

AI VIET NAM – AIO Course

# Mamba for Vision

Minh-Duc Bui và Quang-Vinh Dinh

*PR-Team: Đăng-Nhã Nguyễn, Minh-Châu Phạm và Hoàng-Nguyễn Vũ*

Ngày 23 tháng 2 năm 2024



## 1 Introduction

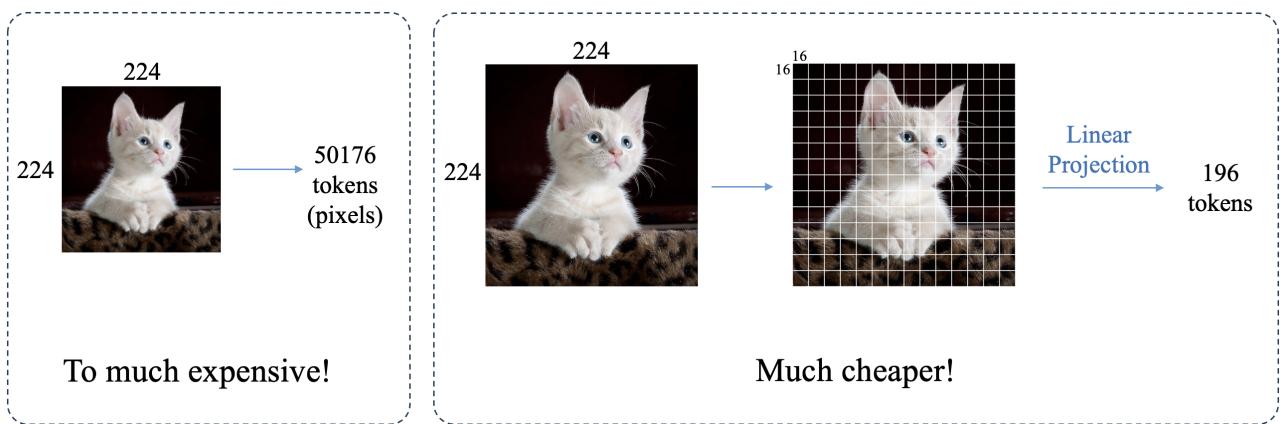
**Mamba**, tên gọi khác là S6 (Structured State Space for Sequence Modeling with Selective Scan), là một kiến trúc mới xuất hiện gần đây. Điểm khác biệt chính giữa Mamba và các mô hình hiện tại như Transformer là Mamba không sử dụng cơ chế attention. Nhờ vậy, Mamba có độ phức tạp thấp hơn nhiều so với Transformer, chỉ  $O(n)$  so với  $O(n^2)$  của Transformer (với  $n$  là chiều dài sequence). Bên cạnh đó, Mamba cho độ chính xác cao khi chiều dài sequence tăng (lên đến hàng triệu).

Trong bài viết này, ta sẽ tìm hiểu cách các nhà nghiên cứu đã áp dụng kiến trúc Mamba vào data dạng ảnh, thông qua các bài paper sau:

Title	Link	Github
Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model	<a href="#">here</a>	<a href="#">here</a>
VMamba: Visual State Space Model	<a href="#">here</a>	<a href="#">here</a>
U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation	<a href="#">here</a>	<a href="#">here</a>
Swin-UMamba: Mamba-based UNet with ImageNet-based pretraining	<a href="#">here</a>	<a href="#">here</a>

## 2 Text vs. Image

Điểm khác biệt chính giữa text và image chính là thông tin về dimension, text chỉ có 1D trong khi image là 2D. Điều này gây ra khó khăn khi áp dụng các model từ NLP (ví dụ Mamba, Transformer) sang CV.

