Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model

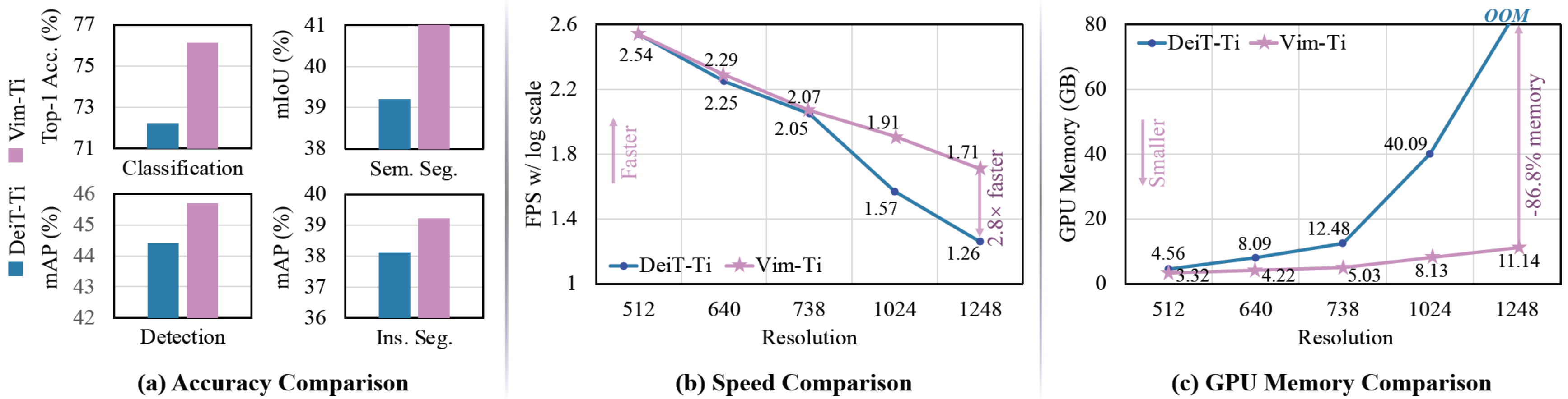
Authors: Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, Xinggang Wang

Alberto Becerra Tomé

Barcelona, April 2024

Motivation

- Introducing Vision Mamba (Vim) with **Bidirectional SSM**
- Improving existing SOTA Transformer based models (DeiT) for high resolution in terms of:
  - **Memory efficiency**
  - **Performance in Vision Tasks**



State Space Models (SSM)

They are inspired in basic 1-D continuous differential models for sequences

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t),$$
$$y(t) = \mathbf{C}h(t).$$

Discretization

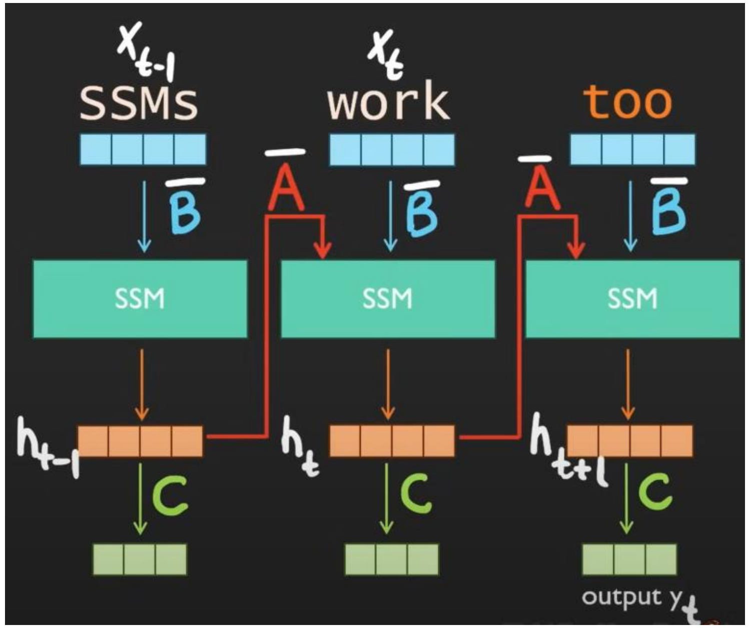
$$\overline{\mathbf{A}} = \exp(\boldsymbol{\Delta}\mathbf{A}),$$
$$\overline{\mathbf{B}} = (\boldsymbol{\Delta}\mathbf{A})^{-1}(\exp(\boldsymbol{\Delta}\mathbf{A}) - \mathbf{I}) \cdot \boldsymbol{\Delta}\mathbf{B}.$$

$$h_t = \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t,$$
$$y_t = \mathbf{C}h_t.$$

