Comparison of ResNet and Vision Transformer (ViT) Models

Ximing Shen **Zhijing Zhang** **Xi Yan**

Abstract

Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) are two of the most widely used architectures for various computer vision tasks. However, they are fundamentally differ in how they perceive and process images. We study the impact of dataset size on the performance of two architectures, discovering that pre-trained ResNet can perform better than pre-trained ViT due to ResNet's inductive biases. We also explore how ResNet and ViT differ in processing images at different layers, and how their approach differ in performance, demonstrating ViT's tendency to global features. Finally, we design a hybrid ViT combining ResNet, aiming to improve ViT's locality on image processing.

1 Introduction

ViTs lack image-specific inductive biases, requiring large pretraining datasets, while ResNets leverage strong inductive biases to perform well on smaller datasets . To explore the effect of dataset size on performance, we evaluate the memory and time complexity of both models on image classification tasks using datasets of varying sizes.

A key distinction lies in how ViTs and ResNets incorporate local and global information. As noted by Raghu et al. [2021], ViTs integrate global information at lower layers, unlike ResNets, influencing their behavior. We analyze this difference by examining how local and global representations evolve across layers, using Grad-CAM proposed by Selvaraju et al. [2017] to visualize the regions contributing most to predictions.

2 Related Work

Several studies have compared ViT and ResNet, a deviation of CNN by He et al. [2016]. ghi [2021] showed that ViTs capture broader contextual information through optimization-based visualization, whereas CNNs is good at extracting local features. Paul and Chen [2021] found that ViTs demonstrate stronger robustness against common corruptions due to their self-attention mechanisms. Raghu et al. [2021] found that ViTs rely heavily on skip connections and global spatial information, while CNNs depend on localized receptive fields. These studies emphasize that there are some architectural and functional differences between ViTs and CNNs, showing their respective strengths and limitations across various datasets and tasks.

3 Experimental Setup

3.1 Model Selection

We use pretrained ResNet-101 and ViT-B/16 models from PyTorch, initially trained on ImageNet, and modify their final layers to output two classes for binary classification. We fine-tune the last two layers for ResNet-101, and the last three layers for ViT-B/16 A . Earlier layers are freezed to leverage

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

transferable features from the pretrained models and reduces overfitting, given the similarity between our dataset and ImageNet and the limited size of our dataset.

3.2 Dataset

For this study, we use a subset of the CIFAR-10 benchmark dataset, selecting only the "cat" and "dog" classes. This results in a binary classification dataset of 12,000 32x32 images (6,000 per class). During preprocessing, we apply random horizontal flipping and random rotations (up to 10 degrees), followed by normalization using CIFAR-10 mean and standard deviation. Images are resized to 224x224 for compatibility with both ResNet-101 and ViT-B/16 models.

3.3 Training Setup

To investigate the effect of dataset size on model performance, we conduct experiments using 25%, 50%, and 100% of the training data. Training is performed on an RTX 2060 GPU with consistent hyperparameters across both models: Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and momentum of 0.9. The batch size is set to 64, and each training run spans 10 epochs.

3.4 Evaluation Metrics

We evaluate model performance using validation accuracy, as well as memory and time complexity. Memory complexity is assessed based on GPU memory usage during training, and time complexity is determined by the average time per epoch.

