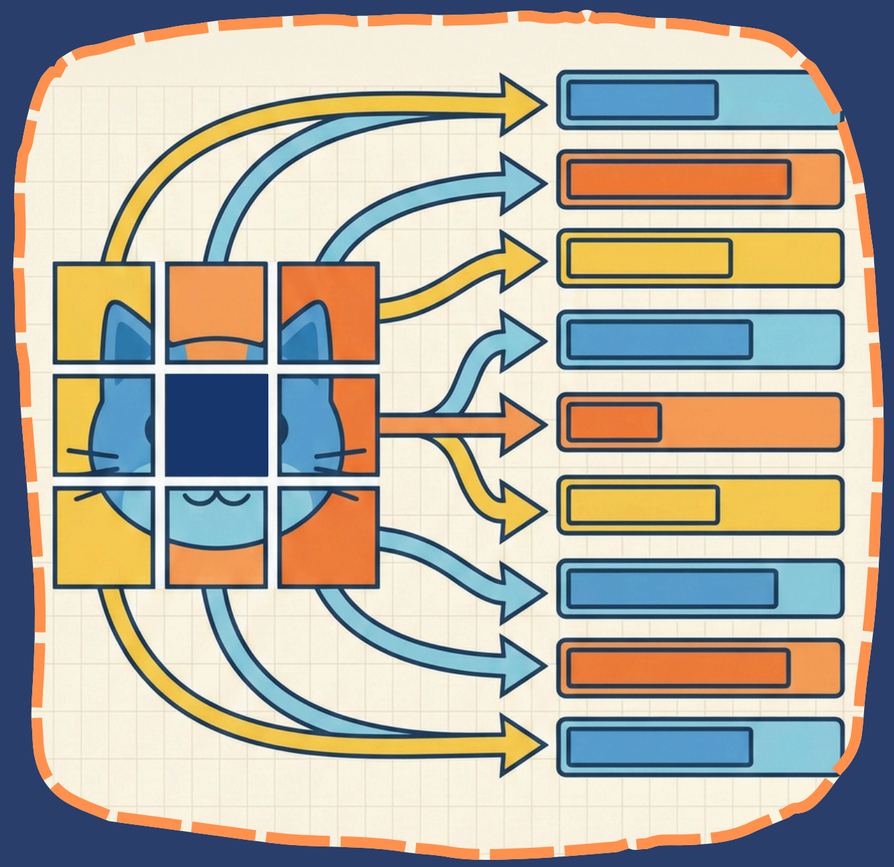
Rumah Coding | Technical Webinar Series

ViT vs CNN: The Clash of Architectures

Is Convolution Dead? The Rise of Attention in Computer Vision.



Lhuqita Fazry

Founder Rumah Coding

Lhuqita Fazry

Rumah Coding

Lhuqita Fazry, S.Si., M.Kom.

(Founder Rumah Coding)

Pendidikan:

- S1: Math, Universitas Indonesia
- S2: Ilmu Komputer, Universitas Indonesia

Pengalaman Riset:

- Intelligent Robots and Systems (IRoS) Lab, Faculty of Computer Science, Universitas Indonesia
- Human-AI Interaction Laboratory, Japan Advance Institute of Technology (JAIST)

Lhuqita Fazry

Rumah Coding

HifiDiff: Two Stream Diffusion Models for High Fidelity Speech Generation of Unseen Languages 28th International Conference of Oriental COCOSDA

Lhuqita Fazry, Kurniawati Azizah, Dipta Tanaya, Ayu Purwarianti, Dessi Lestari and Sakriani Sakti

Hybrid Wavelet-Attention Model for Detecting Changes in High-Resolution Remote Sensing Images IEEE Access

Lhuqita Fazry, Mgs M Luthfi Ramadhan, Alif Wicaksana Ramadhan, Muhammad Febrian Rachmadi, Aprinaldi Jasa Mantau, Lukito Edi Nugroho, Chi-Hung Chi, Wisnu Jatmiko