Learnable parameters: Step size (Δ), B, C

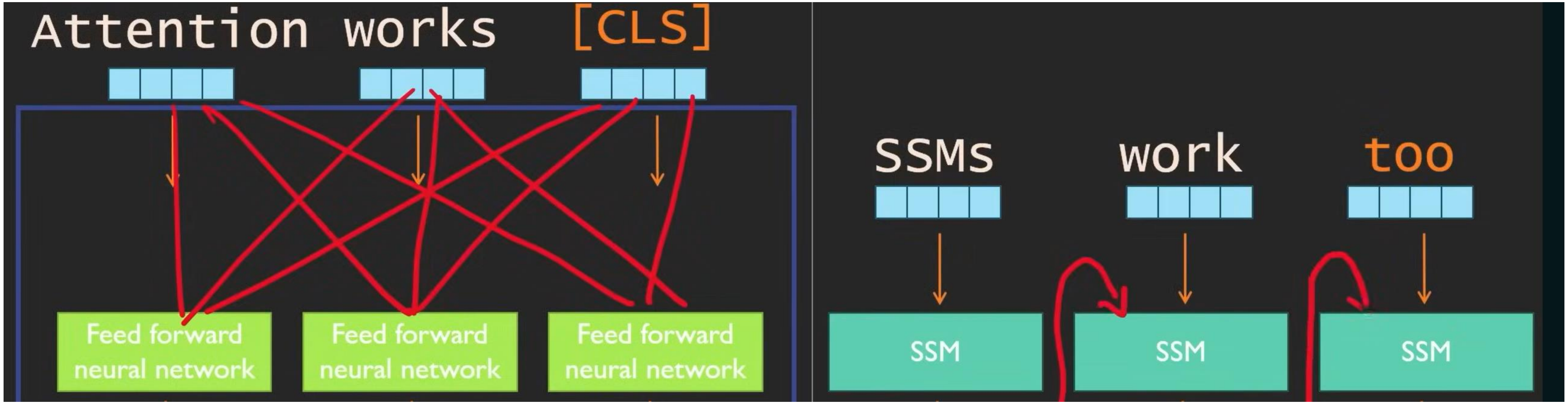
Method: Convolution (Efficient in GPU)

$$\overline{\mathbf{K}} = (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{AB}}, \dots, \mathbf{C}\overline{\mathbf{A}}^{\mathrm{M}-1}\overline{\mathbf{B}}),$$
$$\mathbf{y} = \mathbf{x} * \overline{\mathbf{K}},$$



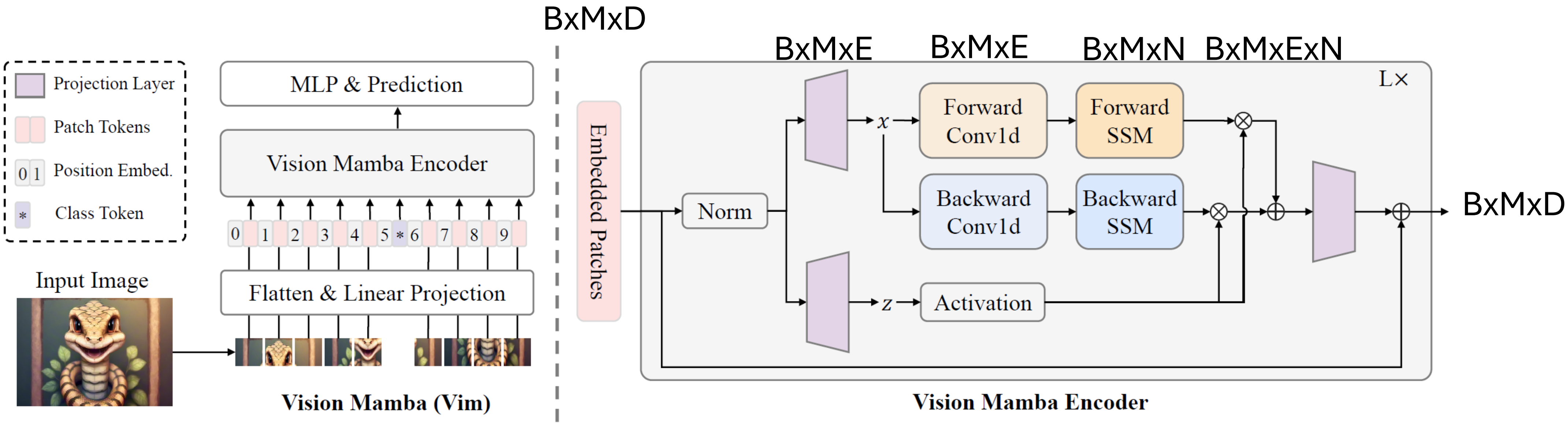
*Images: AICoffeeBreak, (2024, April 08). *MAMBA and State Space Models explained | SSM explained* [Video]. YouTube. URL: https://www.youtube.com/watch?v=vrF3MtGwD0Y&t=1s&ab_channel=AICoffeeBreakwithLetitia

