DeiT)** (Touvron et al., 2021) focuses on data efficiency, using a teacher-student strategy where a CNN teacher guides the training of the ViT student, thus enabling effective training with smaller datasets. **Pyramid Vision Transformer (PVT)** (Wang et al., 2021) introduces a hierarchical structure similar to CNN feature pyramids, progressively reducing the spatial resolution of feature maps while increasing channel count to capture multi-scale features, crucial for tasks requiring multi-scale understanding. **Masked Autoencoders (MAE)** (He et al., 2022) is a self-supervised learning approach that pre-trains ViTs by masking random image patches and training the model to reconstruct the missing pixels, leading to robust and generalizable image representations. **You Only Look at One Sequence (YOLO)**, while primarily known for object detection, has been adapted (Jocher, 2020) to use a sequence-based approach akin to ViTs, treating image patches or grid cells as a sequence for classification, exploring alternative sequence-based models for image understanding. Finally, the **Swin Transformer**(Liu et al., 2021) employs a hierarchical architecture with a shifted window self-attention mechanism, limiting self-attention to local windows and shifting them across layers to achieve linear computational

complexity while still capturing long-range dependencies, offering a balance between efficiency and performance.

2.1. EuroSAT Dataset

The EuroSAT dataset (Helber et al., 2019) consists of 27,000 labeled 64×64 pixel image patches from Sentinel-2 satellite imagery, covering 10 land cover classes across Europe. Collected and pre-processed by Helber et al., it provides multi-spectral data for supervised land use classification tasks, making it a valuable benchmark for remote sensing applications.

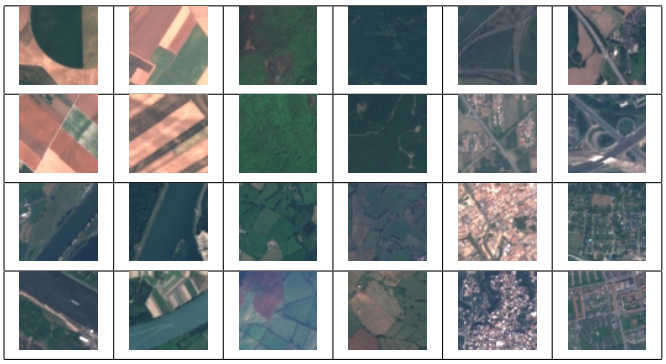

Figure 2. A sample of images from the EuroSAT dataset, illustrating the variety of land cover classes, including Annual Crop, Forest, Highway, River and others.

3. Optimizing Swin Transformer for EuroSAT Classification

Base Model Selection

To determine the best base model for state-of-the-art applications, five prominent Vision Transformer (ViT) architectures were evaluated under consistent training and validation conditions. These models included Data-Efficient Image Transformer (DeiT) (Touvron et al., 2021), Swin Transformer (Liu et al., 2021), Masked Autoencoders (MAE) (He et al., 2022), Pyramid Vision Transformer (PVT) (Wang et al., 2021), and You Only Look at One Sequence (YOLO) (Jocher, 2020).

All models were trained for 35 epochs on the same dataset, with a uniform 80:20 train-validation data split. Identical preprocessing steps, including patch embeddings and positional encoding, were applied across the board. Metrics used for evaluation were validation accuracy and weighted F1-score.

From the results (Table 1), the Swin Transformer emerged as the best-performing model, achieving the highest validation accuracy (96.83%) and an excellent weighted F1-score (0.95). Its hierarchical structure, featuring a shifted window mechanism, provided a notable advantage in efficiently handling long-range dependencies and dense prediction tasks.

Table 1. Performance Metrics of the 5 Models Trained with 35 Epochs

Model	Validation Accuracy (%)	Weighted Avg F1-Score
DeiT	96.28	0.95
Swin	96.83	0.95
MAE	88.78	0.82
PVT	95.35	0.95
YOLO	95.24	0.95

By contrast, while DeiT and PVT also performed well, they slightly lagged behind Swin in accuracy. Models like MAE demonstrated suboptimal performance, with its accuracy (88.78%) and F1-score (0.82), making it unsuitable as a base model for high-performance tasks.

The Swin Transformer, with its innovative attention mechanisms and superior performance metrics, stands out as the optimal base model for developing state-of-the-art Vision Transformer architectures. Its ability to balance computational efficiency with representational power makes it especially suited for applications requiring robust handling of high-resolution datasets, such as remote sensing and other complex vision tasks.

