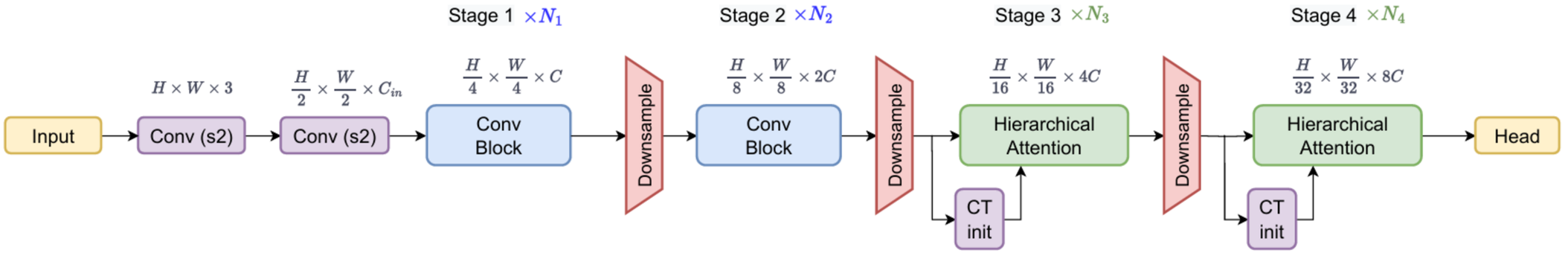
FASTERVIT

FAST VISION TRANSFORMERS WITH HIERARCHICAL ATTENTION

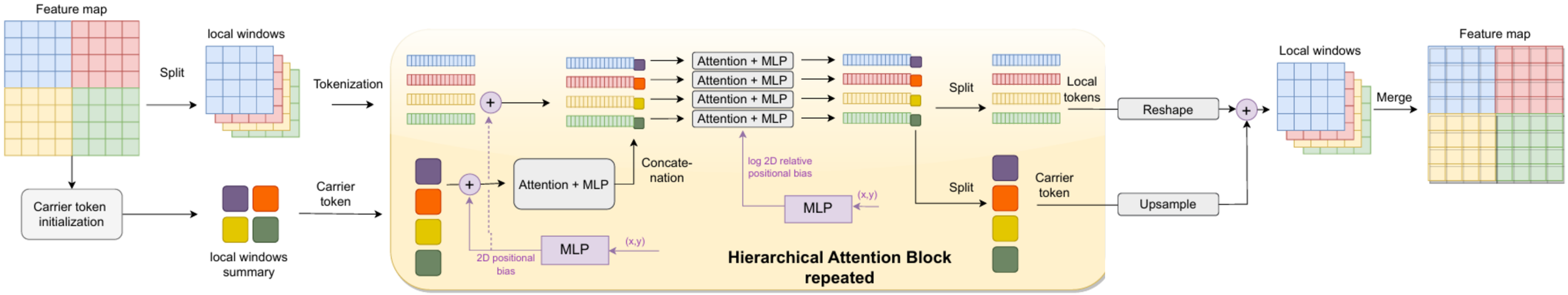
Ali Hatamizadeh, Greg Heinrich, Hongxu Yin,
Andrew Tao, Jose M. Alvarez, Jan Kautz, Pavlo Molchanov
NVIDIA

Presented by:-
Group: **VisionX**
210503H - I.P.D.D. Rajapaksha
210169L - W.W.R.N.S. Fernando

FasterViT- Architecture



- **Optimize throughput and GPU utilization.**
- **Hierarchical Attention** for efficient and scalable **modeling of high-resolution images**.



Experiment 01

key components:

✓ Training on Imagenet 1K and accuracy evaluation.