SSMs vs. Transformers (Efficiency)



*Images: AICoffeeBreak, (2024, April 08). *MAMBA and State Space Models explained | SSM explained* [Video]. YouTube.
URL: https://www.youtube.com/watch?v=vrF3MtGwD0Y&t=1s&ab_channel=AICoffeeBreakwithLetitia

Vision Mamba



$$\mathbf{T}_0 = [\mathbf{t}_{cls}; \mathbf{t}_p^1\mathbf{W}; \mathbf{t}_p^2\mathbf{W}; \cdots; \mathbf{t}_p^J\mathbf{W}] + \mathbf{E}_{pos},$$

$$\mathbf{T}_l = \mathbf{Vim}(\mathbf{T}_{l-1}) + \mathbf{T}_{l-1},$$

$$\mathbf{f} = \mathbf{Norm}(\mathbf{T}_L^0),$$

$$\hat{p} = \mathbf{MLP}(\mathbf{f}),$$

	Tiny	Small
L : Number of vim blocks	24	24
D: Hidden state dimension	192	384
E: Expanded state dimension	384	768
N: SSM dimensión	16	16

Experiments: Classification

Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

Table 1. Comparison with different backbones on ImageNet-1K validation set. † represents the model is fine-tuned with our long sequence setting.

ImageNet-1K Dataset:

- **1.28M training** images
- **50K validation** images
- 1,000 categories

Long Sequence Fine-Tuning: Double of patches than DeiT with the same size (stride 8, 16x16).

Results:

Experiments: Classification

Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

+3.9

Table 1. Comparison with different backbones on ImageNet-1K validation set. † represents the model is fine-tuned with our long sequence setting.

ImageNet-1K Dataset:

- **1.28M training** images
- **50K validation** images
- 1,000 categories

Long Sequence Fine-Tuning: Double of patches than DeiT with the same size (stride 8, 16x16).

Results:

- **3.9 points** higher for **Vim-Tiny** over **DeiT-Tiny**

Experiments: Classification

Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

+0.7

Table 1. Comparison with different backbones on ImageNet-1K validation set. † represents the model is fine-tuned with our long sequence setting.

ImageNet-1K Dataset:

- **1.28M training** images
- **50K validation** images
- 1,000 categories

Long Sequence Fine-Tuning: Double of patches than DeiT with the same size (stride 8, 16x16).

Results:

- **3.9 points** higher for **Vim-Tiny** over **DeiT-Tiny**
- **0.7 points** higher for **Vim-Small** over **DeiT-Small**

Experiments: Classification

Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

Table 1. Comparison with different backbones on ImageNet-1K validation set. † represents the model is fine-tuned with our long sequence setting.

ImageNet-1K Dataset:

- **1.28M training** images
- **50K validation** images
- 1,000 categories

Long Sequence Fine-Tuning: Double of patches than DeiT with the same size (stride 8, 16x16).