4 Model Comparison

4.1 Performance

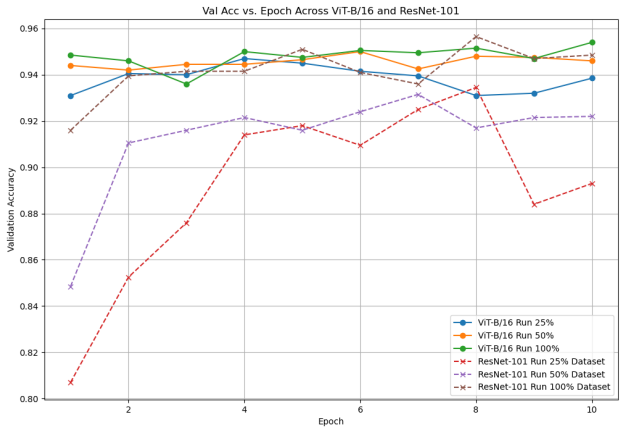


Figure 1: Validation accuracy vs. epoch for ResNet-101 and ViT-B/16

Table 1: ViT-B/16 and ResNet-101 performance

Model	Dataset Size	Memory Usage Per Epoch (MB)	Avg Time Per Epoch (s)	Highest Acc
ViT-B/16 (25%)	2500	$\sim$ 2559	$\sim$ 31.9	0.9470
ViT-B/16 (50%)	5000	$\sim$ 2559	$\sim$ 63.5	0.9500
ViT-B/16 (100%)	10000	$\sim$ 2559	$\sim$ 127.7	0.9540
ResNet-101 (25%)	2500	$\sim$ 4534	$\sim$ **22.9**	0.9345
ResNet-101 (50%)	5000	$\sim$ 4534	$\sim$ 48.5	0.9315
ResNet-101 (100%)	10000	$\sim$ 4534	$\sim$ 92.9	**0.9565**

From Figure1, we observe that ViT-B/16 demonstrates better performance on the smaller datasets (25% and 50%), while ResNet-101 slightly outperforms ViT-B/16 when trained on the full dataset. Additionally, Vit-B/16 shows less significant improvement both within individual dataset sizes across epochs and as the dataset size increases. This trend suggests that ViT-B/16 benefits from its pretraining on a significantly larger dataset, which enables it to generalize better to new data, particularly when the training dataset is small.

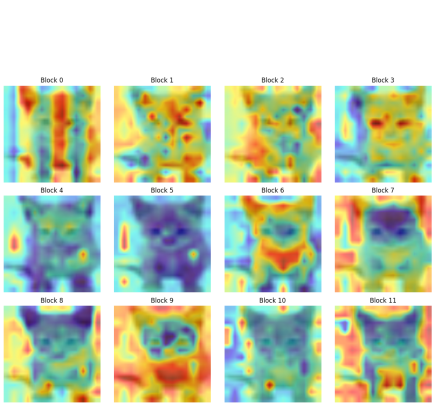
Focusing on the full dataset, ResNet-101 achieves its highest validation accuracy of 0.9565 at epoch 8, while ViT-B/16 achieves its highest validation accuracy of 0.9540 at epoch 10. The average validation accuracy across epochs for the full dataset is approximately 0.9415 for ResNet-101 and 0.9481 for ViT-B/16, indicating that both models perform similarly when trained on larger datasets.

As the dataset size increases, ResNet-101 benefits more significantly, narrowing the performance gap with ViT-B/16 on the full dataset. This result suggests that ViT-B/16's advantage on smaller datasets stems from its pretrained capacity and global self-attention, rather than an inherent superiority of the global attention mechanism over ResNet's convolutional approach.

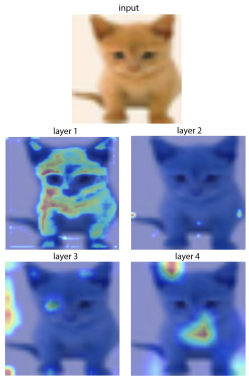
4.2 Computational Efficiency

Table 1 shows the comparison of time and memory complexity for each model.
We observe that ResNet-101 is much faster than ViT-B/16 during both training and validation, despite ResNet-101 requiring more memory, especially during training. As the dataset size increases, both models experience slower speeds, but ViT-B/16's speed is impacted more significantly than ResNet-101. This greater computational cost may stem from the self-attention mechanism in ViT, which involves significant overhead due to its quadratic complexity with respect to the number of patches. In contrast, convolutional networks like ResNet-101 can efficiently process images using localized feature extraction, storing more intermediate feature maps. While this results in higher memory usage, it also leads to faster computations.

