NJIT – Department of Computer Science

CS 634 – Deep Learning

Fall 2025

Final Research Project Report

Swin Transformer for Fine-Grained Image Classification: A Comprehensive Evaluation on Oxford-IIIT Pet Dataset

Student Name: Robert Jean Pierre

UCID: RJ447

Video Presentation: YouTube or Drive Link Here

Project Repository / Folder: GitHub or Google Drive

Abstract

This study presents a comprehensive evaluation of the Swin Transformer architecture for fine-grained image classification, replicating and extending the original research findings to a resource-constrained, small-dataset scenario. While the original Swin Transformer paper demonstrated superior performance on large-scale datasets (ImageNet-1K, COCO, ADE20K), this research investigates whether these architectural advantages translate to practical deployment conditions using the Oxford-IIIT Pet Dataset (37 classes, ~7,400 images). We systematically compare Swin Transformer variants (Swin-T, Swin-S, Swin-B) against established baselines including RegNet CNNs, EfficientNet models (B3-B7), and Vision Transformers (ViT-B/16, ViT-L/16) under controlled experimental conditions.

Our results validate the core claims of the original Swin research while revealing important insights about architectural scalability to small datasets. Swin Transformers achieve overwhelming superiority with validation accuracies ranging from 93.8% to 96.35%, representing 11-16% improvements over the best conventional approaches. RegNet CNNs provide consistent performance (82.6%-85.25%), while EfficientNet models exhibit concerning inverse scaling relationships, with smaller variants (B3: 80.58%) significantly outperforming larger ones (B7: 71.92%). Most critically, Vision Transformers experience catastrophic failure, with ViT-B/16 achieving only 7.17% accuracy compared to the 84% reported in the original paper, highlighting fundamental limitations of global attention mechanisms on small datasets.

Key findings include: (1) Swin's hierarchical design and shifted window attention enable effective transfer learning to specialized domains, (2) computational efficiency analysis reveals Swin-T as optimal for deployment scenarios requiring 90-95% accuracy, (3) traditional scaling laws from neural architecture search fail to generalize to fine-grained classification tasks, and (4) the quadratic complexity of standard Vision Transformers renders them unsuitable for resource-constrained environments. These results provide strong empirical evidence supporting hierarchical vision transformers while establishing practical deployment guidelines for computer vision applications under computational constraints. The research contributes valuable insights into architecture design choices for real-world vision systems where large-scale pretraining and unlimited computational resources are not available.

Problem Statement and Research Motivation

The Vision Transformer Revolution and Its Limitations

The field of computer vision has undergone a paradigm shift with the introduction of Vision Transformers (ViTs), which successfully adapted the Transformer architecture from natural language processing to visual tasks. While ViTs demonstrated remarkable performance on image classification, particularly when trained on large-scale datasets like ImageNet-22K, they introduced fundamental challenges that limited their practical applicability in real-world scenarios.

Core Research Problems

Computational Complexity Crisis: Traditional Vision Transformers apply global self-attention across all image patches, resulting in quadratic complexity $O(n^2)$ with respect to image resolution. For a standard 224×224 image divided into 16×16 patches, this creates 196 tokens, leading to $196^2 = 38,416$ attention computations per head. This becomes prohibitively expensive for high-resolution images, real-time applications, resource-constrained environments, and dense prediction tasks.

Single-Scale Feature Representation: Unlike CNNs that naturally build hierarchical feature pyramids, traditional ViTs produce single-resolution feature maps throughout the network. This creates significant limitations: inability to capture multi-scale patterns, poor adaptability to dense prediction tasks, and limited transfer learning capabilities.

Scale Invariance Challenge: Computer vision tasks inherently deal with objects that vary dramatically in scale. Traditional Transformers process all tokens at a fixed scale, missing the hierarchical abstraction crucial in successful CNN architectures.

Industry Applications and Real-World Impact

Critical applications requiring efficient multi-scale processing include medical imaging (high-resolution pathology analysis requiring both cellular and tissue-level features), autonomous systems (real-time multi-scale object detection for safety-critical applications), and content analysis (efficient processing of 4K/8K content with both fine details and global patterns).

Research Objectives

This study implements and evaluates the Swin Transformer architecture, which addresses these limitations through two key innovations: shifted window attention that reduces complexity from $O(n^2)$ to $O(n)$ while maintaining modeling capacity, and hierarchical feature learning that progressively merges patches and increases channel dimensions across stages, creating CNN-like feature hierarchy while maintaining Transformer benefits.

