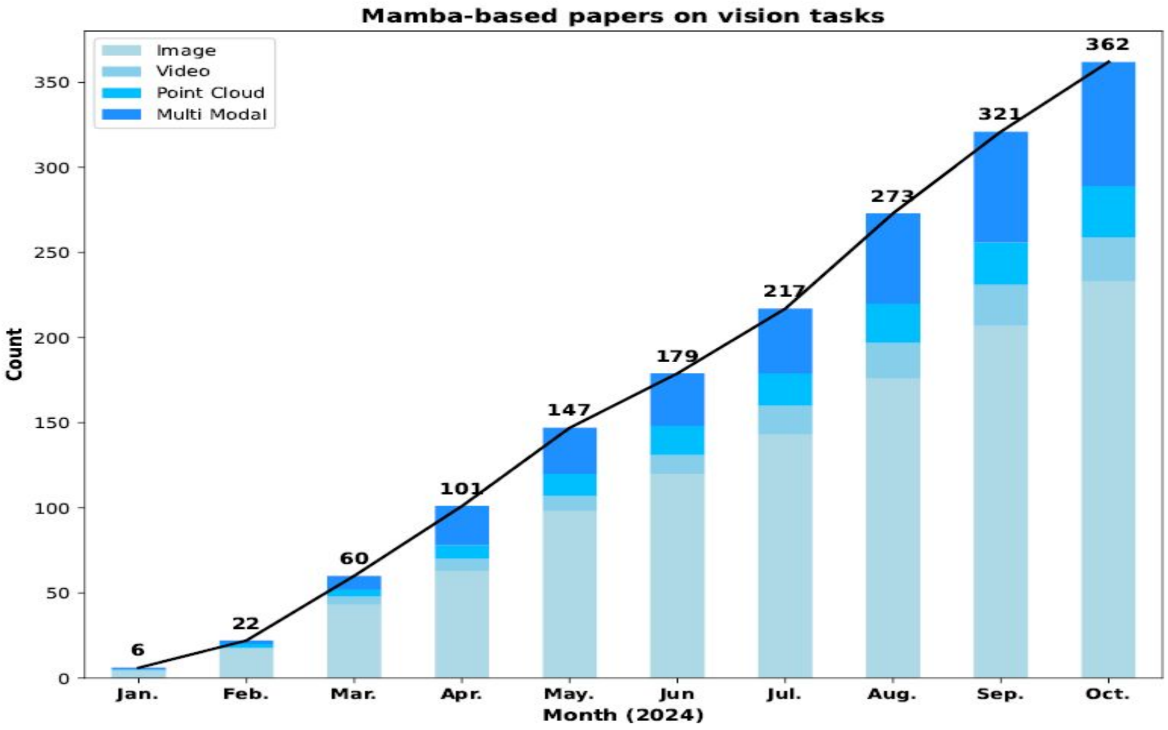
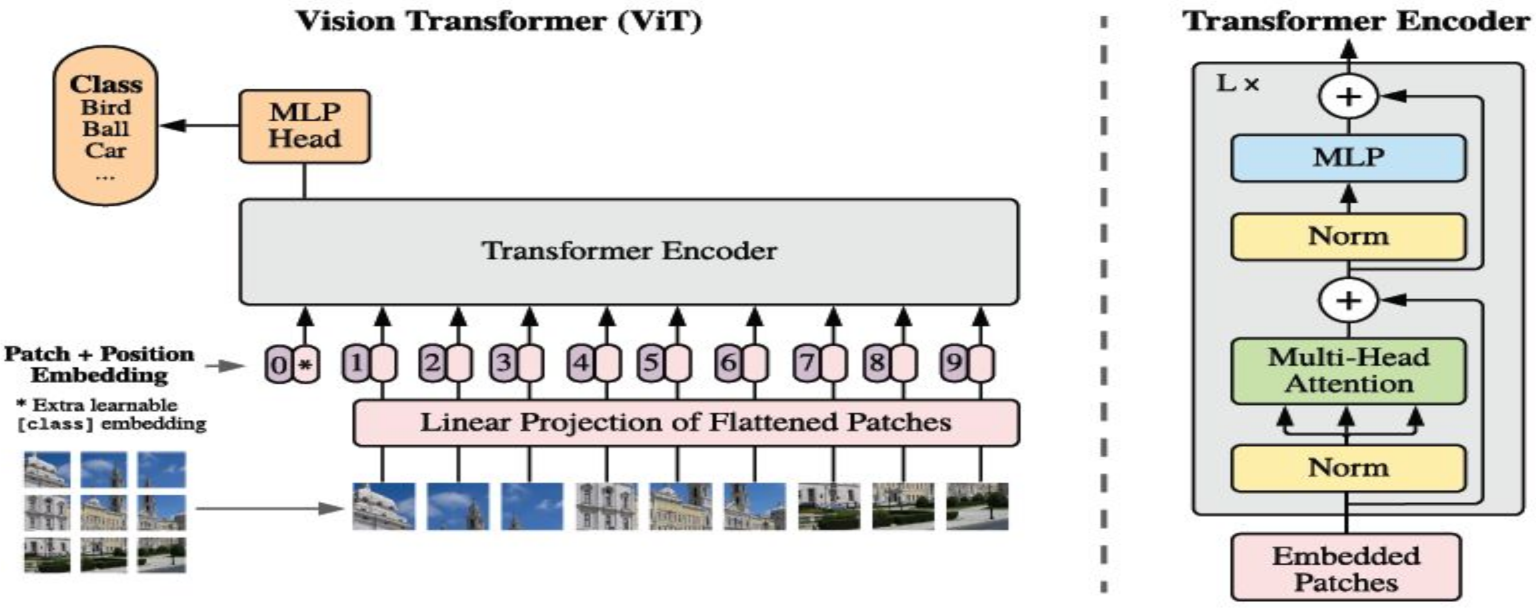
Rising Popularity of Mamba: A Snapshot of Vision Task Publications Across Modalities