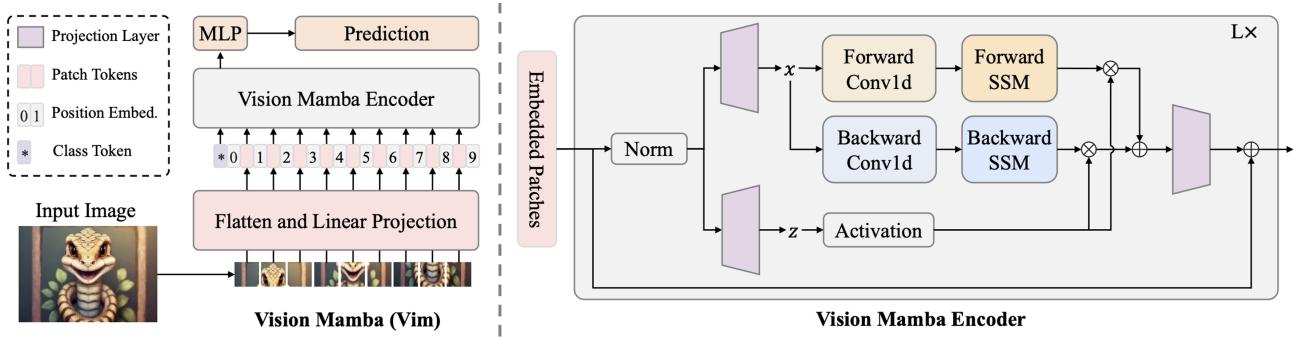


Hình 1: Pixel-level và Patch-level khi sử dụng mô hình Transformer cho data dạng ảnh.

Hình 1 mô tả 2 cách chuyển đổi từ data 2D sang data 1D. Nếu ta chia ảnh đầu vào theo pixel-level thì với ví dụ như hình (ảnh 224x224) ta sẽ có hơn 50k token, nếu ánh xạ tương ứng sang text là 1 câu có 50k từ. Điều này cực kì khó khăn đối với các GPU để tính toán. Cách phổ biến được sử dụng đối với Transformer chính là chia ảnh đầu vào thành các patch nhỏ (ví dụ 16x16 pixel), ảnh có kích thước 224x224 sau khi chia sẽ có tổng cộng  $14 \times 14 = 196$  patch (hoặc token). 196 token là hoàn toàn hợp lý so với 50k và các GPU vẫn có thể hoạt động bình thường.

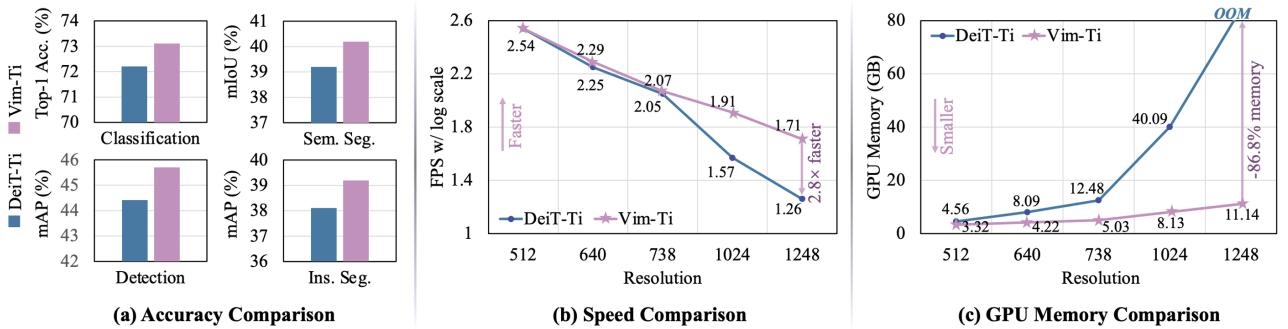
### 3 Vision Mamba

Vision Mamba (Vim) là bài paper đầu tiên áp dụng Mamba vào data dạng ảnh. Các tác giả đã thay thế Transformer Encoder trong Vision Transformer (ViT) bằng **Vision Mamba Encoder**, và thay đổi kiến trúc SSM thành **Bidirectional SSM** (tương tự Bidirectional LSTM). Kiến trúc của Vim và Vision Mamba Encoder được mô tả như hình 2. Từng Vision Mamba Encoder block sẽ có 2 block SSM là **Forward SSM** và **Backward SSM**. Giả sử ta đánh số các token theo thứ tự từ 1 đến  $N$ , thì Forward SSM sẽ tính SSM theo chiều thuận từ 1 đến  $N$ , Backward SSM sẽ tính theo chiều ngược lại từ  $N$  đến 1. Tương tự trong data dạng text, Forward là từ đầu câu đến cuối câu, và Backward là từ cuối câu đến đầu câu. Việc tính toán như vậy giúp model học được thông tin từ 2 hướng khác nhau.



Hình 2: Kiến trúc Vision Mamba (Vim) và Vision Mamba Encoder.