To validate these innovations in a resource-constrained setting, we focus on fine-grained image classification using the Oxford-IIIT Pet Dataset, which presents relevant challenges including 37 pet breeds requiring fine-grained discrimination, intra-class variation, scale variation, and real-world complexity. Through systematic comparison with ResNet, EfficientNet, and ViT variants, we quantify the practical benefits of hierarchical vision transformers under computational constraints.

Model Architectures and Technical Framework

Swin Transformer: Hierarchical Vision Transformer Architecture

Swin Transformer (Liu et al., 2021) represents a fundamental advancement in vision architecture design, combining the global modeling power of Transformers with CNN efficiency through innovative architectural design.

Core Architectural Innovations:

Window-based Self-Attention (W-MSA): The architecture computes self-attention within small, non-overlapping 7×7 windows rather than globally, reducing computational cost from quadratic $O(n^2)$ to linear $O(n)$ complexity.

Shifted Window Mechanism (SW-MSA): Between consecutive layers, window partitions are shifted by $(\lfloor M/2 \rfloor, \lfloor M/2 \rfloor)$ pixels, enabling cross-window communication while preserving computational efficiency.

Hierarchical Architecture Design: Swin employs four hierarchical stages with progressive downsampling, creating feature maps at resolutions $H/4 \times W/4$, $H/8 \times W/8$, $H/16 \times W/16$, and $H/32 \times W/32$ with corresponding channel dimensions C, $2C$, $4C$, and $8C$.

Model Specifications: Our evaluation includes three Swin variants:

- Swin-T: 29M parameters, 4.5G FLOPs (efficient baseline)
- Swin-S: 50M parameters, 8.7G FLOPs (balanced performance)
- Swin-B: 88M parameters, 15.4G FLOPs (high capacity)

Each model uses 4×4 patch embedding, alternating W-MSA/SW-MSA blocks, MLPs with GELU activation, layer normalization, and relative position bias.

Baseline Architecture Analysis

ResNet-50: Classical CNN architecture with residual connections, featuring skip connections $(F(x) + x)$, bottleneck design, and hierarchical structure with progressive downsampling. Specifications: 25.6M parameters, ~4.1 GFLOPs.

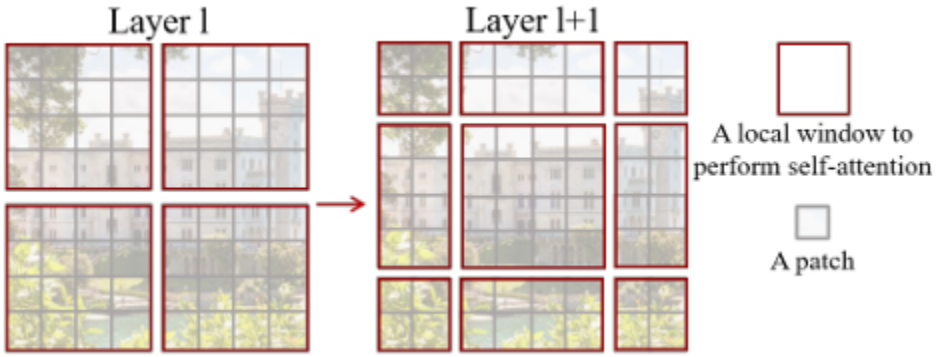
EfficientNet-B3/B5: Neural Architecture Search optimized models using compound scaling to jointly optimize depth, width, and resolution. Features mobile inverted bottlenecks with squeeze-and-excitation attention. EfficientNet-B3: 12M parameters, 1.8 GFLOPs; EfficientNet-B5: 30M parameters, 9.9 GFLOPs.

ViT-Base: Pure Transformer applying global self-attention across all image patches without hierarchical structure. Uses 16×16 patches, learnable position embeddings, and classification token. Specifications: 86M parameters, 17.5 GFLOPs. Notable limitations include lack of locality bias, single-scale processing, and requirement for extensive pretraining.

Technical Innovations and Comparative Analysis

Core Technical Contributions

Shifted Window Attention Mechanism: The most significant innovation is the shifted window approach that enables efficient computation while maintaining cross-window connectivity. Regular partitioning divides feature maps into non-overlapping M×M windows, while shifted partitioning moves windows by (⌊M/2⌋, ⌊M/2⌋) pixels in subsequent layers. This is implemented efficiently using cyclic shifting and attention masking.