Results:

- **3.9 points** higher for **Vim-Tiny** over **DeiT-Tiny**
- **0.7 points** higher for **Vim-Small** over **DeiT-Small**
- **Vim-S** achieves results **similar to DeiT-B** with LSFT

Experiments: Classification

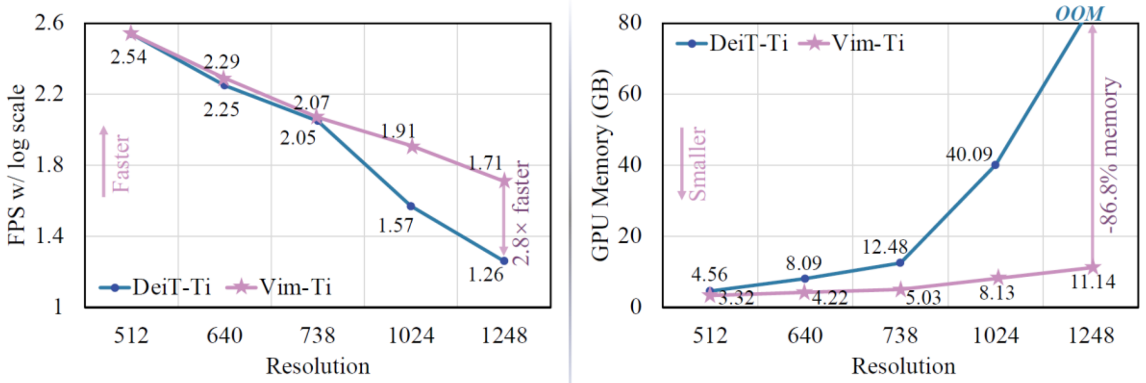
Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

Table 1. Comparison with different backbones on ImageNet-1K validation set. † represents the model is fine-tuned with our long sequence setting.

ImageNet-1K Dataset:
- **1.28M training** images
- **50K validation** images
- 1,000 categories

Long Sequence Fine-Tuning: Double of patches than DeiT with the same size (stride 8, 16x16).

Results:
- **3.9 points** higher for **Vim-Tiny** over **DeiT-Tiny**
- **0.7 points** higher for **Vim-Small** over **DeiT-Small**
- **Vim-S** achieves results **similar to DeiT-B** with LSFT
- **1248×1248: Vim** is **2.8× faster** than DeiT and **saves 86.8% GPU memory** in batch inference



Experiments: Semantic Segmentation

Method	Backbone	image size	#param.	*val* mIoU
DeepLab v3+	ResNet-101	512^2	63M	44.1
UperNet	ResNet-50	512^2	67M	41.2
UperNet	ResNet-101	512^2	86M	44.9
UperNet	DeiT-Ti	512^2	11M	39.2
UperNet	DeiT-S	512^2	43M	44.0
UperNet	Vim-Ti	512^2	13M	41.0
UperNet	Vim-S	512^2	46M	44.9

Table 2. Results of semantic segmentation on the ADE20K *val* set.

ADE20K Dataset:

- **20K training** images
- **2K validation** images
- 150 categories
- **UperNet** framework

Results:

Experiments: Semantic Segmentation

Method	Backbone	image size	#param.	*val* mIoU
DeepLab v3+	ResNet-101	512^2	63M	44.1
UperNet	ResNet-50	512^2	67M	41.2
UperNet	ResNet-101	512^2	86M	44.9
UperNet	DeiT-Ti	512^2	11M	39.2
UperNet	DeiT-S	512^2	43M	44.0
UperNet	Vim-Ti	512^2	13M	41.0
UperNet	Vim-S	512^2	46M	44.9

+1.8

Table 2. Results of semantic segmentation on the ADE20K *val* set.

ADE20K Dataset:

- **20K training** images
- **2K validation** images
- 150 categories
- **UperNet** framework

Results:

- **1.8 mIoU** higher for **Vim-Ti** over **DeiT-Ti**

Experiments: Semantic Segmentation

Method	Backbone	image size	#param.	*val* mIoU
DeepLab v3+	ResNet-101	512^2	63M	44.1
UperNet	ResNet-50	512^2	67M	41.2
UperNet	ResNet-101	512^2	86M	44.9
UperNet	DeiT-Ti	512^2	11M	39.2
UperNet	DeiT-S	512^2	43M	44.0
UperNet	Vim-Ti	512^2	13M	41.0
UperNet	Vim-S	512^2	46M	44.9

+0.9

Table 2. Results of semantic segmentation on the ADE20K *val* set.

ADE20K Dataset:

- **20K training** images
- **2K validation** images
- 150 categories
- **UperNet** framework

Results:

- **1.8 mIoU** higher for **Vim-Ti** over **DeiT-Ti**
- **0.9 mIoU** higher for **Vim-S** over **DeiT-S**

Experiments: Semantic Segmentation

Method	Backbone	image size	#param.	*val* mIoU
DeepLab v3+	ResNet-101	512^2	63M	44.1
UperNet	ResNet-50	512^2	67M	41.2
UperNet	ResNet-101	512^2	86M	44.9
UperNet	DeiT-Ti	512^2	11M	39.2
UperNet	DeiT-S	512^2	43M	44.0
UperNet	Vim-Ti	512^2	13M	41.0
UperNet	Vim-S	512^2	46M	44.9

Table 2. Results of semantic segmentation on the ADE20K *val* set.