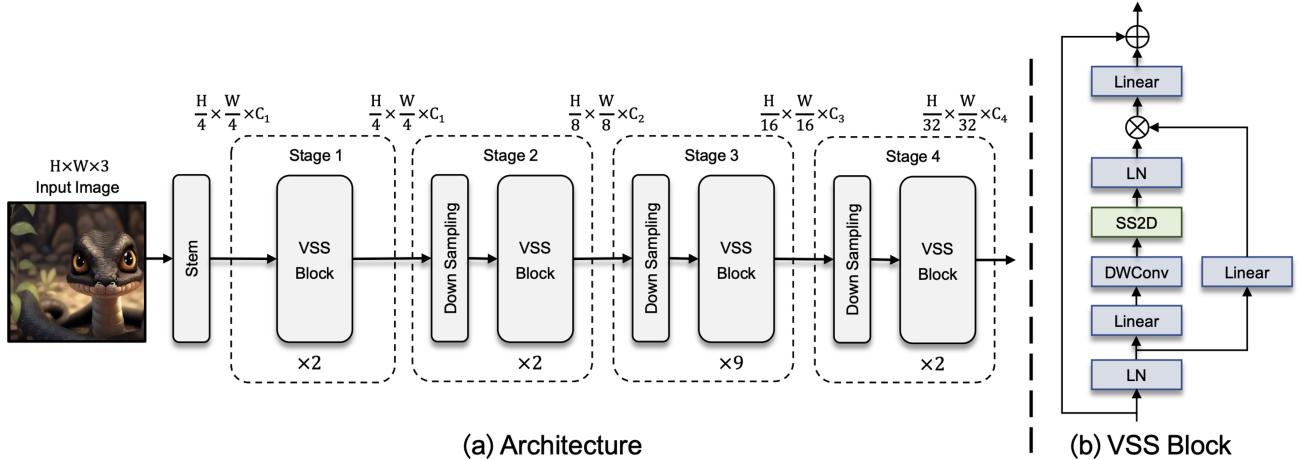
Hình 3 mô tả kết quả giữa Transformer (DeiT) và Vision Mamba (Vim). Ta thấy, Vim vượt trội hơn DeiT ở tất cả các task bao gồm: image classification, object detection, và instance segmentation. Hơn nữa, khi kích thước ảnh đầu vào tăng, Vim outperform DeiT ở cả speed và memory. Cụ thể, Vim nhanh hơn và tiết kiệm bộ nhớ hơn DeiT lần lượt là  $2.8\times$  và  $86.8\%$ . Khi công nghệ ngày càng phát triển, kích thước ảnh  $224\times 224 \rightarrow 512\times 512$  dần trở nên lỗi thời, con người luôn muốn sử dụng ảnh có kích thước lớn hơn để các model có thể đạt độ chính xác cao hơn. Điều này càng chứng tỏ Mamba sẽ có khả năng cao trở thành backbone được sử dụng trong tương lai.



Hình 3: Kết quả so sánh giữa Transformer (DeiT) và Vision Mamba.

## 4 Visual Mamba

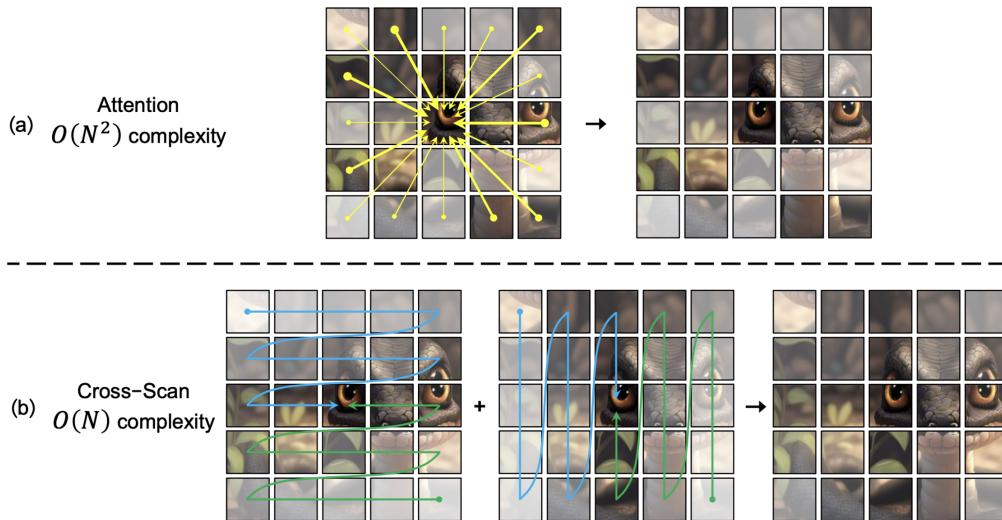
Sau khi Vim được công bố chỉ 1 ngày thì ta tiếp tục có bài paper thứ 2 sử dụng Mamba vào data dạng ảnh mang tên Visual Mamba (VMamba). Xuất phát từ Swin Transformer (một model cải tiến của ViT) các tác giả đã thay thế Transformer block bằng Visual State Space (VSS) block (hình 4).