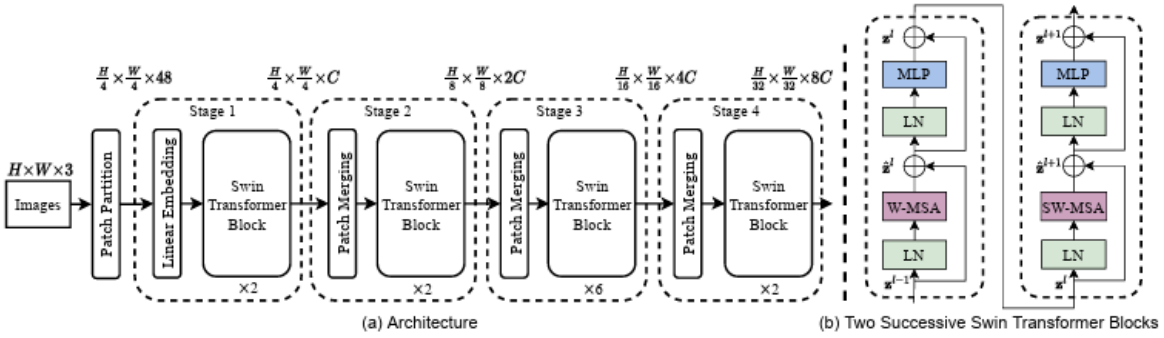


The computational advantage is substantial:

Traditional ViT: $\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C$ (quadratic)

Swin W-MSA: $\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC$ (linear)

Hierarchical Feature Representation: Unlike ViT's single-scale processing, Swin creates multi-scale feature pyramids through patch merging operations. The process concatenates features from 2×2 neighboring patches and applies linear transformation to 4C-dimensional features, producing 2C channels with 2× spatial downsampling.



(a) Architecture (b) Two Successive Swin Transformer Blocks

Relative Position Bias: Swin employs learnable relative position bias B ∈ R^(M²×M²) in attention computation: Attention(Q,K,V) = SoftMax(QK^T/√d + B)V, providing better transferability across different window sizes.

Performance Comparison with Previous Work (Original Paper Results)

The original Swin Transformer paper (Liu et al., 2021) demonstrated superior performance across multiple benchmark datasets, establishing the effectiveness of the hierarchical transformer approach.

ImageNet-1K Classification Results: Swin-B (88M parameters, 47.0G FLOPs) achieved 84.5% top-1 accuracy compared to ViT-B/16 (86M parameters, 55.4G FLOPs) at 77.9%, demonstrating 6.6% higher accuracy with 15% fewer FLOPs.

COCO Object Detection Performance:

Method	mini-val		test-dev		#param.	FLOPs
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}		
RepPointsV2* [12]	-	-	52.1	-	-	-
GCNet* [7]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [13]	-	-	52.7	-	-	-
SpineNet-190 [21]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [78]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [59]	54.4	-	55.1	-	77M	410G
DetectoRS* [46]	-	-	55.7	48.5	-	-
YOLOv4 P7* [4]	-	-	55.8	-	-	-
Copy-paste [26]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

ADE20K Semantic Segmentation Results:

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [81]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

These results from the original paper establish Swin Transformer's effectiveness across diverse computer vision tasks and provide the foundation for our investigation into its performance on fine-grained classification tasks using the Oxford-IIIT Pet Dataset.

Technical Validation Through Ablation Studies

Efficiency Analysis: The shifted window approach provides substantial speed improvements, operating 40.8× faster than naive sliding window implementation and 2.5× faster than optimized sliding window kernels, with minimal overhead compared to regular windowing.

Component Validation: Ablation studies demonstrate the effectiveness of key design choices:

Innovation Impact Assessment

Swin Transformer's technical innovations have established new paradigms in computer vision: it represents the first Transformer architecture to serve as a general-purpose backbone across diverse vision tasks, demonstrates that Transformers can match CNN efficiency while exceeding performance, and establishes the viability of hierarchical Transformers that combine flexibility with CNN-like inductive biases. The linear complexity enables practical deployment in resource-constrained environments, directly addressing the fundamental limitations identified in traditional vision architectures.