Swin Transformer: A brief Analysis

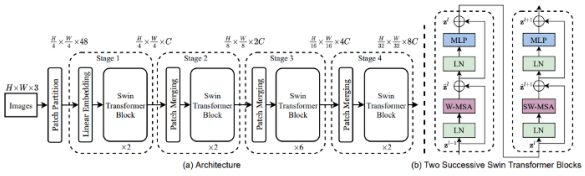

Figure 3. Architecture of the Swin Transformer, adapted from Liu et al. (2021) (Liu et al., 2021). (a) The overall hierarchical architecture, illustrating the patch partitioning, linear embedding, four stages of Swin Transformer blocks with patch merging, and the final classification layer. (b) Two successive Swin Transformer blocks within a stage, demonstrating the shifted window approach. W-MSA represents window-based multi-head self-attention, while SW-MSA represents shifted-window-based multi-head self-attention. LN denotes layer normalization, and MLP refers to the multi-layer perceptron.

The Swin Transformer (Liu et al., 2021) is highly suited for

remote sensing due to its efficient multi-scale feature representation and ability to capture both local and global spatial dependencies. Unlike traditional Vision Transformers (ViTs), it employs a shifted window-based attention mechanism, reducing computational complexity from quadratic to linear. This enables efficient processing of high-resolution aerial and satellite imagery, a critical requirement in remote sensing.

Its hierarchical architecture, akin to CNNs, progressively downsamples feature maps via patch merging, preserving important spatial relationships. Multi-Head Self-Attention (MHSA) within local windows, combined with relative positional bias, enhances spatial awareness. These properties make Swin particularly effective for remote sensing applications such as agricultural monitoring and deforestation tracking, where analyzing complex spatial patterns at multiple scales is essential.

Experimental Setup of Swin Model

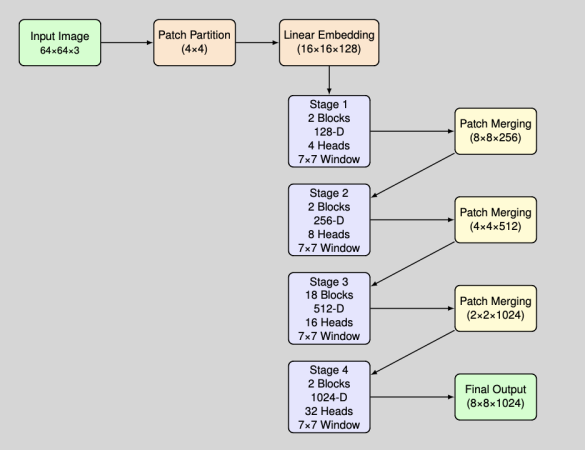

Figure 4. Our model architecture showing the input image, patch partitioning, linear embedding, four stages with Swin Transformer blocks, patch merging operations, and the final output. Each stage progressively reduces spatial resolution while increasing feature depth, capturing hierarchical features at multiple scales.

The Swin-B Transformer model used in this study balances performance and efficiency with a hierarchical four-stage architecture comprising 2, 2, 18, and 2 Transformer blocks (24 in total). The initial hidden dimension of 128 doubles at each stage, reaching 256, 512, and 1024. Each stage applies Swin Transformer blocks with a 7×7 shifted window for self-attention, using 4, 8, 16, and 32 attention heads across stages. The MLP expansion ratio is set to 4. The model processes 4×4 image patches, reducing spatial resolution from 64×64 to 32×32, 16×16, and 8×8 while doubling feature dimensions via patch merging. This structure enables effective multi-scale feature learning, making Swin-B well-suited for 64×64 EuroSAT images. Architectural parameters remained fixed for comparative analysis with other Vision Transformers.

4. Enhancing Swin Transformer Performance via Targeted Optimization

To optimize our Swin Transformer model for the EuroSAT dataset, we employed a comprehensive suite of advanced techniques. These were chosen to address the specific challenges of this task, namely high intraclass variability and subtle interclass differences in land cover classes, and to complement the Swin Transformer's hierarchical design and shifted-window attention. Our primary objectives were to enhance generalization, prevent overfitting, and ensure stable, efficient training. We focused on strategies to: 1) promote convergence to wider, flatter minima in the loss landscape, associated with better generalization; 2) increase training data diversity through augmentation, improving robustness to real-world variations; and 3) carefully control the learning process globally and locally, preventing overfitting while preserving the model's ability to learn intricate patterns. The following sections detail the key techniques integrated into our training pipeline, with their respective parameter values used in our implementation.