Hình 4: Kiến trúc VMamba và VSS Block.

Khác với Vim sử dụng Bidirectional, VMamba sử dụng Four-Directional (4 hướng) để tính SSM và sử dụng tên gọi Cross-Scan. 4 hướng này bao gồm:

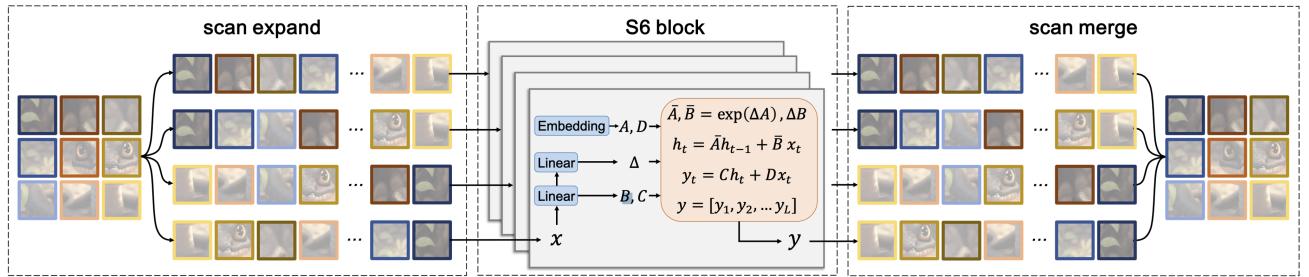
1. trái->phải->trên->dưới
2. phải->trái->dưới->trên
3. trên->dưới->trái->phải
4. dưới->trên->phải->trái



Hình 5: So sánh giữa Attention (hình a) và Cross-Scan (hình b).

Hình 5 mô tả cách hoạt động của Cross-Scan, và hình 6 mô tả cách tính toán với Cross-Scan trong SSM block. Cross-Scan giúp VMamba đạt được độ phức tạp linear  $O(n)$  mà không bị mất bất kỳ thông tin global nào. Ví dụ, token ở vị trí chính giữa trong hình 5 được tổng hợp thông tin từ tất cả các token khác theo nhiều hướng khác nhau. Cross-Scan được tính toán thông qua 3 bước chính:

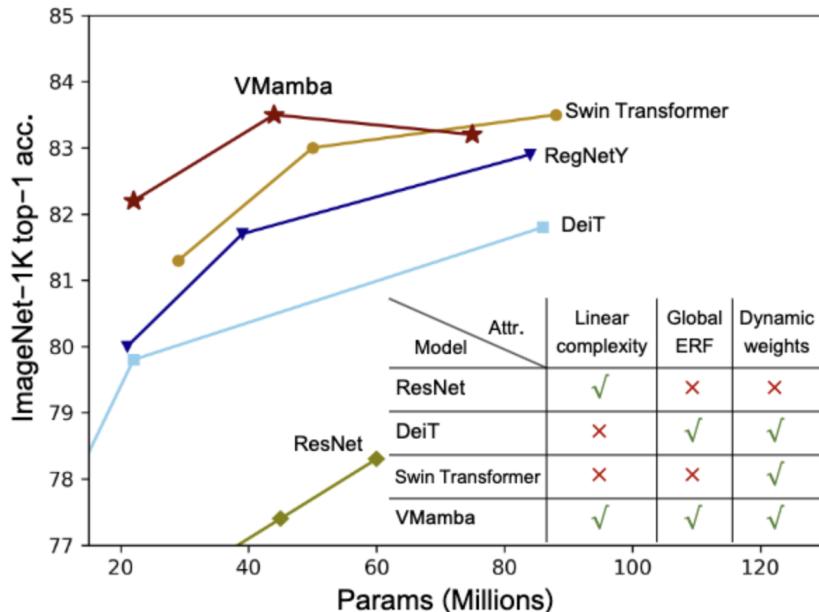
1. **Scan expand:** các token đầu vào được chia thành 4 sequence khác nhau (theo 4 hướng).
2. **S6 block:** 4 sequence được đưa vào S6 block để tính toán.
3. **Scan merge:** cuối cùng, 4 sequence sẽ được tổng hợp lại để tiếp tục expand ở block tiếp theo.



Hình 6: Cách tính toán với Cross-Scan.