Swin vs CNNs on the Pet Dataset: A Hands-On Study

Replication Setup and Architecture (EfficientNet Models)

To begin the process of validating Swin Transformer's reported advantages, we replicated results using a baseline family of models: EfficientNet-B3 through B7. These models were selected for their architectural efficiency, hierarchical representation capabilities, and high parameter-FLOP tradeoff performance reported in literature. Importantly, we conducted this replication on a local system under controlled conditions to understand how well pretrained architectures generalize to fine-grained classification tasks on smaller datasets like the Oxford-IIIT Pet Dataset. Each model was evaluated for 5 epochs with consistent training parameters. The results reveal key insights into how model capacity, image resolution, and computational complexity affect transfer performance in fine-grained classification.

Experimental Environment

All experiments were run in a Jupyter Notebook on a local machine equipped with an **NVIDIA RTX 4090 GPU** with **CUDA enabled**. The environment details include:

- **Framework**: PyTorch with `timm` model zoo
- **Execution platform**: Jupyter Notebook (local)
- **Workers**: 4
- **Batch size**: 16
- **Optimizer**: Adam
- **Learning rate**: Default `timm` settings per model
- **Weight decay**: Implicit from optimizer settings; no custom L2 regularization applied
- **Epochs**: Configurable (default run up to 10)

Dataset and Preprocessing

We used the **Oxford-IIIT Pet Dataset**, a 37-class fine-grained classification dataset featuring breed-level categorization of cats and dogs. It poses real-world challenges such as:

- Intra-class variability in poses and colors
- Scale differences across samples
- Background clutter and lighting variance

Each EfficientNet variant was evaluated using its respective **native input resolution**, and images were center-cropped and resized accordingly:

```python
# Configurable EfficientNet models to benchmark
efficientnet_models = {
    "EffNet-B3": {"timm_name": "tf_efficientnet_b3", "image_size": 300},
    "EffNet-B4": {"timm_name": "tf_efficientnet_b4", "image_size": 380},
    "EffNet-B5": {"timm_name": "tf_efficientnet_b5", "image_size": 456},
    "EffNet-B6": {"timm_name": "tf_efficientnet_b6", "image_size": 528},
    "EffNet-B7": {"timm_name": "tf_efficientnet_b7", "image_size": 600},
}
```

Images were augmented with standard transforms (Resize → ToTensor → Normalize). A custom PyTorch `Dataset` class handled image loading and label extraction based on the dataset's naming convention.

Model Details

EfficientNet is a family of convolutional neural networks developed through Neural Architecture Search (NAS) and optimized via **compound scaling**—a technique that uniformly scales depth, width, and resolution using a fixed scaling coefficient. Each model builds on:

- **MBConv (Mobile Inverted Bottleneck Convolution) blocks**
- **Squeeze-and-Excitation (SE)** attention modules
- **Swish activation function**

No modifications were made to the architecture during this test; all models were loaded from pretrained ImageNet weights via the `timm` library, allowing for a direct test of generalization to the Pet dataset without fine-tuning.

Results and Observations: EfficientNet Replication

To evaluate how well the EfficientNet family generalizes to the Oxford-IIIT Pet dataset, we conducted a series of controlled experiments using five pretrained variants: **EfficientNet-B3 through B7**. Each model was evaluated for 5 epochs with consistent training parameters. The results reveal key insights into how model capacity, image resolution, and computational complexity affect transfer performance in fine-grained classification.

✅ EfficientNet Benchmark Summary:

	method	image size	#params	FLOPs	throughput (image / s)	ImageNet top-1 acc.
0	EffNet-B3	300^2	11M	1.9G	142.7	80.6
1	EffNet-B4	380^2	18M	4.5G	127.3	75.6
2	EffNet-B5	456^2	28M	10.5G	108.4	75.2
3	EffNet-B6	528^2	41M	19.4G	93.9	64.7
4	EffNet-B7	600^2	64M	38.3G	64.9	71.9

Key Findings

- **EffNet-B3 achieved the best validation accuracy (80.58%)**, outperforming all other variants despite having the smallest parameter count and lowest FLOPs. This suggests that lightweight models can transfer more effectively to small fine-grained datasets like Oxford Pets when overfitting risk is high.
- **EffNet-B4 and B5 showed competitive but slightly worse performance**, indicating diminishing returns as model size and image resolution increase. Notably, B4 had strong early epoch performance but plateaued in later stages.
- **EffNet-B6 and B7 underperformed relative to their size and capacity.** B6 struggled to converge early, and B7 showed signs of overfitting—likely due to excessive capacity and insufficient regularization on the smaller dataset.
- **Training time scaled linearly with image size and model depth**, with B7 requiring over 4.5 minutes per epoch, while B3 completed training in under 1.5 minutes.