4.3 Grad-CAM Visualization



(a) Grad-CAM on ViT-B/16 across all blocks



(b) Grad-CAM on ResNet-101 across all layers

Figure 2: Grad-CAM on ViT-B/16 and ResNet-101

We use Grad-CAM to visualize which portions of the images contributes most at each layer. From Figure2, we observe that the ViT gradually refines its focus across blocks. However, in the final block, the focus shifts predominantly to the background instead of the cat. This behavior highlights ViT's tendency to prioritize global information, which may sometimes lead to overemphasis on less relevant regions. In contrast, ResNet exhibits a stronger focus on local patterns, with its attention progressively narrowing to semantically important features such as the cat's face and body.

4.4 Hybrid Approach

To better experience ViT and ResNet's differences, we also implement a naive hybrid model for better comparison. We add a convolutional block with two 3×3 convolutional layers, each followed by ReLU activation, and output the image without downsampling. This pre-processed image is then fed into ViT-B/16 with the same training setup as used when training ViT-B/16 alone. This approach is inspired by the above Grad-CAM result and Raghu et al.'s findings, which shows that ViT exhibits lower attention to local features and does not learn to attend locally with limited training data. To address this, we added a convolutional block to help the model capture more local information in the early stages.

From figure 3, we observe that although this hybrid model achieves decent validation accuracy the full dataset, its accuracy is worse than training the two models (ResNet and ViT) independently, as done in our earlier experiments. This is likely because ViT-B/16 is pretrained on a large dataset that already incorporates local features effectively in its early layers. Furthermore, since we freeze most of the ViT weights during training, those pretrained features are used directly without additional modification.

Due to the increasing performance as the dataset size increases, we believe that the hybrid model has the potential to outperform ViT with further fine-tuning and the use of a larger dataset to optimize both the convolutional block and the ViT's transformer encoder block.

4.4.1 A smaller hybrid model

To verify our idea, we use a slightly more complex convolutional layer B with a smaller pretrained ViT model, ViT-Tiny Patch16 224, from PyTorch to compare the effect of an additional convolutional layer. By using a smaller model with fewer trainable parameters, we reduce the model's pretrained capacity and can thus better focus on the impact of the additional convolutional layer. Figure 4 shows the results of the hybrid ViT-Tiny and the standard ViT-Tiny.

From the figure, we observe that when using the full dataset, the hybrid model achieves a higher validation accuracy, suggesting that the newly added convolutional layer helps the model generalize better.

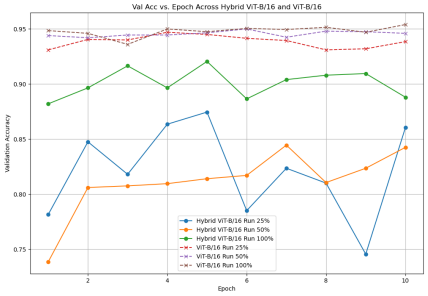


Figure 3: Validation accuracy vs. epoch for hybrid ViT-B/16 and ViT-B/16

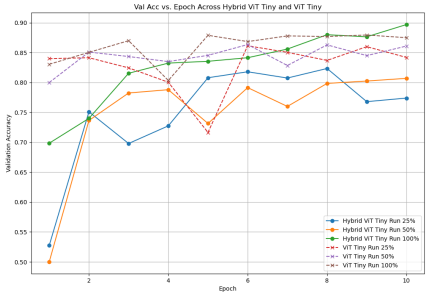


Figure 4: Validation accuracy vs. epoch for hybrid ViT-Tiny and ViT-Tiny

5 Conclusion

Pretrained ViT trained on large datasets are found to rely less on the fine-tuning dataset. In contrast, ResNets leverage fine-tuning datasets more effectively due to their strong inductive biases. Additionally, we demonstrate that ViTs tend to make inferences based on global features, a result of their self-attention mechanism spanning all image patches. On the other hand, ResNets focus on local features, as their convolutional kernels are designed to extract localized patterns. To enhance ViT's ability to capture local features, we implement a naive hybrid ViT model; it incorporates a

convolutional block to preprocess input images into patches before passing them through the self-attention blocks. This design enables the hybrid ViT to first extract local features from the images and subsequently capture global features. It achieves better validation accuracy than original ViT-Tiny.