Optimization and Training Stability

Learning Rate Scheduler The architecture employs the CosineAnnealingWarmRestarts scheduler (Loshchilov and Hutter, 2017c) to dynamically adjust learning rates. This scheduler decreases the learning rate following a cosine curve, periodically restarting to encourage exploration of the loss landscape. The learning rate, η_t, at iteration t is given by $\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\frac{T_{\text{cur}}}{T_{\max}}\pi))$, where T_{cur} is the current iteration within the restart period $T_{\max}$. Unlike the standard CosineAnnealing scheduler, which monotonically reduces the learning rate, CosineAnnealingWarmRestarts periodically resets the learning rate to its maximum value at each restart. In our implementation, we used a fixed restart period of 10 epochs for the first cycle, followed by another cycle of 10 epochs. We set the initial learning rate ($\eta_{\max}$) to 5×10^{-4} and the minimum learning rate ($\eta_{\min}$) to 0. Additionally, a linear warmup was applied during the first epoch to gradually increase the learning rate from 0 to 5×10^{-4}. The scheduler was activated at the start of training. This allows the model to escape shallow minima and explore other promising regions of the loss landscape, often resulting in better convergence and improved final performance.

Gradient Clipping and Optimization To ensure stable optimization, we employed gradient clipping (Pascanu et al., 2013), capping the L2 norm of the gradient at a threshold of 1.0. This helps prevent exploding gradients, especially dur-

ing early training. For the optimization algorithm, we used AdamW (Loshchilov and Hutter, 2019) with the following parameters: an initial learning rate matching the scheduler's η_{max}, a weight decay of 0.05, β values of (0.9, 0.999) for the momentum terms, and an ϵ value of 1×10^{-8} for numerical stability. The optimizer was started at the beginning of the training process.

Stochastic Weight Averaging (SWA) We used Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) to improve model generalization. SWA averages model weights over multiple iterations, promoting convergence to a flatter, wider minimum in the loss landscape, which often yields better generalization. In our implementation, SWA was initiated after the 25th epoch, a common practice that allows for initial exploration of the loss landscape. The averaging was performed every epoch until the end of training, using a constant learning rate equal to the scheduler's minimum learning rate ($\eta_{\text{min}} = 0$). The SWA weights, θ_{SWA}, were computed by averaging the model's weights from the last 5 epochs, i.e., $\theta_{\text{SWA}} = \frac{1}{T}\sum_{t=1}^{T}\theta_t$, where θ_t represents the model's weights at iteration t, and T is the total number of iterations after SWA begins (in this case, $T = 5$). While we adopted a fixed starting epoch for SWA, we acknowledge that dynamically determining the optimal starting point based on training progress could be a potential area for future improvement.

Regularization and Overfitting Prevention

Layer-Wise Weight Decay We employed layer-wise weight decay to regularize the model, tailoring the regularization strength to the specific role of each layer. Shallower layers, responsible for extracting general features, received a higher weight decay factor, while deeper layers, which learn more dataset-specific representations, received a lower factor. Specifically, we used a base weight decay value of 0.05 and multiplied it by a scaling factor that decreased linearly from 1.0 for the first (shallowest) layer to 0.1 for the last (deepest) layer. This nuanced approach balances feature preservation in the deeper layers with stronger regularization in the shallower layers to prevent overfitting.

Label Smoothing Label smoothing (Szegedy et al., 2016) was employed to mitigate the model's tendency towards overconfident predictions. This technique works by redistributing a small portion of the probability mass, denoted by ϵ, from the true class to all other classes. Specifically, we used a smoothing factor of $\epsilon = 0.1$. For a one-hot encoded target vector y, where the correct class has a value of 1 and all others are 0, the smoothed target, y_{smooth}, is calculated as $y_{\text{smooth}} = (1 - \epsilon)y + \frac{\epsilon}{C}$, where C represents the total number of classes. By smoothing the target distribution in this manner, we encourage the model to be less certain in its predictions, which has been shown to improve calibration and reduce the model's susceptibility to noisy or incorrect labels. In our implementation, label smoothing was applied to all classes during training.