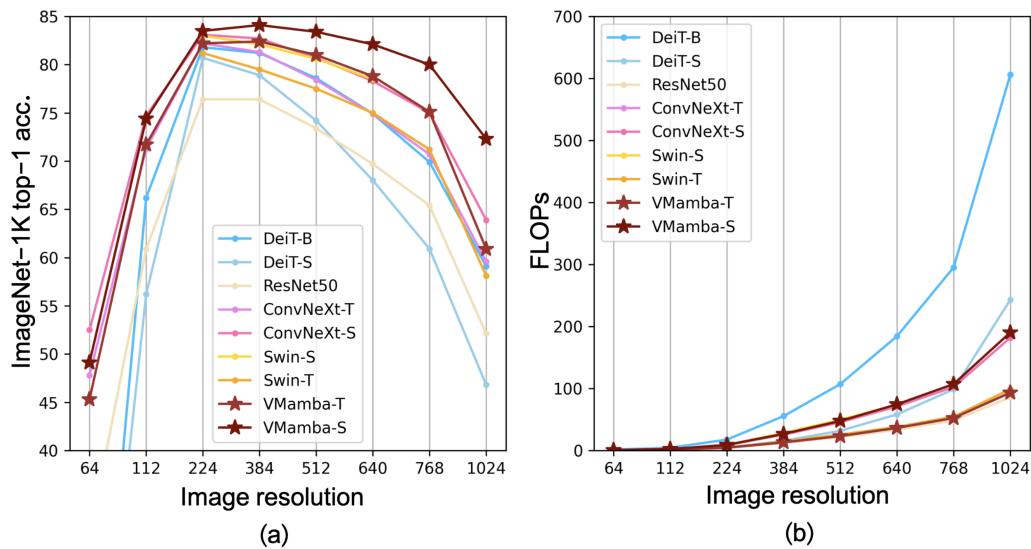
Hình 7 so sánh kết quả giữa VMamba và các model Transformer, CNN khác. Ta thấy VMamba là model sở hữu cả 3 tính chất là:

- Độ phức tạp  $O(n)$  (Linear Complexity).
- Tổng hợp thông tin global (Global ERF).
- Tạo ra param phụ thuộc vào input (Dynamic weights).



Hình 7: So sánh VMamba và các model Transformer, CNN khác.

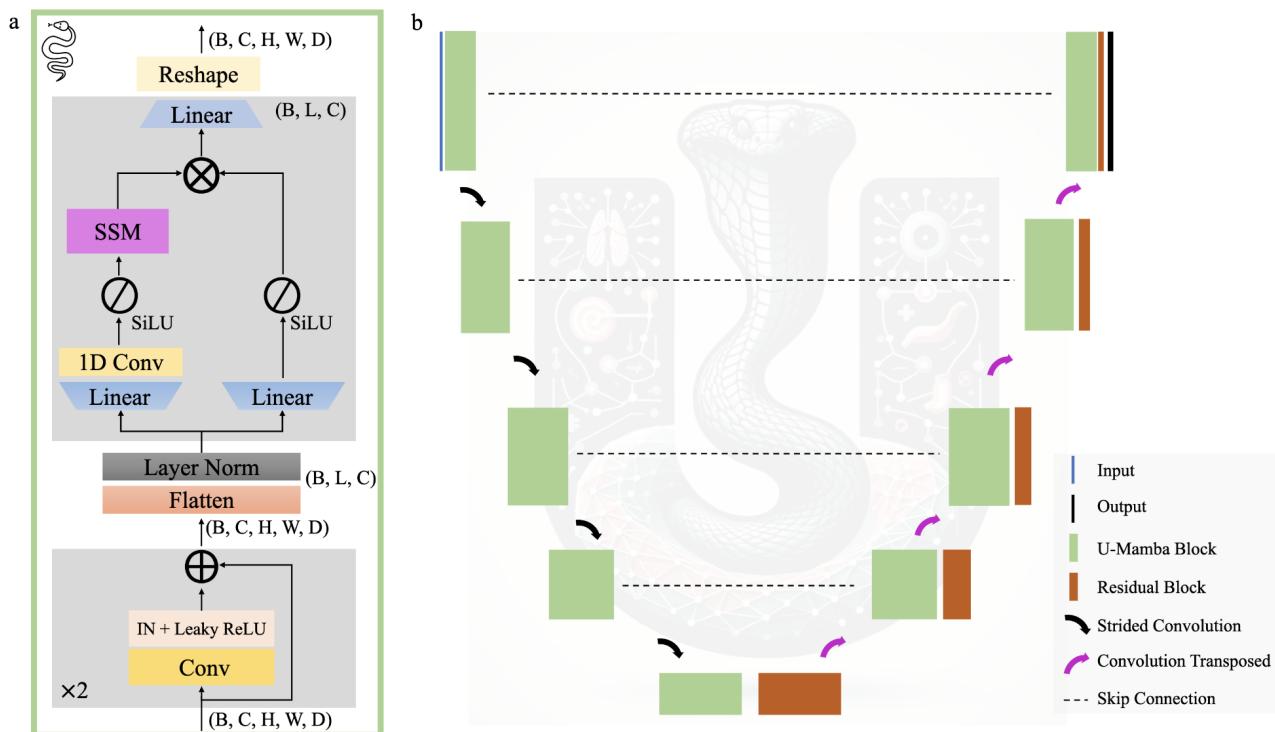
Tương tự Vim, các tác giả của VMamba cũng đánh giá backbone Mamba dưới nhiều size ảnh khác nhau (tham khảo hình 8). Ta thấy, VMamba vượt trội hơn hầu hết các model khác khi kích thước ảnh tăng dần.