Change Detection of High-resolution Remote Sensing Images Through Adaptive Focal Modulation on Hierarchical Feature Maps IEEE Access

Lhuqita Fazry, Mgs M Luthfi Ramadhan, Wisnu Jatmiko

Hierarchical Vision Transformers for Cardiac Ejection Fraction Estimation

7th International Workshop on Big Data and Information Security (IWBIS)

Lhuqita Fazry, Asep Haryono, Nuzulul Khairu Nissa, Sunarno, Naufal Muhammad Hirzi, Muhammad Febrian Rachmadi, Wisnu Jatmiko

A Split-then-Join Approach to Abstractive Summarization for Very Long Documents in a Low Resource Setting

9th International Conference of Computer and Informatics Engineering

Lhuqita Fazry, Evi Yulianti

Improving Remote Sensing Change Detection via Locality Induction on Feed-forward Vision Transformer Journal of Computer Science and Information

Lhuqita Fazry, Mgs M Luthfi Ramadhan, Wisnu Jatmiko

Unsupervised Raindrop Removal from a Single Image using Conditional Diffusion Models arXiv preprint arXiv:2505.08190

Lhuqita Fazry, Valentino Vito, Laksmita Rahadianti, Aniati Murni Arymurthy

Contents

Rumah Coding

Before we debate CNN vs. ViT, let's zoom out.
What is the actual job of these architectures?

The Anatomy of Vision

CNN and ViT are here

A simple layer that draws a line between those numbers.