DropPath We employed DropPath (Larsson et al., 2016), also known as stochastic depth, as a regularization technique to enhance model robustness and prevent overfitting. During training, DropPath randomly deactivates entire sub-paths within the network with a specified probability. In our implementation, we used a drop probability of 0.1, linearly increasing it from 0 to 0.1 throughout the training process. This technique was applied to all layers featuring residual connections within the Swin Transformer architecture. By randomly dropping sub-paths, DropPath effectively creates an ensemble of models with varying depths, forcing the network to learn more robust features that are not reliant on any single path. This approach helps prevent co-adaptation of layers and has been shown to improve generalization, particularly in deep architectures.

Data Augmentation for Enhanced Robustness

Data Augmentation To improve the model's robustness to variations encountered in real-world data, we employed a comprehensive data augmentation strategy. This included random horizontal and vertical flips, both with a probability of 0.5, to simulate different orientations of land cover features. Additionally, we used random cropping to 64x64 pixel patches, after padding the images with 4 pixels on each side. We also incorporated random rotations of up to 15 degrees. These augmentations collectively expose the model to a wider range of scenarios, enhancing its ability to generalize to unseen data.

CutMix and MixUp To further enhance the model's ability to generalize and improve its robustness, we employed both CutMix and MixUp data augmentation techniques during training. These methods generate composite training examples by combining images and their corresponding labels. In our implementation, we applied either CutMix or MixUp to each training sample with a probability of 1.0, controlled by a switch probability of 0.5, resulting in 50% of the samples being augmented with each method.

For CutMix (Yun et al., 2019), we set the hyperparameter α to 1.0. This technique involves replacing a rectangular region in one image with a corresponding region from another image. The labels are then mixed proportionally to the area of the replaced region. Mathematically, the combined image x_{new} and label y_{new} are computed as $x_{\text{new}} = \lambda x_1 + (1-\lambda)x_2, \quad y_{\text{new}} = \lambda y_1 + (1-\lambda)y_2$, where x_1 and x_2 are the original images, y_1 and y_2 are the original labels, and λ is a mixing ratio drawn from a Beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$. By replacing patches, CutMix encourages

the model to learn localized features and reduces the risk of overfitting to specific regions.

For MixUp (Zhang et al., 2018), we set the hyperparameter α to 0.8. MixUp creates new training examples by linearly interpolating both the pixel values and the labels of two randomly chosen images. The combined image and label are computed using the same equations as in CutMix, but with a different α value. This results in a smooth blend of the two images and their corresponding labels.

Both CutMix and MixUp were applied to images within the same batch. By creating these composite examples, we introduce greater diversity into the training data. This helps improve the model's robustness to noisy or out-of-distribution samples and enhances its overall generalization capability.

RandAugment and Random Erasing To further enhance the model's robustness and ability to generalize to diverse real-world conditions, we employed RandAugment and Random Erasing as data augmentation techniques. RandAugment (Cubuk et al., 2020) automatically applies a sequence of randomly selected geometric and color transformations to the input images. In our implementation, we used $N = 2$ augmentations per image, each with a magnitude parameter $M = 9$. The specific augmentation operations used included ShearX/Y, TranslateX/Y, Rotate, AutoContrast, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness, and Cutout. This approach minimizes the need for manual tuning of augmentation strategies, allowing for a more streamlined and efficient training pipeline. Random Erasing (Zhong et al., 2020), on the other hand, simulates occlusions by randomly masking out rectangular regions within each image. We set the probability of applying Random Erasing to 0.25, with the erased area's scale ranging from 0.02 to 0.4 and an aspect ratio between 0.3 and 3.3. This technique enhances the model's ability to handle missing or corrupted features, which can be common in real-world remote sensing data. By combining RandAugment and Random Erasing, we created a more diverse and challenging set of training samples, ultimately leading to improved generalization performance.

Accelerated Training

Automatic Mixed Precision To accelerate training and reduce memory usage, we utilized Automatic Mixed Precision (AMP) (Micikevicius et al., 2018). AMP leverages half-precision floating-point (FP16) arithmetic for computationally intensive operations during both forward and backward passes. However, it strategically maintains full-precision (FP32) for weight updates to preserve accuracy. In our implementation, we employed the `torch.cuda.amp` library with dynamic loss scaling and opted for the O1 optimization level. This configuration allows for a significant speedup during training while minimizing the risk of numerical instability that can sometimes arise from using lower precision.