Hình 8: So sánh VMamba và các model Transformer và CNN khác dưới nhiều size ảnh khác nhau.

## 5 U-Mamba

Hơn 1 tháng sau khi Mamba được công bố, các tác giả tại Đại học Toronto, Canada đã hướng ứng trend này với bài paper: “U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation”, một bài paper về bài toán Segmentation trong lĩnh vực Y Sinh. U-Mamba là model theo hướng encoder-decoder tương tự U-Net, nhưng các block nhỏ trong encoder và decoder được thay bằng CNN và SSM block (các tác giả gọi là U-Mamba block). Kiến trúc U-Mamba block và U-Mamba hoàn chỉnh được mô tả như hình 9.



Hình 9: Kiến trúc U-Mamba block và model U-Mamba hoàn chỉnh.

Sự kết hợp giữa CNN và SSM tạo nên hybrid CNN-SSM model, model hybrid này vừa có khả năng tổng hợp thông tin local nhờ vào CNN và thông tin global nhờ vào SSM. Từ đó giúp model tổng hợp được 2 khía cạnh khác nhau trong ảnh đầu vào.

U-Mamba block hoạt động bằng cách đưa input đầu vào sang CNN block, sau đó output từ CNN block này sẽ được đưa vào SSM block. Tùy thuộc vào CNN block thuộc phần encoder hay decoder thì sẽ được thiết kế khác nhau. Đối với encoder, CNN block sẽ có tác dụng giảm kích thước feature map và tăng chiều sâu, ngược lại đối với decoder, CNN làm nhiệm vụ tăng kích thước feature map và giảm chiều sâu.

**Bàn luận về hybrid model:** Các model CNN thuần túy (AlexNet, Resnet, VGG,...) là những model chỉ dùng CNN để trích xuất thông tin, nói cách khác chỉ tổng hợp được thông tin local. Ngược lại, các model Transformer (ViT) chỉ trích xuất thông tin global. Và Transformer cho performance vượt trội hơn CNN. Nhưng không vì thế mà ta khẳng định rằng thông tin global luôn luôn tốt hơn local, dẫn tới tình huống chỉ tập trung vào global. Đây là một hiểu lầm phổ biến, cả thông tin local và global đều có những tính chất khác nhau, model mang cả 2 thông tin này sẽ cho performance cao hơn model chỉ mang 1 trong 2. Ví dụ một số paper về Transformer sử dụng thông tin hybrid:

- Neighborhood Attention Transformer
- Dilated Neighborhood Attention Transformer
- MaxViT: Multi-Axis Vision Transformer