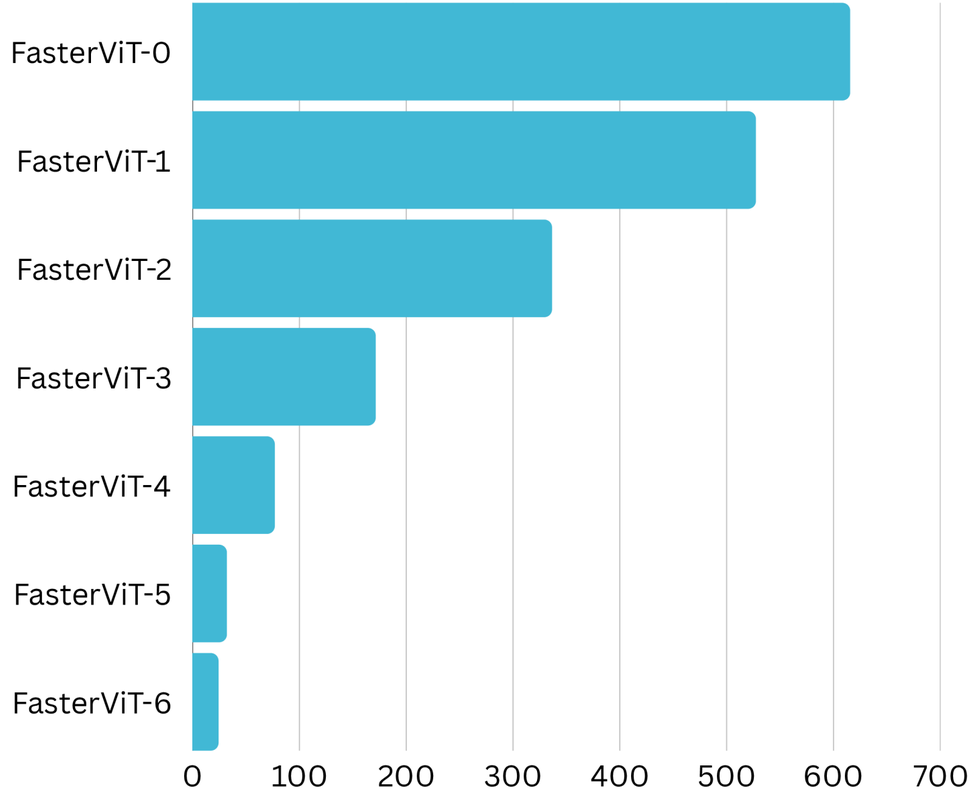
✓ Throughput benchmarking (images/sec).

Throughput benchmarking

- Used official **ImageNet-1K pretrained FasterViT models.**
- For each model, we run it multiple times with a simulated image batch (Batch Size = 128).
- The batch only imitates the size of real images(224×224 image).
- We record how long the model takes to process a fixed number of batches.

Model	Our Throughput (img/s) on NVIDIA GeForce RTX 5060	Paper Throughput on A100 (img/s)
FasterViT-0	615.58	755
FasterViT-1	527.34	650
FasterViT-2	336.48	596
FasterViT-3	171.42	490
FasterViT-4	76.92	410
FasterViT-5	32.09	340
FasterViT-6	24.27	275

Slower as model grows



Observation

✓ Larger models → slower throughput.

✓ Order matches the paper trend.

✓ Differences arise due to GPU type.

Experiment 01

Inference-only ImageNet accuracy evaluation

- Used Official **ImageNet-1K pretrained FasterViT** models.
- Evaluated on **1K-ImageNet validation subset**
- Used exact timm preprocessing.
- Same as paper Metrics reproduced.

```
========= SUMMARY OVER ALL MODELS =========
faster_vit_0_224    | N=1000 | Top-1= 87.50% | Top-5= 97.70%
faster_vit_1_224    | N=1000 | Top-1= 87.60% | Top-5= 97.90%
faster_vit_2_224    | N=1000 | Top-1= 89.00% | Top-5= 98.20%
faster_vit_3_224    | N=1000 | Top-1= 88.90% | Top-5= 98.80%
faster_vit_4_224    | N=1000 | Top-1= 88.70% | Top-5= 98.50%
faster_vit_5_224    | N=1000 | Top-1= 89.50% | Top-5= 98.50%
============================================
```

Reproduced results

Name	Acc@1(%)	Acc@5(%)
FasterViT-0	82.1	95.9
FasterViT-1	83.2	96.5
FasterViT-2	84.2	96.8
FasterViT-3	84.9	97.2
FasterViT-4	85.4	97.3
FasterViT-5	85.6	97.4

Paper results

- Our accuracy values = validation accuracy on ImageNet-1K (subset)
- Paper accuracy values = validation accuracy on ImageNet-1K (full 50k set)(not publically available)

Observations

✓ Accuracy trend matches the paper: FasterViT-0 < 1 < 2 < 3 < 4 < 5.