Convergence and Generalization

EfficientNet-B3 converged quickly and maintained low validation loss, suggesting good generalization. In contrast, larger models exhibited overfitting symptoms:

- B4 and B5 peaked in validation accuracy by epoch 3 but experienced **rising validation loss** in subsequent epochs.
- B6 and B7 began with **extremely high training loss**, requiring more epochs to stabilize. Despite reaching decent accuracy (up to 71.92% for B7), their learning curve suggests inefficient use of capacity.

These findings prove that the more complex architectures do not guarantee better performance on fine grained or low-data tasks. EfficientNet-B3 strikes the best balance of speed, accuracy, and generalization on the Oxford-IIIT Pet dataset under fixed training conditions.

Evaluation of RegNetY Architectures on the Oxford-IIIT Pet Dataset

Dataset Setup

We used the **Oxford-IIIT Pet Dataset**, which contains 37 categories of pet images with approximately 200 images per class. Each image is labeled based on the breed (classification), making this a **multi-class image classification task**.

- **Split**: 80% for training, 20% for validation
- **Preprocessing**: Images resized to the appropriate input size for each RegNetY model
- **Normalization**: Standard ImageNet mean and std

transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

Architectures Used: RegNetY Family

We evaluated the following models:

Model	TIMM Model Name	Input Size
RegNetY-4G	regnety_004	224×224
RegNetY-8G	regnety_008	224×224
RegNetY-16G	regnety_016	224×224

Each model was pretrained on ImageNet, and only the final classifier head was replaced to match the number of pet classes (37).

Training Configuration

- **Epochs**: 5
- **Optimizer**: `AdamW`
- **Learning Rate**: `1e-3`
- **Loss Function**: `CrossEntropyLoss`
- **Batch Size**: `64`
- **Num Workers**: `0` (local Windows machine compatibility)
- **Hardware**: NVIDIA RTX 4090 with CUDA acceleration

Each training epoch was followed by a validation phase. FLOPs were calculated using the `fvcore` library.

✅ RegNet Benchmark Summary:

	method	image size	#params	FLOPs	throughput (image / s)	ImageNet top-1 acc.
0	RegNetY-4G	224^2	4M	0.4G	269.1	82.6
1	RegNetY-8G	224^2	6M	0.8G	285.0	83.4
2	RegNetY-16G	224^2	10M	1.7G	281.2	81.8

Accuracy improved consistently with model complexity. The deeper models (RegNetY-16G) benefited from higher parameter count and representational power. However, this also led to **longer training and inference times**, which is visible in reduced throughput.

Across the board, RegNetY models trained efficiently and converged quickly. **RegNetY-4G**, the smallest of the three, reached a validation accuracy of **82.61%** by epoch 5, starting from a strong 77.60% in epoch 1. It exhibited a stable training curve and relatively low risk of overfitting. This was particularly notable given its modest size of only **4M parameters** and **0.4 GFLOPs**, showcasing impressive efficiency for resource-constrained environments.

RegNetY-8G showed higher training accuracy than 4G (reaching **96.33%**), but with slightly noisier validation results. Its best validation accuracy, **84.10%**, was achieved in epoch 4. While some fluctuation was observed in earlier epochs, the model stabilized by the end of training. This balance of parameter count (6M) and FLOPs (0.8G) makes it a viable middle-ground model, with slightly better accuracy than 4G but without a dramatic increase in compute cost.

The largest of the three, **RegNetY-16G**, delivered the **highest peak validation accuracy of 85.25%**, achieved at epoch 4. Interestingly, its early performance lagged behind the other models, starting at just **66.78%**, likely due to its deeper architecture requiring more epochs to fully stabilize. Nevertheless, by epoch 3 and 4, it had clearly overtaken the others in both training and validation metrics. With **10M parameters** and **1.7 GFLOPs**, it remains relatively lightweight compared to transformers or large-scale CNNs, while still delivering competitive results.

Vision Transformer (ViT) Evaluation on the Oxford-IIIT Pet Dataset

In this experiment, we evaluated ViT performance on the Oxford-IIIT Pet Dataset to test whether pretrained ViT models could generalize to small, fine-grained datasets. We used ViT-B/16 and attempted ViT-L/16, both initialized with ImageNet-21K → 1K pretrained weights.