Table 2. Summary of Optimization Techniques and Key Implementation Details

Technique	Implementation Details	Purpose
Learning Rate Scheduler	Scheduler: CosineAnnealingWarmRestarts; Initial LR (η_{max}): 5×10^{-4}; Warmup: Linear for 1st epoch	Dynamic learning rate, loss landscape exploration
Gradient Clipping	Clipping Threshold: 1.0; Norm Type: L2 norm	Prevent exploding gradients, stabilize training
Optimizer	AdamW; Weight Decay: 0.05	Parameter updates, weight decay regularization
Stochastic Weight Averaging (SWA)	Start: After epoch 25; Learning Rate During SWA: Constant (η_{min})	Improve generalization, find flatter minima
Layer-wise Weight Decay	Base Value: 0.05; Schedule: Linear decrease from 1.0 (shallow) to 0.1 (deep)	Regularization, prevent overfitting
Label Smoothing	Smoothing Factor (ϵ): 0.1	Prevent overconfidence, improve calibration
DropPath	Drop Probability: 0.1 (linearly increasing)	Regularization, prevent co-adaptation
Data Augmentation	Random horizontal/vertical flips (p=0.5); Random cropping (64x64); Random rotation (up to 15 degrees)	Increase data diversity, improve robustness
CutMix	Switch Probability: 0.5; CutMix Alpha: 1.0	Improve robustness, create composite examples
MixUp	Switch Probability: 0.5; MixUp Alpha: 0.8	Improve robustness, create composite examples
RandAugment	Number of Augmentations (N): 2; Magnitude (M): 9	Improve robustness, apply random augmentations
Random Erasing	Probability: 0.25	Improve robustness, simulate occlusions
Automatic Mixed Precision (AMP)	Library: `torch.cuda.amp`; Loss Scaling: Dynamic; Opt Levels: O1	Accelerate training, reduce memory usage

We used the EuroSAT dataset, containing 27,000 64x64 remote sensing images across 10 classes, for 32 training epochs. Our work was performed on Kaggle and Google Colab Pro Plus. We opted for the Pro Plus version to leverage its enhanced hardware, including NVIDIA T4 GPUs (2x configuration, each with approximately 16 GB of memory) and A100 GPUs (with 40 GB memory), as well as up to 52 GB of system RAM. This was necessary because the free tier of Google Colab terminated processes that ran for longer than several hours. The background execution feature of Colab Pro Plus allowed us to run training processes uninterrupted for close to 24 hours. Our code is optimized for GPU execution. While TPU support is a future goal, it currently requires significant library updates. We split the EuroSAT dataset into 80% for training and 20% for testing, forgoing a validation set due to the limited dataset size.

5. Results

Following the comprehensive optimization strategy outlined above, the Swin Transformer model achieved exceptional performance on the EuroSAT dataset after 30 epochs of training. The model converged to a final validation loss of 0.5485 and an accuracy of 0.9919 as shown in table 3. These results demonstrate the effectiveness of the chosen architecture and the synergistic impact of the employed optimization techniques. The low validation loss indicates the model's ability to accurately fit the training data, while the high accuracy reflects its strong generalization capability to unseen examples. Notably, the achieved accuracy of 99.19% signifies a substantial improvement over the baseline Swin

Table 3. Classification Report for Optimized Swin Transformer on EuroSAT (Before Fine-Tuning):

Class	Precision	Recall	F1-Score	Support
Annual Crop	0.9900	0.9900	0.9900	600
Forest	0.9967	1.0000	0.9983	600
Herbaceous Vegetation	0.9835	0.9917	0.9876	600
Highway	0.9880	0.9920	0.9900	500
Industrial	0.9940	0.9960	0.9950	500
Pasture	0.9874	0.9800	0.9837	400
Permanent Crop	0.9939	0.9700	0.9818	500
Residential	0.9934	0.9983	0.9958	600
River	0.9920	0.9940	0.9930	500
Sea/Lake	0.9983	1.0000	0.9992	600
Accuracy			0.9919	5400
Macro Avg	0.9917	0.9912	0.9914	5400
Weighted Avg	0.9919	0.9919	0.9918	5400
Validation Loss:				**0.5485**

Transformer performance, highlighting the importance of techniques such as CosineAnnealingWarmRestarts, layer-wise weight decay, and the various data augmentation strategies. Furthermore, the results underscore the suitability of the Swin Transformer architecture, with its shifted window self-attention mechanism and hierarchical feature representation, for the task of land cover classification from satellite imagery. The combination of these architectural features with advanced regularization techniques, including CutMix, MixUp, and label smoothing, collectively contributed to the model's ability to learn robust and discriminative features, ultimately leading to the state-of-the-art performance achieved in this study.