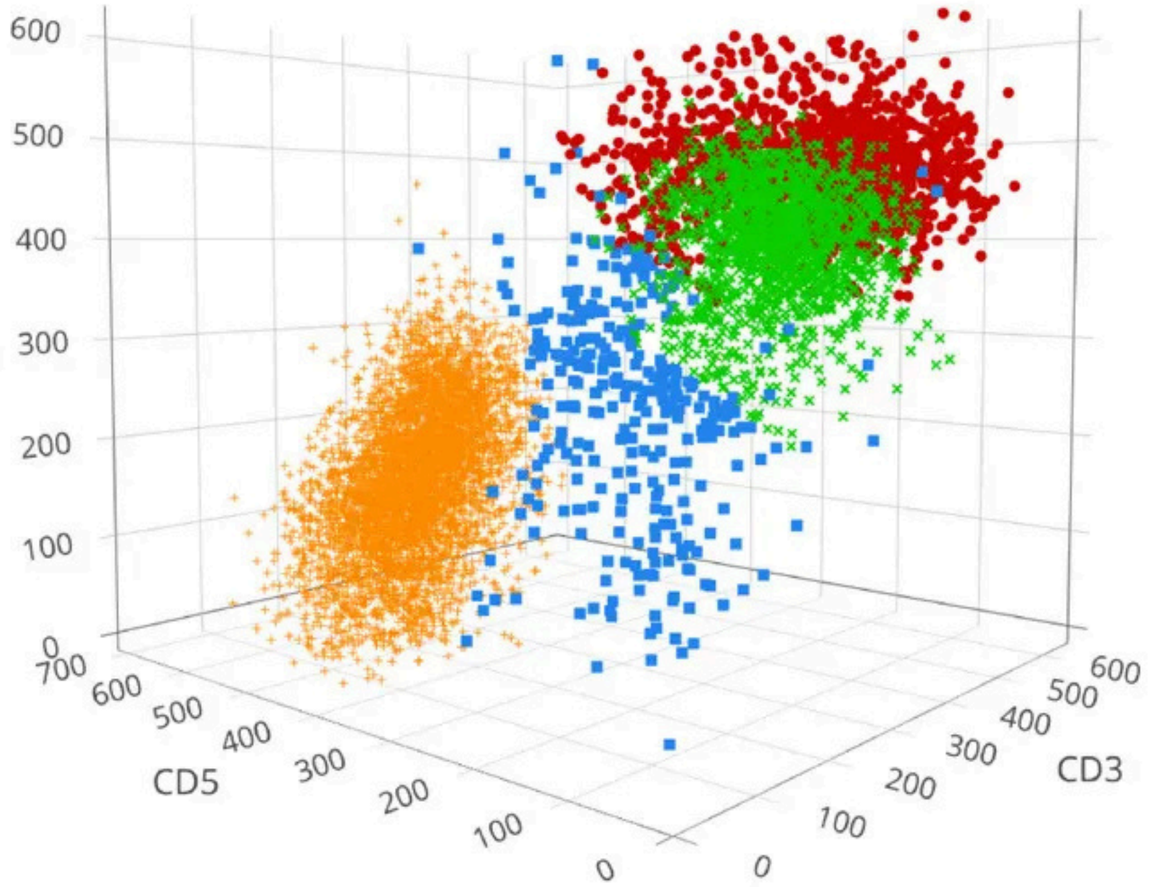
Feature Extractor

Classifier

Cat

Translates raw pixels into meaningful numbers (Features).

The quality of our model depends almost entirely on how good the 'Feature Extractor' is at summarizing the image."



https://pub.towardsai.net/what-are-dimensions-in-machine-learning-0c42051d6693

What makes a "Good" Feature?

- Good features make different classes (e.g., Cat vs. Dog) mathematically distinct (separability).
- The features shouldn't change just because the object moved or the lighting changed (robust).
- If the Extractor does a good job, the Classifier's job is trivial.

How do we generate features that capture the essence of the object, not just its texture?

Rumah Coding

Which architecture creates richer, more unique feature representations?

CNN (Local Context)

- Focuses on neighboring pixels (Edges, Textures).
- Struggles to connect distant parts (e.g., Head <-> Tail) without many layers.

ViT (Global Context)

- Every patch talks to every other patch instantly.
- It understands the relationship between the top-left corner and bottom-right corner immediately.

ViT generates features based on the Global Structure, making them more unique and robust than features based solely on local texture.