✓ Our Top-1/Top-5 values are slightly higher because of dataset difference.

Experiment 02 - Fine-Tuning for Image Classification

key components:

✓ Pretraining on ImageNet-21K.

✓ Fine-tuning on high-resolution ImageNet-1K and Report Top-1 accuracy. (**1000 classes**)

Limitations

- ImageNet-21K (14M images) and ImageNet-1K not publicly available
- Training FasterViT needs 8 × A100 GPUs
- Full finetuning can't be perform using pretrained weights of fastervit on imagenet-21k due to GPU limitations.

Fine-tuning

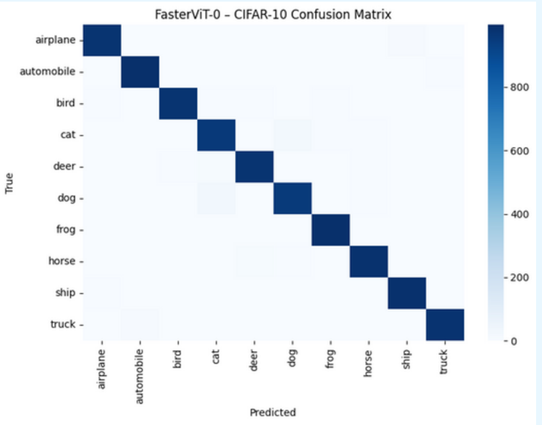
- Used Official **ImageNet-1K pretrained FasterViT** models.
- Dataset: **CIFAR-10 (10 classes)**
- Two fine-tuning settings:
  - FasterViT-0: Full fine-tuning
  - FasterViT-1: Head-only fine-tuning
  - FasterViT-4-21K-224: Head-Only Fine-Tuning

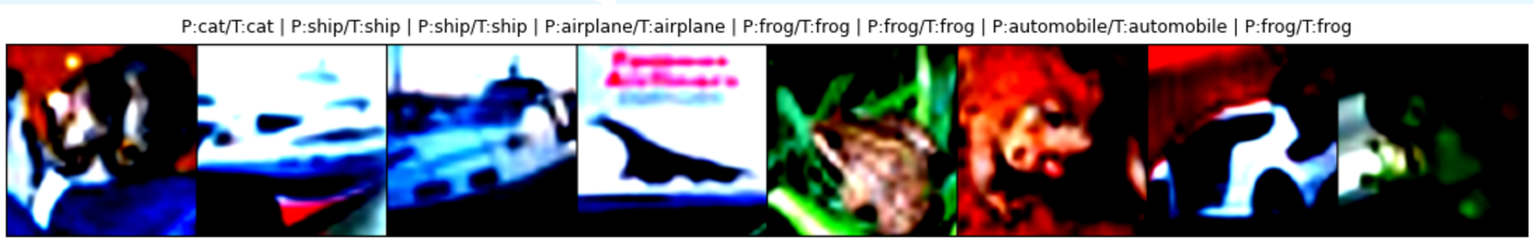
FASTERVIT-0: FULL FINE-TUNING RESULTS

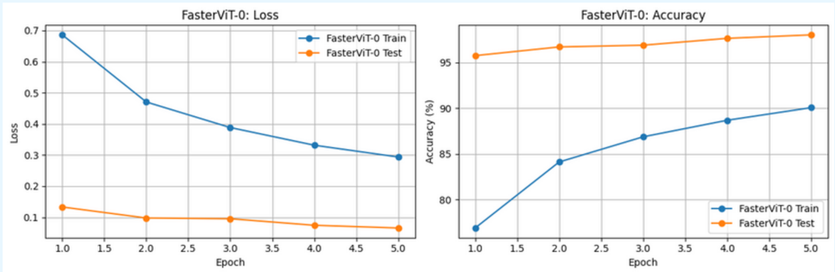
```
=== Building model faster_vit_0_224 ===
[Train] Epoch 01 | Loss: 0.6861 | Acc: 76.90% | Time: 359.7s
[Test ] Loss: 0.1330 | Acc: 95.73%
  -> New best accuracy: 95.73%, model saved.
[Train] Epoch 02 | Loss: 0.4704 | Acc: 84.11% | Time: 364.3s
[Test ] Loss: 0.0976 | Acc: 96.68%
  -> New best accuracy: 96.68%, model saved.
[Train] Epoch 03 | Loss: 0.3883 | Acc: 86.86% | Time: 364.0s
[Test ] Loss: 0.0953 | Acc: 96.87%
  -> New best accuracy: 96.87%, model saved.
[Train] Epoch 04 | Loss: 0.3315 | Acc: 88.67% | Time: 365.0s
[Test ] Loss: 0.0743 | Acc: 97.62%
  -> New best accuracy: 97.62%, model saved.
[Train] Epoch 05 | Loss: 0.2936 | Acc: 90.05% | Time: 364.4s
[Test ] Loss: 0.0656 | Acc: 98.00%
  -> New best accuracy: 98.00%, model saved.

Best test accuracy for faster_vit_0_224: 98.00%
```