The figure is sourced from the provided [HTTPS]() link.

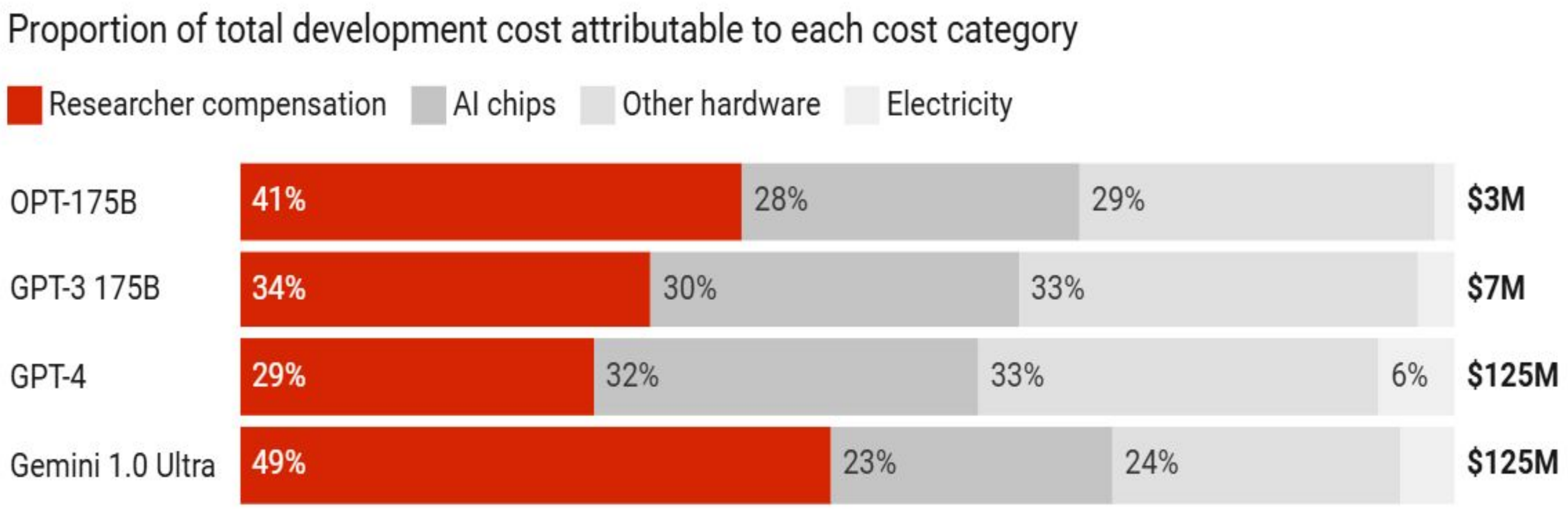


- **Transformers:** Transformers use attention mechanisms to find important features and improve data representation. They are widely used in image analysis because they can understand long-distance relationships in data.