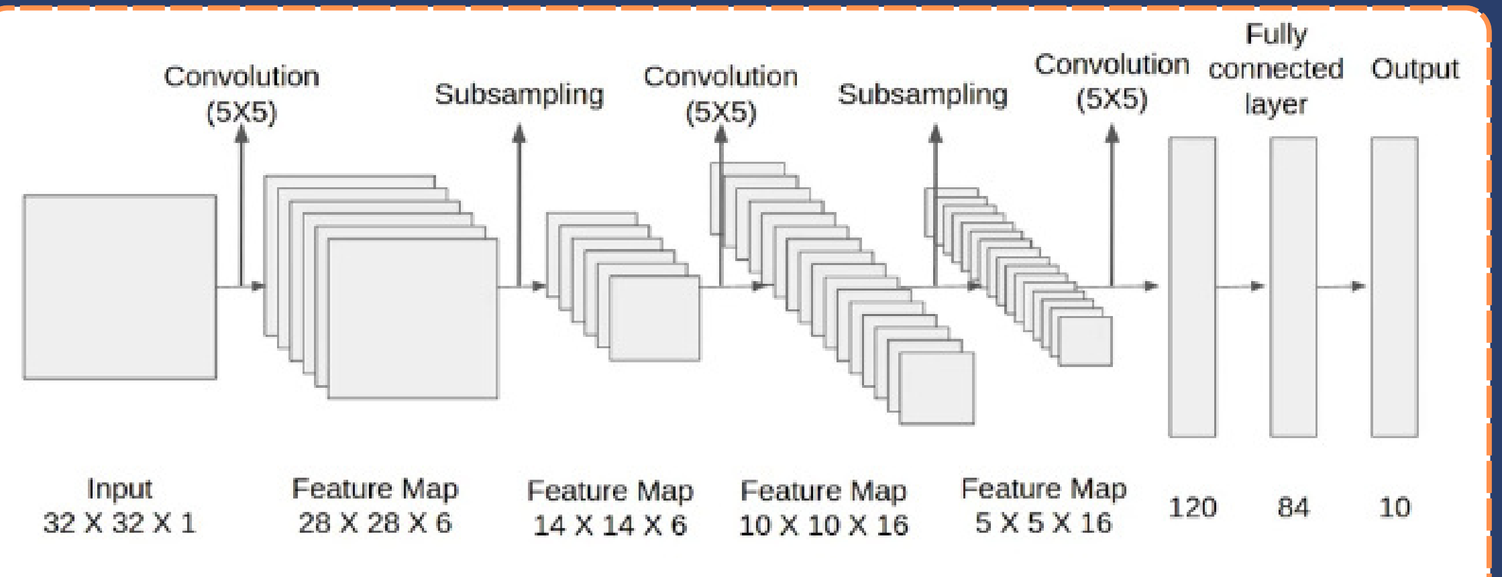
The Decade of Convolutions

How CNNs Conquered Computer Vision

Rumoh Coding

The Genesis: LeNet-5 (1998)

- LeNet is the grandfather of Convolutional Neural Networks.
- Created by Yann LeCun (now Chief AI Scientist at Meta) to recognize handwritten digits (MNIST) on bank checks.
- LeNet architecture is a simple 5-layer network.
- While it solved the digit recognition problem perfectly, it failed to scale to real-world objects (like cats vs. dogs) due to lack of training data and computing power.



Input (32x32)

- Conv
- Avg Pool
- Conv
- Avg Pool
- Fully Connected.

https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/

Rumoh Coding

The Race for Depth

2012 — **AlexNet** — The Breakthrough

- Before AlexNet, Computer Vision relied on hand-crafted features.
- Deep + GPU + ReLU.

2014 — **VGG**

- Instead of guessing filter sizes, VGG proved that stacking multiple small 3 x 3 filters is better than using one large filter.

2014 — **Inception**

- Focus: **Multi-scale** Processing
- Instead of choosing between 1 x 1, 3 x 3, or 5 x 5 filters, it uses all of them in parallel and concatenates the results.

2015 — **ResNet** — The Depth Revolution

- Focus: **Ultra-Deep** Training.
- Enabling networks to go from ~20 layers to 100+ layers (and eventually 1000+).

Convolutions are the only way to process image data.

Rumoh Coding

Before ResNet

State of the Art Accuracy

After ResNet

Efficiency & Latency

Rumoh Coding

The Race for Efficiency

2016 — SqueezeNet

- AlexNet accuracy with 50x fewer parameters
- It uses a "Squeeze" layer (1x1 filters) to compress data before feeding it into an "Expand" layer (1x1 and 3x3 filters).

2017 — MobileNet

- This is arguably the most important architecture for mobile developers. It introduced a new mathematical operation that changed everything, Depthwise Separable Convolutions.

2017 — ShuffleNet

- ShuffleNet introduced a "Shuffle" operation that mixes channels between groups, restoring information flow without the computational cost of dense convolutions.