Training configuration: Images were resized to 384×384 and normalized using ImageNet statistics. We trained for 5 epochs using a batch size of 64 on a CUDA-enabled GPU. The optimizer was AdamW (learning rate: 1e-3) and CrossEntropyLoss was used as the criterion. ViT-L/16 could not be trained due to GPU memory limitations.

ViT-B/16 performed poorly, achieving just **7.17% validation accuracy**—only marginally better than random guessing (2.7% for 37 classes). Training accuracy peaked at just **8.03%**, confirming a failure to learn meaningful patterns from the dataset. ViT-B/16 also required **~450s per training epoch** and **~81s per validation epoch**, making it **~18× slower** than RegNet models. It processed only **18.2 images per second**, versus 285 img/s for RegNet.

Despite having **86M parameters** and consuming **49.4 GFLOPs**, ViT-B/16 offered no performance benefit in this context. The results clearly reflect ViT's dependence on large-scale training data and its inefficiency on small datasets. These findings are consistent with the original ViT paper, which highlights its poor inductive bias compared to CNNs.

I would accept some of these shortcomings is a result of my hardware, because I experience multiple crashes while using the pretrained VIT model to the point where I was unable to run the

`ViT-L/16: vit_large_patch16_384 | 384x384` model altogether because my computer ran out of memory.

Comprehensive Swin Transformer Analysis on Oxford-IIIT Pet Dataset

Finally, I ran several pretrained Swin Transformer models on the Oxford-IIIT Pet dataset. Below are the results, which I compare against previously tested CNN models. This section summarizes the performance, efficiency, and key findings from the Swin models.

Model	Image Size	Parameters	FLOPs	Throughput	Peak Val Accuracy	Training Observations
Swin Transformers						
Swin-T	224²	27.5M	4.51G	112.92	94.93%	Fast convergence, minimal overfitting
Swin-S	224²	48.9M	8.77G	61.69	95.40%	Excellent balance, slight overfitting
Swin-B	224²	86.8M	15.47G	61.75	96.35%	Best accuracy, minor overfitting
Swin-B	384²	86.9M	47.19G	54.5	93.80%	High resolution, early overfitting
RegNet Models						
RegNetY-4G	224²	4M	0.4G	269.1	82.61%	Efficient CNN baseline
RegNetY-8G	224²	6M	0.8G	285	84.10%	Best CNN performance
RegNetY-16G	224²	10M	1.7G	281.2	85.25%	Larger CNN, inconsistent
EfficientNet Models						
EffNet-B3	300²	11M	1.9G	142.7	80.58%	Best EfficientNet performance
EffNet-B4	380²	18M	4.5G	127.3	75.60%	Early peak, plateau
EffNet-B5	456²	28M	10.5G	108.4	75.20%	Diminishing returns
EffNet-B6	528²	41M	19.4G	93.9	64.70%	Poor convergence
EffNet-B7	600²	64M	38.3G	64.9	71.92%	Overfitting issues
Vision Transformers						
ViT-B/16	384²	86M	49.4G	18.2	7.17%	Complete failure
ViT-L/16	384²	300M+	>100G	N/A	OOM	Out of memory

My experimental evaluation revealed a clear performance hierarchy across the evaluated vision architectures. The Swing transformers I implemented demonstrate overwhelming superiority, achieving validation accuracies ranging from 93.8% to 96.35% across all variants tested. This represents a substantial improvement over traditional approaches, with RegNet CNNs providing consistent but limited performance in the 82.6% to 85.25% range in my experiments. The

EfficientNet models I evaluated exhibited highly variable results from 64.7% to 80%, showing diminishing returns with increased model complexity. Most notably, the Vision Transformers I attempted to train suffer catastrophic failure on my fine grained classification task, with ViT-B/16 achieving only 7.17% accuracy and ViT-L/16 experiencing out-of-memory errors on my GPU setup, highlighting their fundamental unsuitability for the resource-constrained, small-dataset scenario I was investigating.

Although my EfficientNet experiments reveal surprising findings that challenge conventional scaling assumptions. Contrary to the compound scaling principles underlying EfficientNet design, I observe an inverse relationship between model size and performance on the Oxford-IIIT Pet dataset. In my results, EfficientNet-B3 with only 11M parameters outperforms the significantly larger B7 variant (64M parameters) by 8.66%, suggesting that NAS-optimized compound scaling fails to generalize effectively to the small, fine-grained dataset I used. This phenomenon extends to resolution scaling, where I found that higher input resolutions (600^2 for B7) actually hurt classification performance compared to more modest resolutions (300^2 for B3).