Classification report for FasterViT-0:

	precision	recall	f1-score	support
airplane	0.987	0.980	0.983	1000
automobile	0.984	0.995	0.990	1000
bird	0.987	0.979	0.983	1000
cat	0.951	0.959	0.955	1000
deer	0.981	0.977	0.979	1000
dog	0.967	0.956	0.961	1000
frog	0.987	0.996	0.992	1000
horse	0.983	0.985	0.984	1000
ship	0.983	0.990	0.987	1000
truck	0.990	0.983	0.986	1000
accuracy			0.980	10000
macro avg	0.980	0.980	0.980	10000
weighted avg	0.980	0.980	0.980	10000





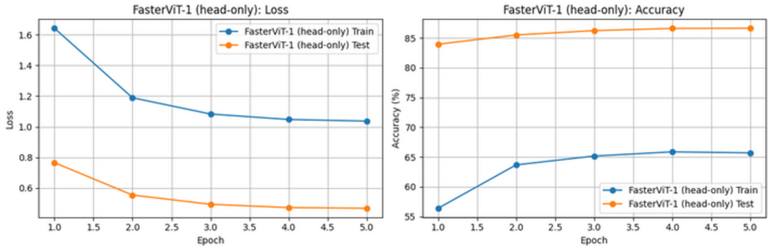


- Full fine-tuning of FasterViT gives highest downstream accuracy.
- Very high accuracy demonstrates that FasterViT-0 transfers extremely well from ImageNet-1K to CIFAR-10.
- Achieving 98% with only a few epochs shows strong pretrained representations..

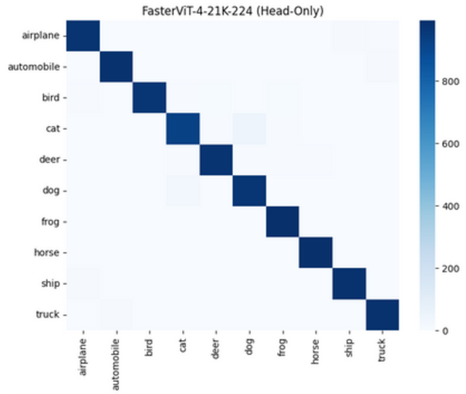
Experiment 02 - Fine-Tuning for Image Classification

FASTERVIT-1: HEAD-ONLY FINE-TUNING RESULTS

```
=== Building model faster_vit_1_224 ===
100%|██████████| 205M/205M [00:06<00:00, 31.8MB/s]
-> Backbone frozen; training head only.
[Train] Epoch 01 | Loss: 1.6421 | Acc: 56.40% | Time: 204.7s
[Test ] Loss: 0.7659 | Acc: 83.95%
  -> New best accuracy: 83.95%, model saved.
[Train] Epoch 02 | Loss: 1.1889 | Acc: 63.68% | Time: 202.1s
[Test ] Loss: 0.5546 | Acc: 85.50%
  -> New best accuracy: 85.50%, model saved.
[Train] Epoch 03 | Loss: 1.0829 | Acc: 65.17% | Time: 203.0s
[Test ] Loss: 0.4944 | Acc: 86.23%
  -> New best accuracy: 86.23%, model saved.
[Train] Epoch 04 | Loss: 1.0474 | Acc: 65.87% | Time: 203.3s
[Test ] Loss: 0.4730 | Acc: 86.60%
  -> New best accuracy: 86.60%, model saved.
[Train] Epoch 05 | Loss: 1.0367 | Acc: 65.71% | Time: 203.3s
[Test ] Loss: 0.4685 | Acc: 86.63%
  -> New best accuracy: 86.63%, model saved.

Best test accuracy for faster_vit_1_224: 86.63%
```