2019 — EfficientNet

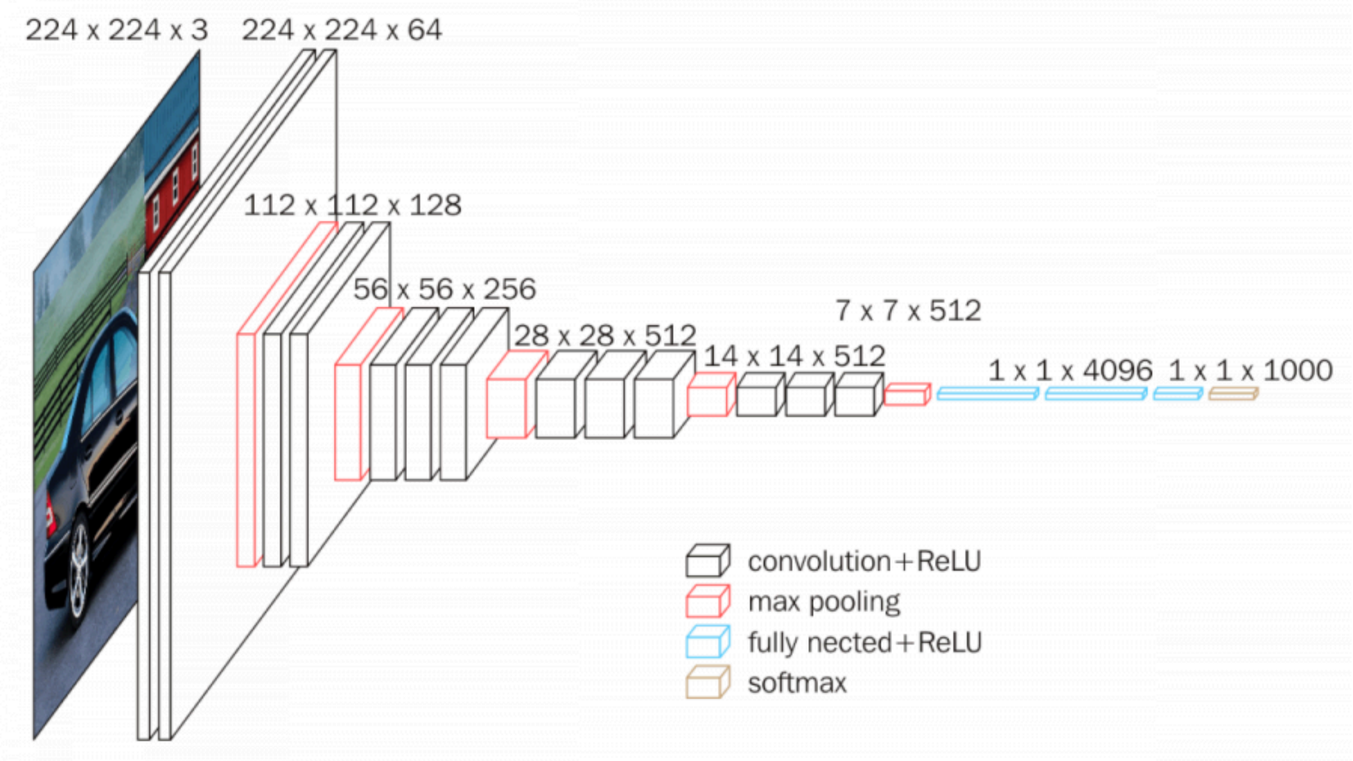
- The researchers discovered a mathematical formula (compound coefficient) that scales all three dimensions (depth, width, resolution) uniformly.

Efficiency & Latency

Rumoh Coding

We've seen the history. Now, let's open up the hood of the three most influential architectures. We need to understand how they work to understand why Vision Transformer was such a shock to the system.