The figure is taken from the famous paper "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE."



- **Why are transformers not suitable for our task?:** Their self-attention mechanism requires a lot of computing power, especially with high-resolution images. This makes it hard to use them on devices with limited resources and to maintain fast performance in real-time systems.
- The estimated cost for training GPT-3 was over $4.6 million using Tesla V100 cloud instances. !!!! The figure is taken from [Time] and was prepared by Will Henshall for TIME.



State Space Models (SSMs)

This section is a summary from the source.

New models using sparse attention mechanisms and different neural network approaches aim to reduce computational costs while still capturing long-range dependencies and maintaining high performance. **State Space Models (SSMs)** have become a key focus in these developments.

- Recently, the state space model (SSM) was brought into deep learning to model sequences, and its parameters are learned using a method called gradient descent (2021).
- SSMs were not used much before because they needed too much computing power and memory. This changed with the introduction of structured SSMs (S4), which solved these problems by improving how state matrices are handled (2022).
- **SSMs' fixed way of handling sequences limits their ability to understand context, which is important for models like Transformers to work well.!!!**

And Mamba enters the scene!

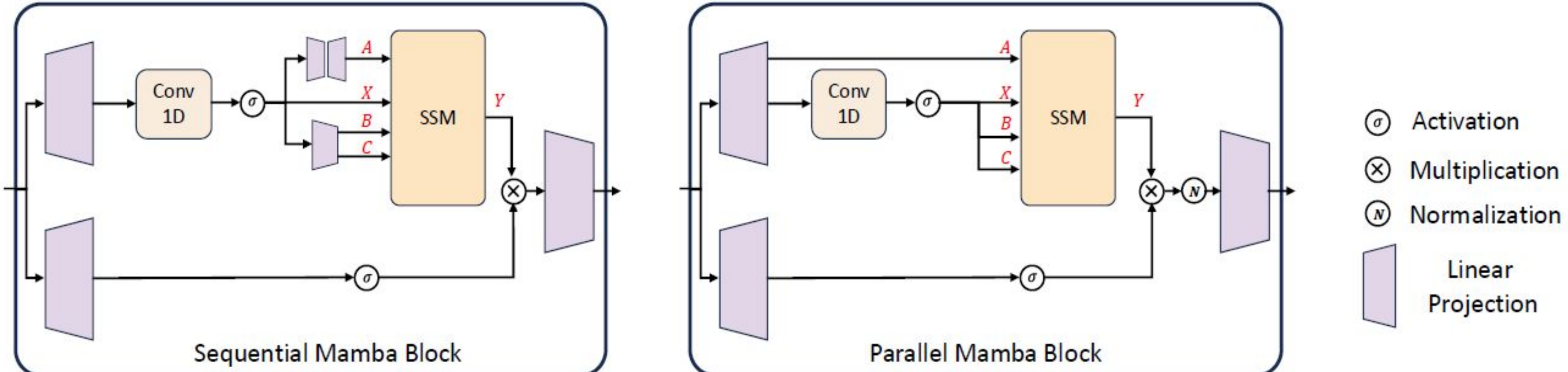
Mamba-1 (2021):

- Added a selection mechanism to SSMs.
- Allows SSMs to choose which information to keep or forget in a sequence based on the current token.
- Introduced a hardware-aware algorithm to make computations more efficient.

Mamba-2 (2022):

- Linked SSMs with attention-based methods.
- Improved the selective SSM by creating a more efficient algorithm called state space duality (SSD).

The figure is taken from the referenced source.