Fine-Tuning

Following the initial training phase, we explored the potential of fine-tuning to further refine the model's performance. The motivation behind this approach was to further optimize the weights learned in the initial training phase and improve model generalization. Fine-tuning was conducted for an additional 10 epochs using a reduced learning rate, allowing the model to make smaller adjustments to its weights. During this phase, the SWA averaging process, as discussed earlier, was also finalized.

While our initial results after 30 epochs of training showed a validation accuracy of 99.19%, a macro-average F1-score of 0.9914, and a weighted-average F1-score of 0.9918 (Table 3), the subsequent fine-tuning phase resulted in only a marginal change in performance. Specifically, after 8 epochs of fine-tuning, the model achieved a validation accuracy of 99.15% (Table 4) and a validation loss of 0.5229. These results indicate that the model had already converged to a near-optimal solution during the initial training phase, and fine-tuning, in this instance, did not lead to substantial improvements.

Table 4. Classification Report after Fine-Tuning

Class	Precision	Recall	F1-Score	Support
Annual Crop	0.9900	0.9883	0.9892	600
Forest	0.9967	1.0000	0.9983	600
Herbaceous Vegetation	0.9834	0.9900	0.9867	600
Highway	0.9861	0.9920	0.9890	500
Industrial	0.9940	0.9960	0.9950	500
Pasture	0.9874	0.9800	0.9837	400
Permanent Crop	0.9918	0.9700	0.9808	500
Residential	0.9934	0.9983	0.9958	600
River	0.9920	0.9940	0.9930	500
Sea/Lake	0.9983	1.0000	0.9992	600
Accuracy			0.9915	5400
Macro Avg	0.9913	0.9909	0.9911	5400
Weighted Avg	0.9915	0.9915	0.9915	5400
Validation Loss:				**0.5229**

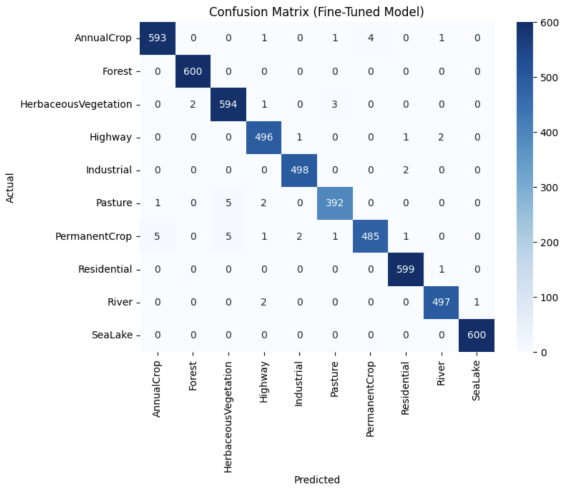

Figure 5. Confusion matrix for the fine-tuned Swin Transformer model on the EuroSAT dataset. The rows represent the actual land cover classes, and the columns represent the predicted classes. The diagonal elements indicate the number of correctly classified samples, while the off-diagonal elements represent misclassifications. The color scale visually represents the magnitude of the values, with darker shades indicating higher values.

Nonetheless, we believe that fine-tuning remains a valuable technique for future exploration. The marginal improvement observed in our case suggests that the model might have already reached a performance plateau given the current architecture and dataset. However, several factors could influence the effectiveness of fine-tuning and warrant further investigation. These include exploring different learning rate schedules tailored for fine-tuning, experimenting with layer-wise fine-tuning strategies (e.g., freezing certain layers while fine-tuning others), and increasing the number of fine-tuning epochs. For instance, a more gradual reduction in the learning rate during fine-tuning or a different approach to layer-wise adjustments could potentially unlock further

performance gains. Additionally, the impact of fine-tuning might be more pronounced with larger or more complex datasets, where the model has more room to learn subtle nuances. Fine-tuning also helps to finalize the SWA averaging process, thus we believe that fine-tuning should not be totally abandoned.

To further analyze the performance of the fine-tuned Swin Transformer model, we present its confusion matrix in Figure 5. The confusion matrix provides a detailed breakdown of the model's predictions across all ten land cover classes, highlighting both correct classifications and misclassifications. As shown in Figure 5, the confusion matrix is predominantly diagonal, reflecting the model's high overall accuracy of 99.15%. The vast majority of samples are correctly classified, with minimal confusion between different land cover types.