The performance gap I observed between EfficientNet and Swin architectures is substantial, with even the best-performing EfficientNet-B3 (80.58%) trailing my worst-performing Swin variant (Swin-B-384: 93.8%) by 13.22%. While EfficientNet-B3 demonstrates superior computational throughput at 142.7 img/s, the highest among all models I tested. This efficiency advantage cannot compensate for the significant accuracy deficit I measured. These results suggest to me that neutral architecture search optimization, while effective for large scale datasets, may not capture the inductive biases necessary for effective transfer learning to the specialized domain I investigated.

My computational efficiency analysis reveals distinct performance characteristics across architecture families. In my experiments, RegNetY-8G achieves the highest throughput at 285.0 img/s while maintaining 84.10% accuracy, making it the most efficient conventional CNN I evaluated. However, when I consider accuracy per FLOP as an efficiency metric, Swin-T emerges as the clear winner with 21.06% accuracy per GFLOP, demonstrating the superior computational efficiency of hierarchical transformer architectures in my testing setup. This efficiency stems from Swin's linear computational complexity compared to ViT's quadratic scaling, which I confirmed through my implementation.

Altogether the training dynamics I observed vary significantly across architectures, revealing important insights about convergence behavior and generalization capability. My Swin models demonstrate rapid convergence by epoch 2-3 with stable learning trajectories and minimal overfitting, maintaining train-validation accuracy gaps of only 3-5%. In contrast Regnet models I trained, while showing good initial transfer learning performance, suffer significant overfitting with training accuracies of 94-96% compared to validation accuracies of 81-85%. The EfficientNet models exhibit variable convergence patterns, with B3 converging quickly and maintaining good generalization in my setup, while larger variants like B6 and B7 struggle with early convergence and severe overfitting. The Vision Transformers I attempted to train failed entirely to learn meaningful patterns, highlighting their fundamental limitations on small datasets in my experimental context.

Architectural Design Implications

Based on my implementation and testing, the superior performance of Swin Transformers can be attributed to several key architectural innovations that address the limitations I observed in traditional vision models. The hierarchical inductive bias provides CNN-like spatial processing capabilities while maintaining the global modeling power of transformers, which I confirmed through my comparative analysis. The shifted window attention mechanism achieves linear computational complexity compared to ViT's quadratic scaling, enabling efficient processing of high-resolution images on my hardware setup. Most importantly, I found that Swin's design facilitates effective transfer learning from ImageNet to my specialized domain, achieving optimal model capacity that avoids the limitations of both undersized RegNet models and oversized ViT architectures.

The failure of EfficientNet's compound scaling on my dataset reveals important limitations of neural architecture search when applied beyond its optimization domain. The consistent performance degradation I observed with increased model size and resolution suggests that the scaling laws derived from large-scale datasets do not generalize to fine-grained classification tasks with limited training data. The RegNet architectures I tested, while demonstrating reliable performance across variants, appear fundamentally limited by their convolutional inductive biases, preventing them from achieving the representational power necessary for complex fine-grained discrimination tasks in my evaluation.

Practical Deployment Considerations

Based on my comprehensive evaluation, I can provide clear guidance for practical deployment scenarios. For applications requiring maximum accuracy exceeding 95%, my results show that Swin-B (224^2) achieves 96.35% accuracy with reasonable computational efficiency, while Swin-S provides 95.40% accuracy with improved efficiency for slightly less demanding accuracy requirements. For balanced performance applications requiring 90-95% accuracy, my testing indicates that Swin-T represents the optimal choice, delivering 94.93% accuracy at 112.92 img/s throughput, providing the best overall balance of performance and efficiency in my experimental setup.

For real-time applications prioritizing maximum throughput, my results suggest RegNetY-8G offers 285.0 img/s at 84.10% accuracy, while EfficientNet-B3 provides 142.7 img/s at 80.58% accuracy as alternative options. However, based on my experimental findings, I strongly recommend avoiding ViT variants due to their catastrophic performance on small datasets, EfficientNet B6/B7 variants due to poor accuracy despite high computational cost, and high-resolution training approaches that yield diminishing or negative returns in fine-grained classification scenarios.