6 Limitation

The limited availability of GPU resources necessitated the use of relatively small datasets, which may not fully showcase the potential of ViT and ResNet. Additionally, these computational constraints impeded our ability to conduct extensive hyperparameter tuning and to train the models over an adequate number of epochs. Consequently, the models may not have attained their optimal performance levels. Future research could address these limitations by utilizing larger and more diverse datasets, exploring a wider range of hyperparameter configurations, and leveraging increased computational power to facilitate longer and more comprehensive training processes.

References

What do vision transformers learn? a visual exploration. *University of Maryland - College Park and New York University*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Swarup Paul and Pin-Yu Chen. Vision transformers are robust learners. *Carted and IBM Research*, 2021.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 12116–12128. Google Research, Brain Team, 2021.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626. IEEE, 2017. doi: 10.1109/ICCV.2017.74.

A Appendix

Additional details on how we choose the number of layers to train during fine-tuning.

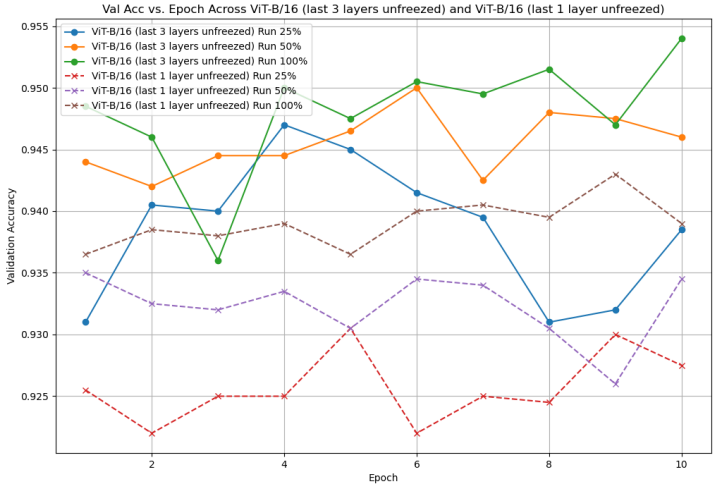


Figure 5: Unfreeze different number of layers of ViT-B/16

It can be observed that training with three unfrozen layers yields the highest validation accuracy.

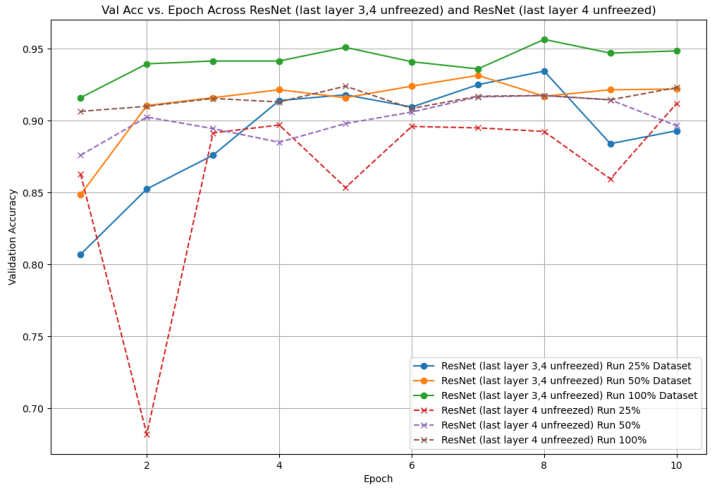


Figure 6: Unfreeze different number of layers of ResNet-101

Similarly, training with the last two unfrozen layers yields the highest validation accuracy.

B Appendix

The more complex convolutional layer.

```
self.cnn_block = nn.Sequential(
    nn.Conv2d(3, 32, kernel_size=3, padding=1),
    nn.BatchNorm2d(32),
    nn.ReLU(inplace=True),
    nn.Conv2d(32, 64, kernel_size=3, padding=1),
    nn.BatchNorm2d(64),
    nn.ReLU(inplace=True),
    # Downsample to reduce computation, keeping a balanced representation
    nn.MaxPool2d(2),  # output now 112x112
    nn.Conv2d(64, 3, kernel_size=3, padding=1),
    nn.BatchNorm2d(3),
    nn.ReLU(inplace=True)
)
```

Figure 7: Added convolutional layer