FASTERVIT-4-21K-224: HEAD-ONLY FINE-TUNING RESULTS



Best Accuracy: 97.61

	precision	recall	f1-score	support
airplane	0.98	0.97	0.98	1000
automobile	0.98	0.98	0.98	1000
bird	0.99	0.97	0.98	1000
cat	0.95	0.93	0.94	1000
deer	0.98	0.97	0.98	1000
dog	0.94	0.97	0.96	1000
frog	0.98	0.99	0.99	1000
horse	0.99	0.99	0.99	1000
ship	0.98	0.98	0.98	1000
truck	0.98	0.98	0.98	1000
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

This behaviour matches the paper: larger FasterViT models give stronger feature representations and therefore perform better on downstream tasks, even with minimal fine-tuning.

- **Our results demonstrate this same trend and confirm that FasterViT transfers well to new datasets.**
- **Overall, the experiment shows that FasterViT remains effective and scalable, even when reproduced on limited hardware.**

Experiment 02 - Fine-Tuning for Image Classification

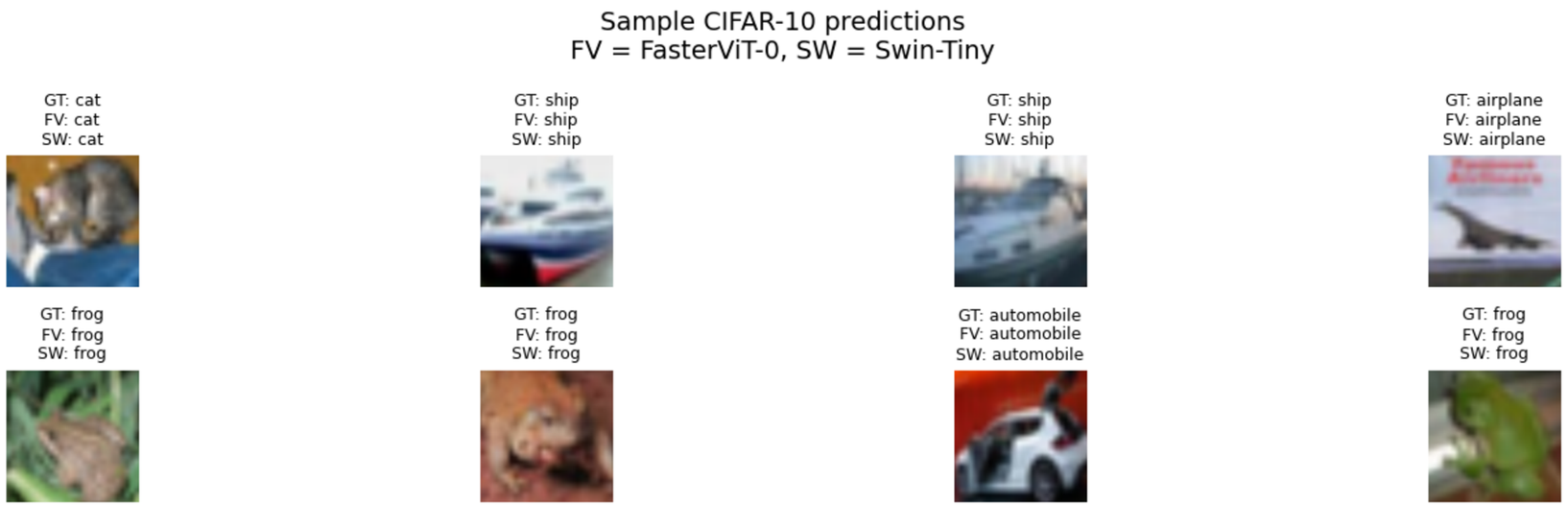
FASTERVIT VS SWIN TRANSFORMER

Accuracy (CIFAR-10 fine-tuning)

```
=== Final Evaluation on CIFAR-10 test set ===
[FasterViT-0 (best) Eval ] Loss=0.0685 | Acc=97.74%
[Swin-Tiny (best) Eval ] Loss=0.0788 | Acc=97.56%
```

```
SUMMARY ACCURACY
FasterViT-0  Test Acc: 97.74%
Swin-Tiny    Test Acc: 97.56%
```