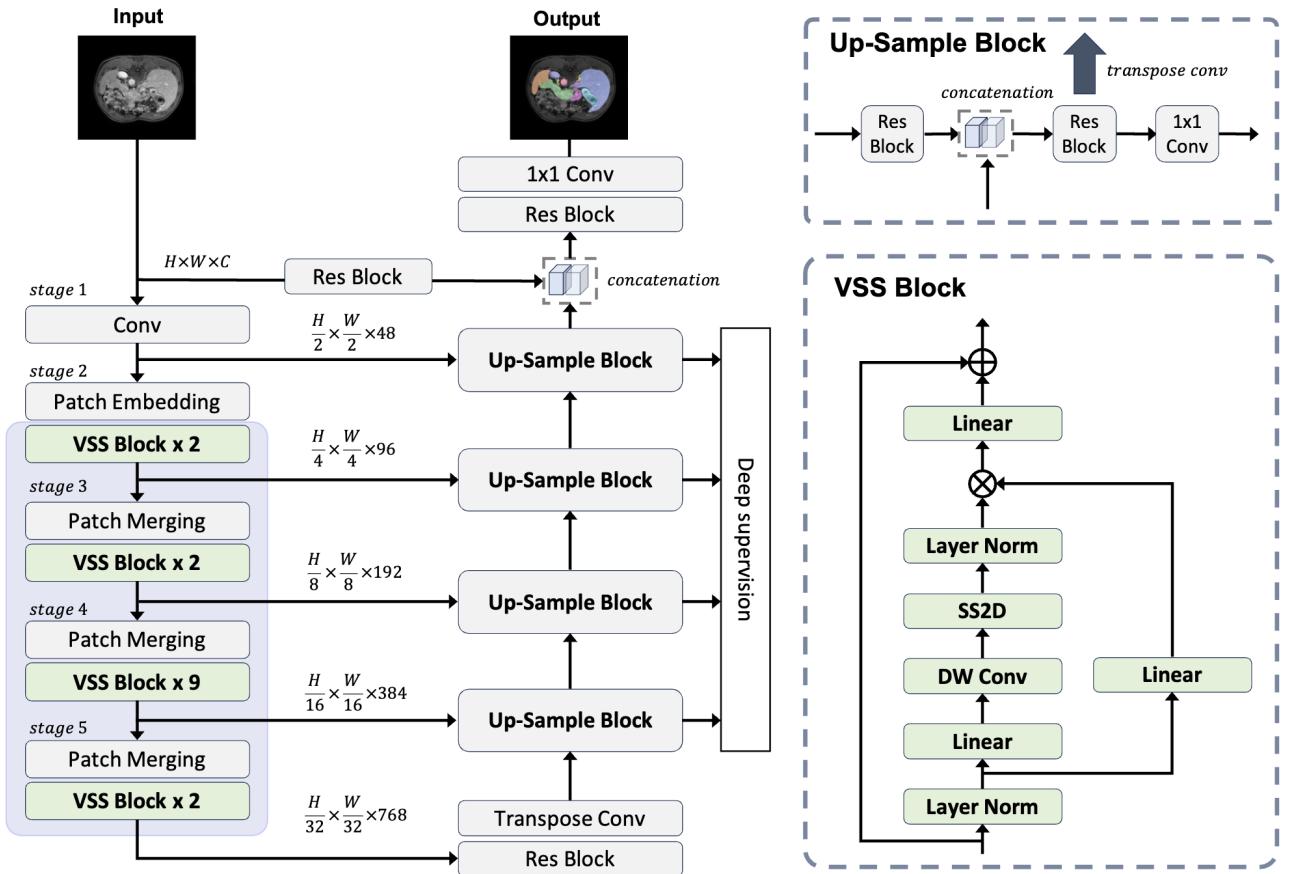
Hình 10 mô tả kết quả so sánh giữa U-Mamba và các model CNN, Transformer khác. Ta thấy U-Mamba vượt trội hoàn toàn so với các model này, khoảng cách rất lớn  $\approx 10\%$ .

Methods	Organs in Abdomen CT		Organs in Abdomen MRI	
	DSC	NSD	DSC	NSD
nnU-Net	0.8615 $\pm$ 0.0790	0.8972 $\pm$ 0.0824	0.8309 $\pm$ 0.0769	0.8996 $\pm$ 0.0729
SegResNet	0.7927 $\pm$ 0.1162	0.8257 $\pm$ 0.1194	0.8146 $\pm$ 0.0959	0.8841 $\pm$ 0.0917
UNETR	0.6824 $\pm$ 0.1506	0.7004 $\pm$ 0.1577	0.6867 $\pm$ 0.1488	0.7440 $\pm$ 0.1627
SwinUNETR	0.7594 $\pm$ 0.1095	0.7663 $\pm$ 0.1190	0.7565 $\pm$ 0.1394	0.8218 $\pm$ 0.1409
U-Mamba_Bot	<b>0.8683<math>\pm</math>0.0808</b>	<b>0.9049<math>\pm</math>0.0821</b>	<b>0.8453<math>\pm</math>0.0673</b>	<b>0.9121<math>\pm</math>0.0634</b>
U-Mamba_Enc	<b>0.8638<math>\pm</math>0.0908</b>	<b>0.8980<math>\pm</math>0.0921</b>	<b>0.8501<math>\pm</math>0.0732</b>	<b>0.9171<math>\pm</math>0.0689</b>

Hình 10: Bảng so sánh kết quả U-Mamba và các model khác trên dataset ảnh 2D.