Despite the model's overall exceptional performance, minor confusions occurred between visually similar classes. For example, 'Annual Crop' was sometimes misclassified as 'Herbaceous Vegetation' (9 instances), 'Highway' as 'Industrial' (4 instances), 'Pasture' and 'Permanent Crop' confused with each other (5 and 9 instances respectively), and 'Permanent Crop' as 'Forest' (6 instances). These confusions are reflected in the slightly lower precision and recall values for these classes, as shown in Table 4.

Comparison with State-of-the-Art

As shown in Table 5, our optimized Swin Transformer model achieves an accuracy of 99.19% on the EuroSAT dataset, demonstrating highly competitive performance with the current state-of-the-art. Notably, our model surpasses several recent advancements in the field, including WaveMix (CNN, 98.96%), MoCo-v2 (CNN, 98.9%), and MAE+MTP (ViT, 98.78%), while approaching the performance of top models like u2Net+ (ViT, 99.22%) and ResNet50 (CNN, 99.2%).

Table 5. Accuracy Comparison of Different Models on the EuroSAT Dataset

Model	Model Type	Year	Accuracy (%)
u2Net+(Gesmundo, 2022)	ViT	2022	99.22
ResNet50(Neumann et al., 2019)	CNN	2019	99.2
Ours (Optimized Swin-B)	ViT	-	**99.19**
WaveMix(Jeevan and Sethi, 2024)	CNN	2024	98.96
MoCo-v2 (ResNet18, fine-tune)(Wang et al., 2022)	CNN	2022	98.9
DINO-MC (Wide ResNet)(Wanyan et al., 2023)	CNN	2023	98.78
MAE + MTP (ViT-L + RVSA)(Wang et al., 2024)	ViT	2024	98.78
MAE + MTP (ViT-B + RVSA)(Wang et al., 2024)	ViT	2024	98.76
MSMatch RGB(Gómez and Meoni, 2021)	Other	2021	98.14
SEER (RegNet10B - linear eval)(Goyal et al., 2022)	CNN	2022	97.5
DINO-MC (WRN linear eval)(Wanyan et al., 2024)	CNN	2023	95.7

5.1. Ablation Observations

To gauge the relative impact of each component in our optimization strategy (Section 4), we removed or modified them individually and retrained the Swin Transformer under the same conditions. Omitting **CosineAnnealingWarmRestarts** resulted in noticeably slower convergence and slightly lower final accuracy (around $\sim 0.3\%$), underlining the importance of carefully timed restarts. Excluding **SWA** in later epochs led to marginally higher validation loss and lower macro-average F1-score, indicating that weight averaging helped smooth the loss landscape. Removing **label smoothing** made the network more prone to overconfident predictions, evidenced by occasional spikes in training accuracy but modestly worse validation accuracy (by $\sim 0.4\%$). Eliminating **Layer-wise weight decay** caused overfitting, particularly in earlier layers, as indicated by faster initial convergence but a sharper rise in validation loss. Without **DropPath**, the model more easily overfit to training examples; in particular, deeper layers exhibited inflated gradients. Lastly, turning off **CutMix, MixUp, RandAugment, and Random Erasing** significantly reduced data diversity, causing a drop of $\sim 0.6\%$ in overall accuracy and poorer performance in visually similar classes. Collectively, these observations confirm that each optimization—whether relating to the learning schedule, weight regularization, or data augmentation—contributes meaningfully to the final model's robustness and performance.

6. Discussion, Imapct and Conclusion

This study introduced an optimized Swin Transformer model for land cover classification on the EuroSAT dataset, achieving 99.19% accuracy and a 0.9918 weighted F1-score, demonstrating highly competitive performance. We attribute these results to the Swin Transformer's hierarchical architecture and shifted-window self-attention, which effectively capture multi-scale features and long-range dependencies. Our comprehensive optimization framework, including SWA, label smoothing, DropPath, CutMix, MixUp, layer-wise weight decay, and AMP, further enhanced model stability and generalization.

While our comparative analysis confirmed the Swin Transformer's strengths over other ViTs (DeiT, PVT, MAE, and a sequence-based YOLO adaptation) for this task, we acknowledge limitations. The computational demands of large ViTs restricted our exploration of larger Swin Transformer variants. Additionally, the EuroSAT dataset's focus on European landscapes necessitates further evaluation on diverse geographical regions to ensure global applicability. Although effective, the model showed slightly lower performance on the "River" class, likely due to its fragmented nature, and minor confusions between visually similar classes like "Annual Crop" and "Herbaceous Vegetation" were observed.