- **Both models reach very high accuracy (>97%).**
- **FasterViT-0 performs slightly better than Swin-Tiny.**
- **FasterViT shows more stable improvements across epochs, especially from Epoch 1–3.**

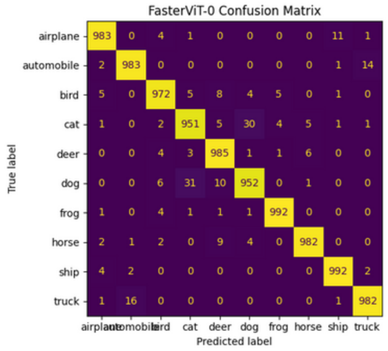


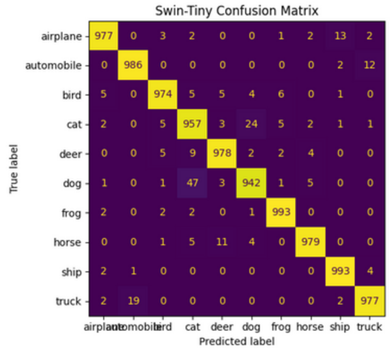
Throughput (224×224 inference speed)

```
=== Throughput Benchmark (224x224) ===
[FasterViT-0] Processed 3200 images in 589.831s
[FasterViT-0] Throughput: 5.43 images/sec
[Swin Tiny] Processed 3200 images in 1131.797s
[Swin Tiny] Throughput: 2.83 images/sec
```

```
=== THROUGHPUT SUMMARY ===
FasterViT-0 throughput: 5.43 img/s
Swin Tiny   throughput: 2.83 img/s
Speedup (FasterViT / Swin): 1.92×
```

- **FasterViT-0 is 1.92× faster than Swin-Tiny.**
- **This aligns almost perfectly with the paper's claim.**





Comparison with the paper

Model	Image Size (Px)	#Param (M)	FLOPs (G)	Throughput (Img/Sec)	Top-1 (%)
ViT-L/16‡ Liu et al. (2021)	384	307.0	190.7	149	85.2
Swin-L‡ Liu et al. (2021)	224	197.0	34.5	787	86.3
Swin-L‡ Liu et al. (2021)	384	197.0	103.9	206	87.3
ConvNeXt-L‡ Liu et al. (2022b)	224	198.0	34.4	508	86.6
ConvNeXt-L‡ Liu et al. (2022b)	384	198.0	101.0	172	87.5
FasterViT-4‡	224	424.6	36.6	**849**	**86.6**
FasterViT-4‡	384	424.6	119.2	**281**	**87.5**

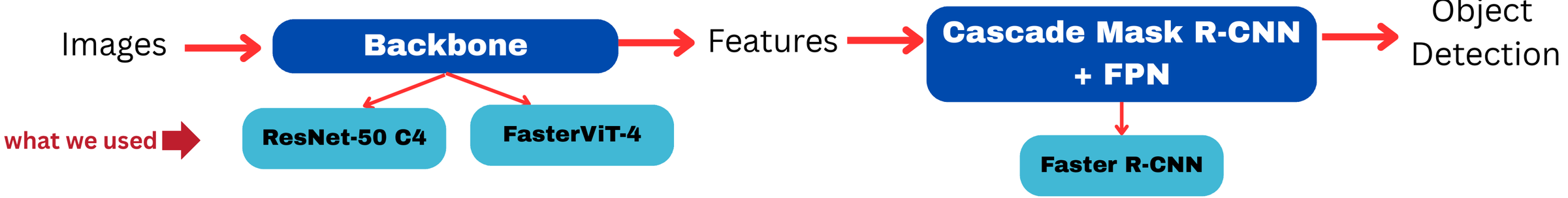
FasterViT is faster

FasterViT has competitive accuracy

Experiment 3 — Evaluating FasterViT as a Detection Backbone

key components:

✓ Tests FasterViT-2/3/4 as backbones inside Cascade Mask R-CNN + FPN. (Uses COCO 2017)