Mamba for Computer Vision

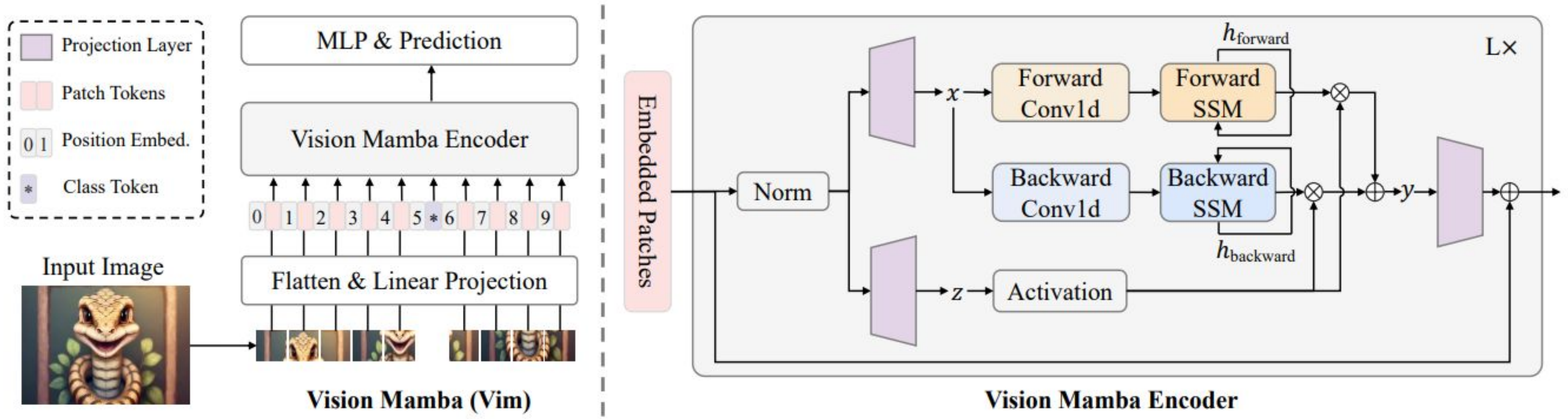
VMamba:(2024)

- Processes image patches by turning them into sequences.
- Scans these sequences in both horizontal and vertical directions.

Vision Mamba (Vim) :(2024)

- Uses position embeddings to add spatial information to the model.
- Inspired by the Vision Transformer (ViT).

The figure is sourced from the Vision Mamba paper.



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	80.3
Vim-S†	224^2	26M	81.4 +1.1
Vim-B	224^2	98M	81.9
Vim-B†	224^2	98M	83.2 +1.3

The table is sourced from the [Vim paper.]

Comparison with Self-Attention-Based ViT:

- Vim outperforms ViT significantly in both parameter efficiency and classification accuracy.

Comparison with Optimized DeiT Variants:

- Vim-Tiny scores 3.9% higher than DeiT-Tiny.
- Vim-Small scores 0.5% higher than DeiT-Small.
- Vim-Base scores 0.1% higher than DeiT-Base.

Comparison with SSM-Based S4ND-ViT-B:

- Vim achieves similar accuracy but uses three times fewer parameters.

Performance After Long Sequence Fine-Tuning:

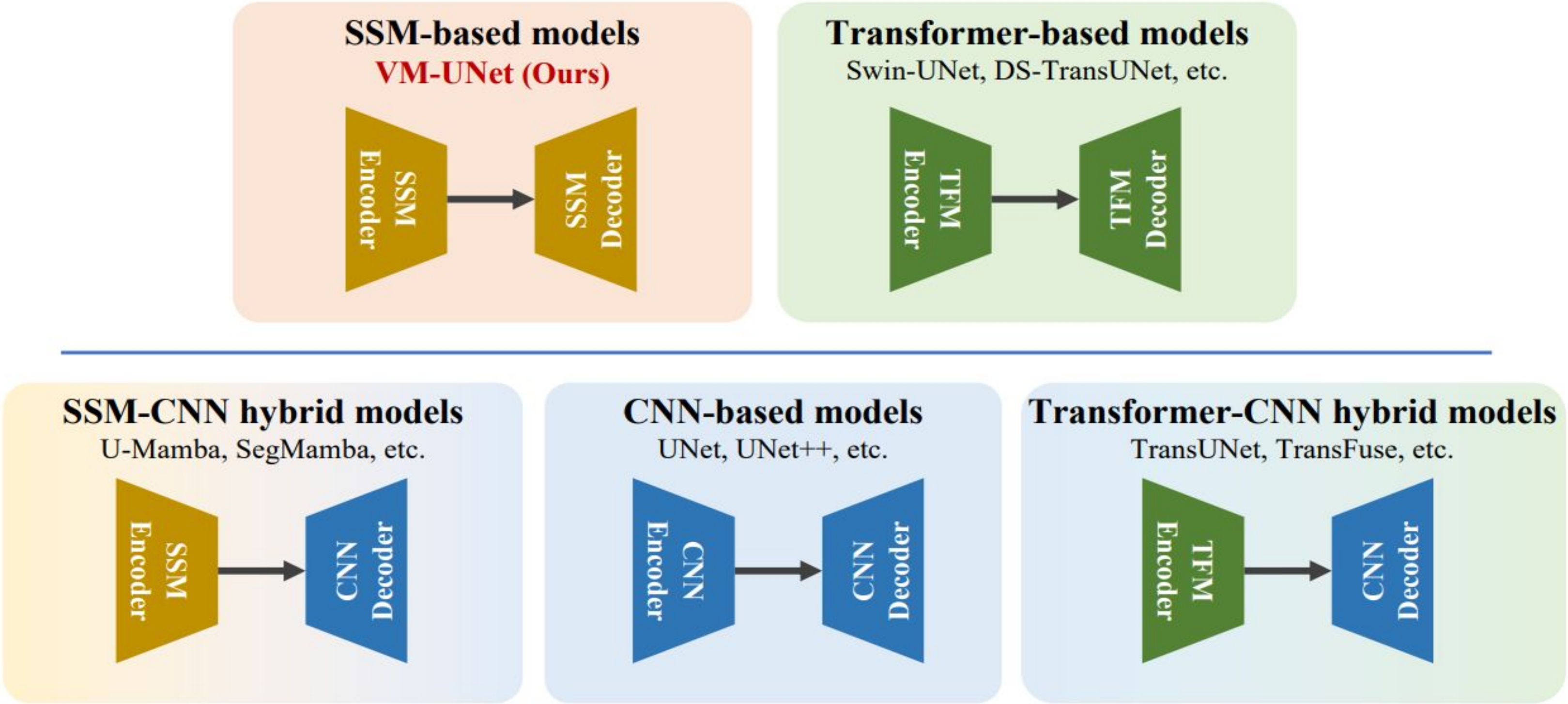
- Vim-Tiny, Vim-Small, and Vim-Base improve further after fine-tuning.
- Vim-Small achieves results comparable to DeiT-Base.

Conclusion:

- Vim adapts well to long-sequence modeling and provides stronger visual representations.

VM-UNet: Vision Mamba UNet for Medical Image Segmentation

The figure is taken from [VM-UNET paper.]



Preliminaries

Structured State Space Models (SSM): S4 and Mamba

- **Concept:**
  These models use classical continuous systems to process a 1D input $x(t)$ into an output $y(t)$ through intermediate states $h(t)$.
- **Mathematical Framework:**
  - Represented by a linear Ordinary Differential Equation (ODE):
  - $h'(t) = A\,h(t) + B\,x(t)$
  - $y(t) = Ch(t)$
  - Parameters:
    - A; State matrix ($N \times N$)
    - B: Input projection ($N \times 1$)
    - C Output projection ($1 \times N$)
- **Adaptation for Deep Learning:**
  - **Discretization:**
    - Introduces a timescale parameter Δ.
    - Converts A and B using **Zero-Order Hold (ZOH): $A = \exp(\Delta A)$)** and **$B = (\Delta A)-1(\exp(\Delta A)-I) \cdot \Delta B$**

Purpose: Discretization enables S4 and Mamba to efficiently handle sequence modeling in deep learning.