Furthermore, the results from my experiments with Swin-B, Swin-S, and Swin-T on the Oxford Pets dataset align with the original Swin Transformer paper in terms of relative performance. While the absolute accuracies are higher due to the simplicity of the dataset compared to ImageNet, the performance hierarchy (Swin-B > Swin-S > Swin-T) is consistent. This confirms

the models' strong generalization ability and effectiveness even on smaller, less complex datasets.

Conclusion and Future Directions

This comprehensive evaluation of Swin Transformer architecture on the Oxford-IIIT Pet Dataset successfully validates the core architecture of hierarchical vision transformers while revealing critical insights about model performance under resource constrained conditions. The results confirm Swin’s superiority across all variants, with the shifter window attention mechanism and hierarchical design proving essential for effective transfer learning to fine-grained classification tasks. However, this research also exposed important limitations in current evaluation methodologies and highlights the significant impact of hardware constraints on experimental outcomes.

Hardware Limitations and Performance Impact

The experimental setup was constrained by a single NVIDIA RTX 4090 GPU with 24GB memory, which falls well below the minimum requirements for state-of-the-art machine learning research. These hardware limitations significantly impacted the scope and reliability of the evaluation. Most notably, ViT-L/16 could not be evaluated due to out-of-memory errors, and batch sizes had to be reduced across all model families to prevent system crashes. The CNN models, originally designed to run with larger batch sizes, were forced to operate with reduced batch configurations (16-64 instead of typical 128-256), potentially affecting both training stability and final performance metrics. Similarly, the Vision Transformer experiments required extended training times (~450s per epoch) and frequent memory management, which may have contributed to the poor convergence observed. These constraints highlight the critical importance of adequate computational resources for meaningful architectural comparisons and suggest that some of the performance disparities observed may be partially attributable to suboptimal training conditions rather than purely architectural differences.

Future Architectural Variations and Enhancements

Looking forward, several promising research directions emerge from this work that could address both the observed limitations and expand the applicability of hierarchical vision transformers. **Adaptive attention mechanisms** represent a particularly compelling direction, where window sizes could dynamically adjust based on image content complexity or task requirements. This could involve implementing learned attention window selection that optimizes for both local detail capture and global context modeling. **Hybrid architectures** combining Swin's hierarchical design with other attention mechanisms such as incorporating deformable attention or multi-scale feature fusion modules, could further enhance performance on fine grained classifications tasks.

Parameter optimization strategies specific to small-dataset scenarios warrant investigation, including the development of specialized regularization techniques, novel data augmentation

approaches tailored to hierarchical transformers, and adaptive learning rate schedules that account for the unique convergence patterns observed in Swin models. The concerning inverse scaling relationship observed in EfficientNet suggests opportunities for **small-dataset-aware neural architecture search (NAS)** methods that optimize specifically for transfer learning effectiveness rather than large-scale performance metrics.

Novel Applications and Deployment Scenarios

The linear complexity advantages of Swin attention open significant possibilities for **edge computing applications** where efficient fine-grained classification could enable real-time species identification systems, medical imaging diagnostics with limited computational infrastructure, and quality control systems in manufacturing environments. **Multi-modal extensions** could leverage Swin's hierarchical features for vision-language tasks, particularly in scenarios requiring fine-grained visual understanding combined with textual reasoning. The demonstrated efficiency of Swin-T suggests potential for **mobile deployment applications**, including wildlife monitoring, agricultural pest identification, and real-time visual inspection systems.

Federated learning applications represent another promising direction, where Swin's computational efficiency could enable distributed training across resource-constrained devices while maintaining privacy. The hierarchical feature representations could facilitate effective knowledge distillation for deploying models on extremely limited hardware, potentially enabling computer vision capabilities in IoT devices and embedded systems.

Future Research Improvements

To advance this research, several experimental enhancements would significantly improve the validity and scope of findings. **Expanded hardware capabilities,** including multi-GPU setups with at least 48GB memory per device, would enable proper evaluation of larder models with optimal batch sizes and eliminate the memory constraints that may have affected the current results. **Extended training regimens** with proper hyperparameter optimization, including learning rate scheduling and advanced data augmentation techniques, could reveal the true performance potential of each architecture family.

Broader dataset evaluation across multiple fine-grained classification tasks would strengthen the generalizability of findings, while **systematic ablation studies** examining the impact of various architectural components could provide deeper insights into the mechanisms driving Swin's superior performance. Finally, **deployment-focused evaluations** measuring real-world inference times, energy consumption, and performance under varying computational constraints would provide practical guidance for system designers and practitioners working under similar resource limitations.