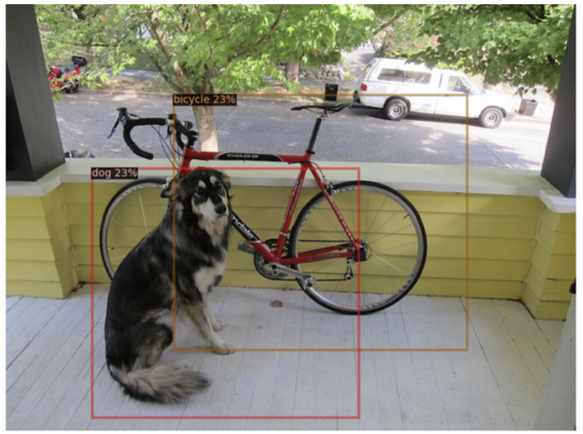


From Paper

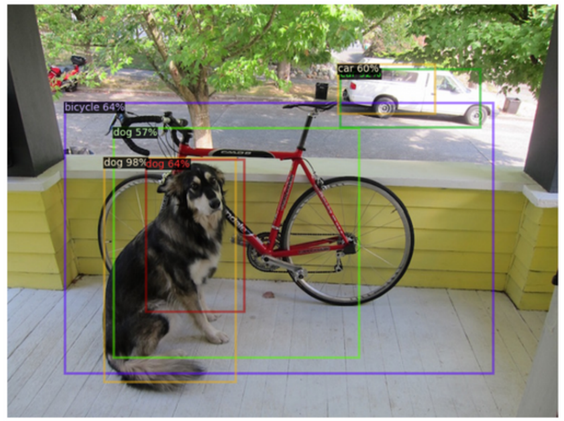
Cascade Mask R-CNN + FPN on COCO

Backbone	Throu. im/sec	AP^{box} Box	50	75
Swin-T Liu et al. (2021)	161	50.4	69.2	54.7
ConvNeXt-T Liu et al. (2022b)	166	50.4	69.1	54.8
DeiT-Small/16 Touvron et al. (2021a)	269	48.0	67.2	51.7
FasterViT-2	**287**	**52.1**	**71.0**	**56.6**
Swin-S Liu et al. (2021)	119	51.9	70.7	56.3
X101-32 Xie et al. (2017)	124	48.1	66.5	52.4
ConvNeXt-S Liu et al. (2022b)	128	51.9	70.8	56.5
FasterViT-3	**159**	**52.4**	**71.1**	**56.7**
X101-64 Xie et al. (2017)	86	48.3	66.4	52.3
Swin-B Liu et al. (2021)	90	51.9	70.5	56.4
ConvNeXt-B Liu et al. (2022b)	101	52.7	71.3	57.2
FasterViT-4	**117**	**52.9**	**71.6**	**57.7**

FasterViT Backbone



ResNet-50 Backbone



Setting	FasterViT C4 Backbone	ResNet-50 C4 Backbone
Dataset	VOC 2007 Test	VOC 2007 Test
Detector	Faster R-CNN (Detectron2)	Faster R-CNN (Detectron2)
Backbone type	FasterViT (transformer-hybrid)	ResNet-50 (convolutional)
Input resolution (inference)	800 × 800 (short edge)	800 × 800 (short edge)
Number of test images / batches	4952	4952
AP (VOC 2007, bbox)	3.293	26.45
AP50	8.67	56.46
AP75	1.707	18.75
Throughput (img/sec)	6.65	8.35