ADE20K Dataset:

- **20K training** images
- **2K validation** images
- 150 categories
- **UperNet** framework

Results:

- **1.8 mIoU** higher for **Vim-Ti** over **DeiT-Ti**
- **0.9 mIoU** higher for **Vim-S** over **DeiT-S**
- **Vim-S similar to ResNet-101** but **2x fewer parameters**

Experiments: Object Detection and Instance Segmentation

Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{s}	AP^{box}_{m}	AP^{box}_{l}
DeiT-Ti	44.4	63.0	47.8	26.1	47.4	61.8
Vim-Ti	45.7	63.9	49.6	26.1	49.0	63.2
Backbone	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{s}	AP^{mask}_{m}	AP^{mask}_{l}
DeiT-Ti	38.1	59.9	40.5	18.1	40.5	58.4
Vim-Ti	39.2	60.9	41.7	18.2	41.8	60.2

Table 3. Results of object detection and instance segmentation on the COCO *val* set using Cascade Mask R-CNN [4] framework.

COCO 2017 Dataset:

- **118K training** images
- **5K validation** images
- Cascade Mask R-CNN base framework

Results:

Experiments: Object Detection and Instance Segmentation

Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{s}	AP^{box}_{m}	AP^{box}_{l}
DeiT-Ti	44.4	63.0	47.8	26.1	47.4	61.8
Vim-Ti	45.7	63.9	49.6	26.1	49.0	63.2
Backbone	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{s}	AP^{mask}_{m}	AP^{mask}_{l}
DeiT-Ti	38.1	59.9	40.5	18.1	40.5	58.4
Vim-Ti	39.2	60.9	41.7	18.2	41.8	60.2

Table 3. Results of object detection and instance segmentation on the COCO *val* set using Cascade Mask R-CNN [4] framework.

COCO 2017 Dataset:

- **118K training** images
- **5K validation** images
- Cascade Mask R-CNN base framework

Results:

- Vim-Ti **surpasses** DeiT-Ti for **medium-size** and **big** objects, demonstrating **better long-range context learning**

Experiments: Object Detection and Instance Segmentation

Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{s}	AP^{box}_{m}	AP^{box}_{l}
DeiT-Ti	44.4	63.0	47.8	26.1	47.4	61.8
Vim-Ti	45.7	63.9	49.6	26.1	49.0	63.2
Backbone	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{s}	AP^{mask}_{m}	AP^{mask}_{l}
DeiT-Ti	38.1	59.9	40.5	18.1	40.5	58.4
Vim-Ti	39.2	60.9	41.7	18.2	41.8	60.2

Table 3. Results of object detection and instance segmentation on the COCO *val* set using Cascade Mask R-CNN [4] framework.

COCO 2017 Dataset:

- **118K training** images
- **5K validation** images
- Cascade Mask R-CNN base framework

Results:

- Vim-Ti **surpasses** DeiT-Ti for **medium-size** and **big** objects, demonstrating **better long-range context learning**
- Not necessary window attention

Conclusions

- **Computational complexity** linear on sequence length as shown for text
- **Modeling power** similar to DeiT and superior for higher resolution images thanks to efficient long sequences management
- Possible **alternative to Transformer** based backbones

Future Lines:

- Broader Exploration. Running on different Datasets and Frameworks
- Self-Supervised Learning
- Comparison of improvements for SOTA systems based on Transformers
- As with Transformer architecture, opening a path to explore nexxt-generation AI based applications.

References

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., & Wang, X. (2024). **Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model**. arXiv preprint arXiv:2401.09417 [cs.CV].

Gu, A., & Dao, T. (2023). **Mamba: Linear-Time Sequence Modeling with Selective State Spaces**. arXiv preprint arXiv:2312.00752 [cs.LG].

Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., & Qiao, Y. (Year). **VideoMamba: State Space Model for Efficient Video Understanding**. arXiv preprint arXiv:2403.06977 [cs.CV].

AICoffeeBreak, (2024, April 08). ***MAMBA and State Space Models explained | SSM explained*** [Video]. YouTube. URL: https://rb.gy/phwzer