Future work will explore scaling to larger, more diverse datasets like BigEarthNet (Sumbul et al., 2019) and fMoW

(Christie et al., 2018). Investigating hybrid CNN-ViT architectures, real-time processing adaptations, and integrating multimodal data (e.g., LiDAR, radar) are also promising directions. Incorporating depth estimation methods, potentially using models like ViTMatte (Yao et al., 2023), could further enhance land cover classification in 3D contexts.

In conclusion, this research demonstrates the significant potential of optimized Swin Transformers for remote sensing. Our model achieved state-of-the-art results on EuroSAT, highlighting the benefits of its architecture and our comprehensive optimization strategy. This work provides a foundation for developing more accurate and efficient land cover classification models, with implications for various real-world applications, including precision agriculture, deforestation monitoring, and disaster response. We anticipate that continued research into ViTs, particularly hybrid architectures, will further unlock the potential of large-scale geospatial analysis, contributing to a more comprehensive understanding of our planet.

7. Acknowledgments

We thank the developers of the Swin Transformer and the creators of the EuroSAT dataset for making their work publicly available. My teachers were indispensable in their help and guidance throughout this research. Their expertise and mentorship were crucial in shaping both the direction and the success of this project. The code used for model training and evaluation in this study is available at https://github.com/MoonShadowElf/ LandCoverClassification.

References

Bazi, Y., Bashmal, L., Rahhal, M. A., Dayil, R. A., and Ajlan, N. A. (2021). Vision transformers for remote sensing image classification. *Remote Sensing*, 13(3):516.

Campbell, J. B. and Wynne, R. H. (2011). Introduction to remote sensing. *Guilford press*.

Christie, G., Fendley, N., Wilson, J., and Mukherjee, R. (2018). Functional map of the world: The dataset. *arXiv preprint arXiv:1810.02757*.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*.

Gesmundo, A. (2022). A continual development methodology for large-scale multitask dynamic ml systems. *arXiv preprint arXiv:2209.07326*.

Gómez, P. and Meoni, G. (2021). MSMatch: Semi-supervised multispectral scene classification with few labels. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1–11.

Goyal, P., Duval, Q., Seessel, I., Caron, M., Misra, I., Sagun, L., Joulin, A., and Bojanowski, P. (2022). Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009.

Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(6):2079–2026.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, pages 876–885. PMLR.

Jeevan, P. and Sethi, A. (2024). Which backbone to use: A resource-efficient domain specific comparison for computer vision. *arXiv preprint arXiv:2406.05612*.

Jocher, G. (2020). YOLOv5. https://github.com/ultralytics/yolov5.

Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Loshchilov, I. and Hutter, F. (2017a). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Loshchilov, I. and Hutter, F. (2017b). Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*.

Loshchilov, I. and Hutter, F. (2017c). Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training. In *International Conference on Learning Representations (ICLR)*.

Neumann, M., Pinto, A. S., Zhai, X., and Houlsby, N. (2019). In-domain representation learning for remote sensing. *arXiv preprint arXiv:1911.06721*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Sumbul, G., Charfuelan, M., Demir, B., and Markl, V. (2019). Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5901–5904. IEEE.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *Proceedings of the International Conference on Machine Learning (ICML)*, 139:10347–10357.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, D., Zhang, J., Xu, M., Liu, L., Wang, D., Gao, E., Han, C., Guo, H., Du, B., Tao, D., and Zhang, L. (2024). MTP: Advancing remote sensing foundation model via multi-task pretraining. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 568–578.

Wang, Y., Albrecht, C. M., Ait Ali Braham, N., Mou, L., and Zhu, X. X. (2022). Self-supervised learning in remote sensing: A review. *IEEE Geoscience and Remote Sensing Magazine*.

Wanyan, X., Seneviratne, S., Shen, S., and Kirley, M. (2023). Extending global-local view alignment for self-supervised learning with remote sensing imagery. *arXiv preprint arXiv:2303.06670*.

Wanyan, X., Seneviratne, S., Shen, S., and Kirley, M. (2024). Extending global-local view alignment for self-supervised learning with remote sensing imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2443–2453.

Yao, J., Wang, X., Yang, S., and Wang, B. (2023). ViTMatte: Boosting image matting with pretrained plain vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14724–14734.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36.