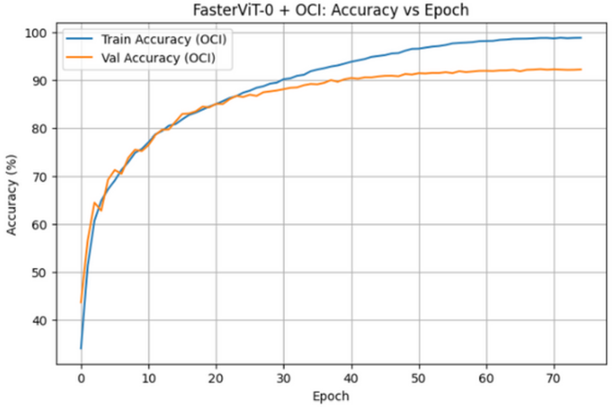
ResNet-50 C4 → better accuracy

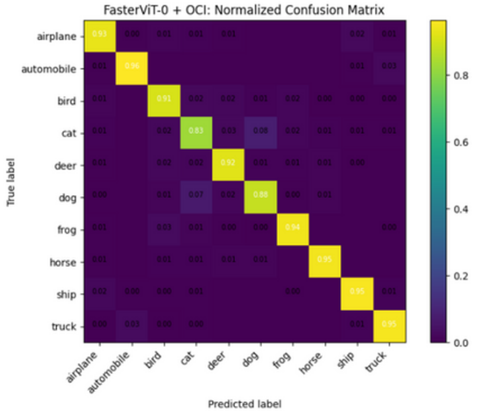
FasterViT C4 → better efficiency

Improvement-Fastervit0+Overlapping windows

FasterVit0+Overlapping windows

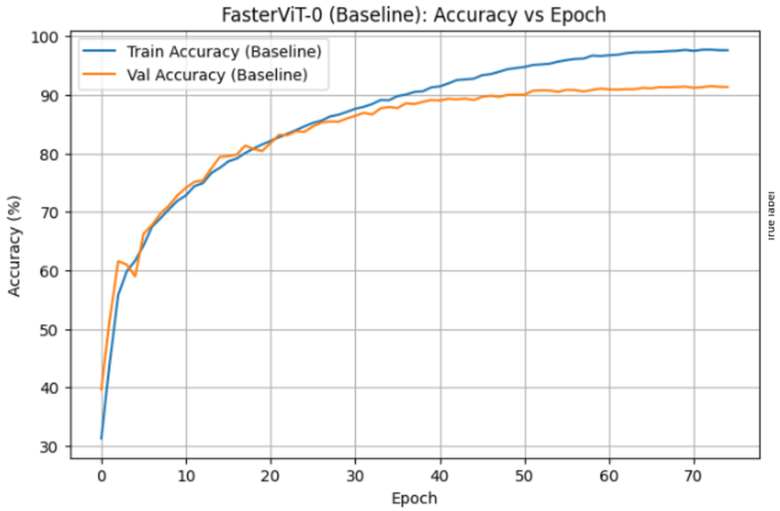
- increased the **Carrier Token initialization kernel from 3×3 to 5×5** by adding padding. (so it can “see” slightly into the neighboring windows)
- This helps the model **capture features at window borders more accurately** while keeping the computational cost almost the same.

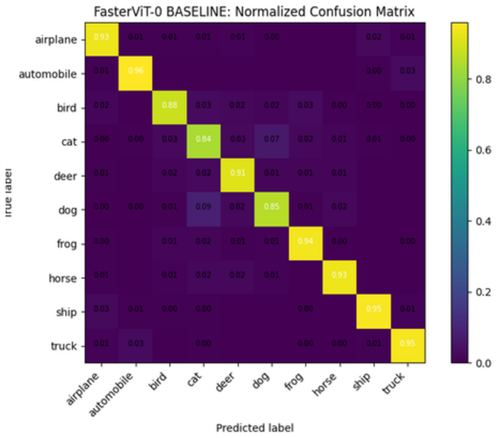




CIFAR-10 (5000+1000)
75 epoch
Accuracy-92.27%| Loss0.3074
Throughput(imgs/sec)-765.92
Latency(ms/image)-27.0320

FasterVit0





CIFAR-10 (5000+1000)
75 epoch

Accuracy-91.47%| Loss0.3184
Throughput(imgs/sec)-790.79
Latency(ms/image)-26.6586

Few more Improvements

- **Lightweight Window Overlap for Local Tokens** :Add small window overlaps to recover boundary information.
- **Dynamic Carrier Token Scaling**: Dynamically scale carrier token influence for more adaptive global communication
- **Hybrid Patch Embedding**: Replacing the basic patch embedding with a stronger patch embedding block to improve early feature extraction.

Goal: Improve accuracy while keeping the model lightweight

SUMMARY

We reproduced the paper results under three experiments and could show that **FasterViT delivers strong accuracy, high throughput, and excellent transfer-learning performance.**

References

- Mehta, S., Kar, A., & Rastegari, M. (2023). FasterViT: Fast Vision Transformers with Hierarchical Attention. arXiv:2306.06189. https://arxiv.org/abs/2306.06189
- Wightman, R. (2019). PyTorch Image Models. https://github.com/rwightman/pytorch-image-models
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database.
- Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/~kriz/cifar.htmlCVPR. https://image-net.org/

THANK YOU