Computation Methods in SSM-Based Models

- **Two Approaches:**
  - **Linear Recurrence:**
    $h'(t) = A\,h(t) + B\,x(t)$ and $y(t)=Ch(t)$
  - **Global Convolution:**
    $K=(CB,CAB,\ldots,CAL-1B)$ and $y=x*K$

    K: Structured convolutional kernel ($\in R^L$)

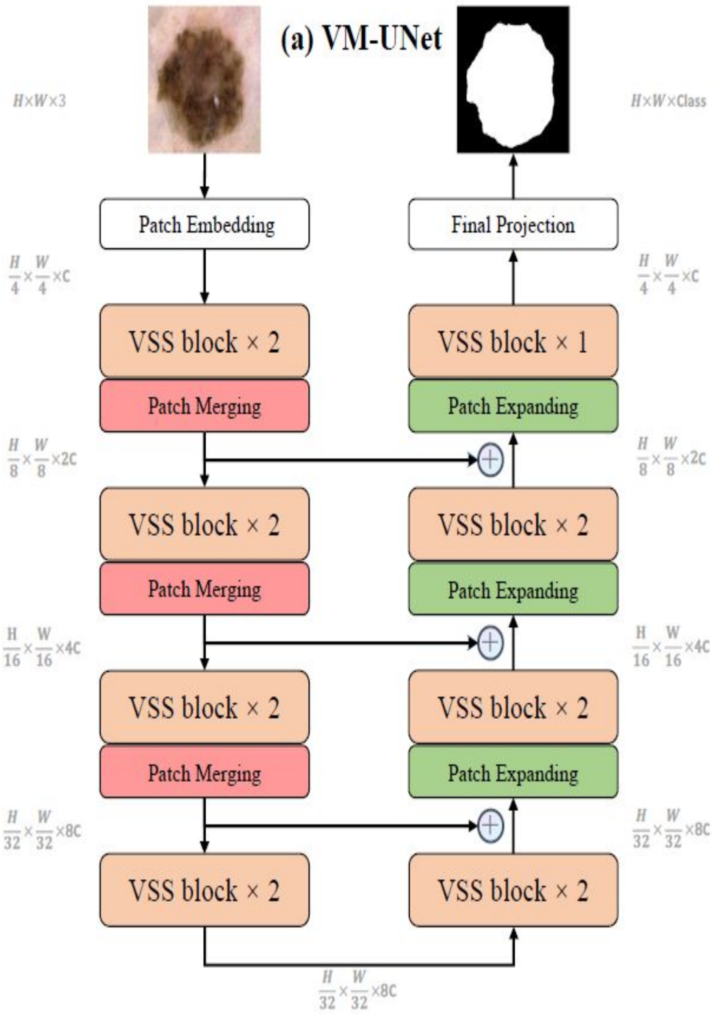
    L: Length of the input sequence x

- **Significance:**
  These methods enable efficient computation of sequence outputs, optimizing SSM models for deep learning tasks.

M-UNext: Architecture Overview

The figure is taken from the source.

- **Patch Embedding Layer:**
  - Splits the input image x ∈ R^{H×W×3) into 4×4 patches.
  - Maps to C (default C=96).
  - Outputs normalized embedded image
- **Encoder:**
  - 4 stages with patch merging at the first 3 stages.
  - Reduces spatial dimensions, increases channels.
  - Channel counts: [C, 2C, 4C, 8C].
  - Uses [2, 2, 2, 2] VSS blocks.
- **Decoder:**
  - 4 stages with patch expanding at the last 3 stages.
  - Increases spatial dimensions, reduces channels.
  - Channel counts: [8C, 4C, 2C, C].
  - Uses [2, 2, 2, 1] VSS blocks.
- **Final Projection Layer:**
  - 4× upsampling to match segmentation target size.
  - Restores feature dimensions.
- **Design Highlights:**
  - **Asymmetric Structure:** Optimized for performance.
  - **Skip Connections:** Simple addition with no extra parameters.



Core module of VM-UNet

The figure above is referenced from source.

Input Processing:

- After Layer Normalization, the input splits into two branches.

First Branch:

- Passes through a linear layer followed by an activation function.

Second Branch:

- Input undergoes a linear layer, depthwise separable convolution, and an activation function.
- Features go through the 2D-Selective-Scan (SS2D) module for extraction.

Feature Merging:

- Normalized features are combined using element wise multiplication with the first branch output
- Linear layer mixes the features and adds a residual connection to form the final output.

Activation Function:

- SiLU used as the default activation function.