## 6 Swin-UMamba

Ngay sau khi U-Mamba được công bố được chưa đầy 1 tháng thì các tác giả tại Chinese Academy of Sciences, Peng Cheng Laboratory, và một số phòng Lab khác đã tiếp tục cải tiến U-Mamba và công bố Swin-UMamba. Swin-UMamba là sự kết hợp giữa việc tận dụng pretrained model (VMamba ở phần 2) và kết hợp với kiến trúc U-Net để giải quyết bài toán **Medical image segmentation**. Kiến trúc Swin-UMamba được mô tả như hình 11.



Hình 11: Kiến trúc Swin-UMamba.

Nhìn vào hình 11 ta thấy phần Encoder được sử dụng theo kiến trúc Swin Transformer với các Visual State Space (VSS) block tương tự bài paper Visual Mamba (VMamba). Chính vì tận dụng kiến trúc Encoder như VMamba như thế, ta có thể tận dụng pretrained VMamba để khởi tạo. Đối với phần Decoder, các tác giả đã thiết kế đơn giản như những model U-Net truyền thống: Transpose Convolution + Skip-Connection.

Hình 12 mô tả kết quả của Swin-UMamba và các model Transformer-, Mamba-based khác. Ta thấy khi sử dụng pretrained model từ VMamba thì performance đã tăng từ 4->6%. Swin-UMamba dù sử dụng ít số lượng parameter hơn so với U-Mamba  $\approx 30 \rightarrow 40\%$  nhưng vẫn cho kết quả cao hơn. Về tổng quát, Swin-UMamba vượt trội hoàn toàn so với các model CNN, Transformer, và các model Mamba trước đó.

Methods	#param	FLOPs	DSC	NSD
<i>CNN-based</i>				
nnU-Net	33M	23.3G	0.7450	0.8153
SegResNet	6M	24.5G	0.7317	0.8034
<i>Transformer-based</i>				
UNETR	87M	42.1G	0.5747	0.6309
SwinUNETR	25M	27.9G	0.7028	0.7669
nnFormer	60M	50.2G	0.7279	0.7963
<i>Mamba-based</i>				
U-Mamba_Bot	63M	45.7G	0.7588	0.8285
U-Mamba_Enc	67M	47.9G	0.7625	0.8327
<i>w/o ImageNet-based pretraining</i>				
Swin-UMamba†*	27M	15.0G	0.6653	0.7312
Swin-UMamba	40M	58.4G	0.7464	0.8023
<i>w/ ImageNet-based pretraining</i>				
Swin-UMamba†	27M	15.0G	0.7705	0.8376
Swin-UMamba	40M	58.4G	<b>0.7768</b>	<b>0.8442</b>

Hình 12: Bảng so sánh kết quả Swin-UMamba và các model khác.

## 7 Conclusion

Như vậy, trong bài viết này ta đã tìm hiểu về kiến trúc Mamba được áp dụng cho data dạng ảnh thông qua nhiều bài toán khác nhau. Ta thấy rằng, các model Mamba-based này hoàn toàn vượt trội so với Transformer về cả 3 tiêu chí: tốc độ, tài nguyên tiết kiệm, và độ chính xác cao. Các tiêu chí này đã đúng đắn với data dạng text và dạng ảnh, vốn là 2 loại data phổ biến nhất. Từ đó, ta có thể mường tượng được rằng Mamba cũng sẽ tốt ở hầu hết các loại data còn lại.

## References

- [1] Gu, Albert, and Tri Dao. “Mamba: Linear-time sequence modeling with selective state spaces.” arXiv preprint arXiv:2312.00752 (2023)
- [2] Zhu, Lianghui, et al. “Vision mamba: Efficient visual representation learning with bidirectional state space model.” arXiv preprint arXiv:2401.09417 (2024).
- [3] Liu, Yue, et al. “Vmamba: Visual state space model.” arXiv preprint arXiv:2401.10166 (2024).
- [4] Ma, Jun, Feifei Li, and Bo Wang. “U-mamba: Enhancing long-range dependency for biomedical image segmentation.” arXiv preprint arXiv:2401.04722 (2024).
- [5] Liu, Jiarun, et al. “Swin-UMamba: Mamba-based UNet with ImageNet-based pretraining.” arXiv preprint arXiv:2402.03302 (2024).