- VGG
- ResNet
- MobileNet



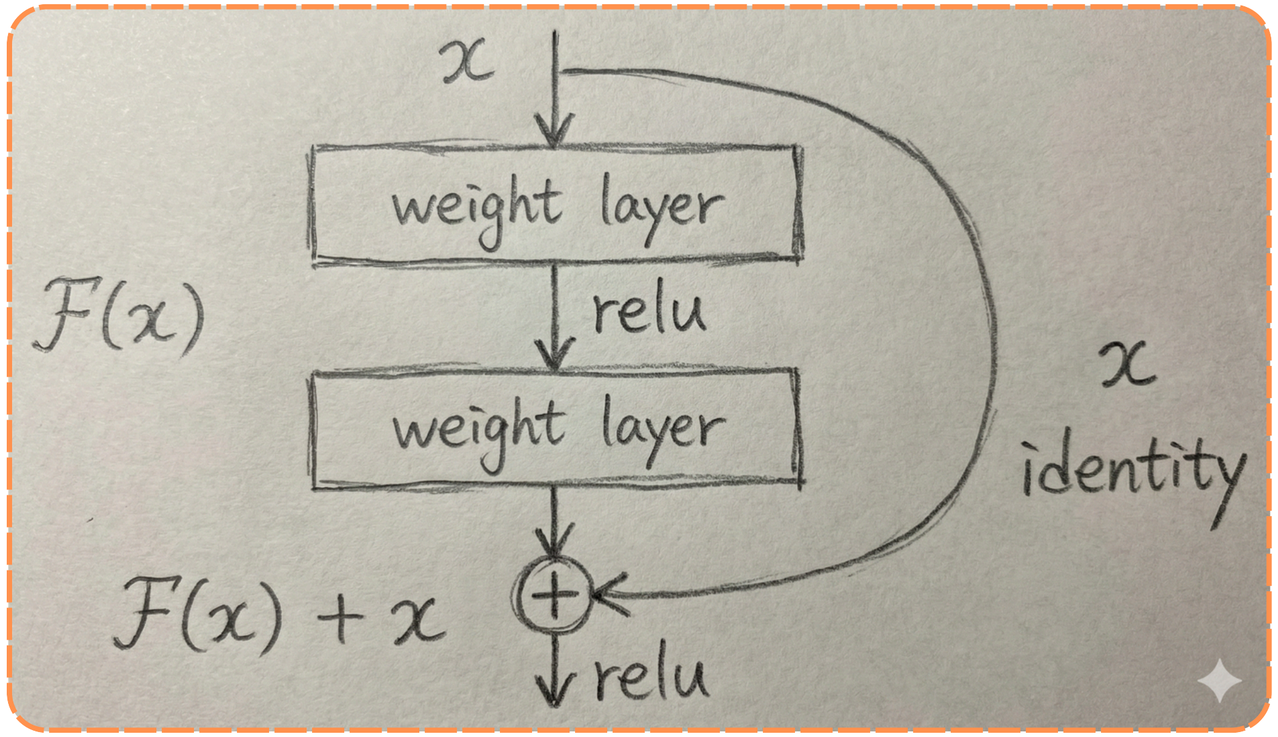
https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide

VGGNet – The Standard of Simplicity

- Uniformity. No more random hyperparameter guessing.
- Replacing large filters (11x11, 7x7) with stacks of small 3x3 filters.
  - Two 3x3 layers have the same "Receptive Field" as one 5x5 layer.
  - Fewer parameters and more non-linearity (more ReLU activations).
- VGG defined the "Standard CNN" structure:

Block of Conv => ReLU => Pooling

Rumoh Coding



ResNet – Solving the Depth Limit

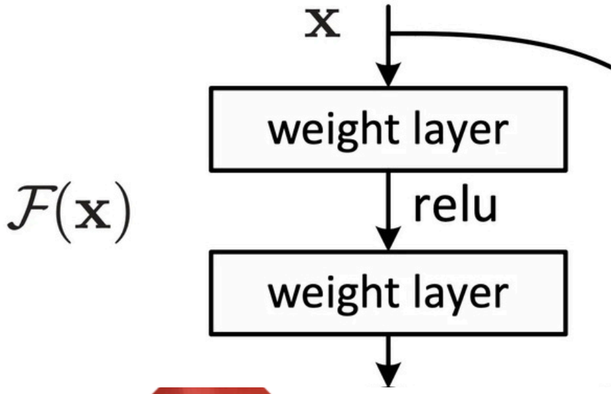
- ResNet solved the vanishing gradient problem
- Vanishing gradient occurs as networks got deeper (20+ layers), accuracy actually dropped.
- ResNet added skip connections (residual blocks). It allows the gradient to flow through the network via an "Identity Shortcut".

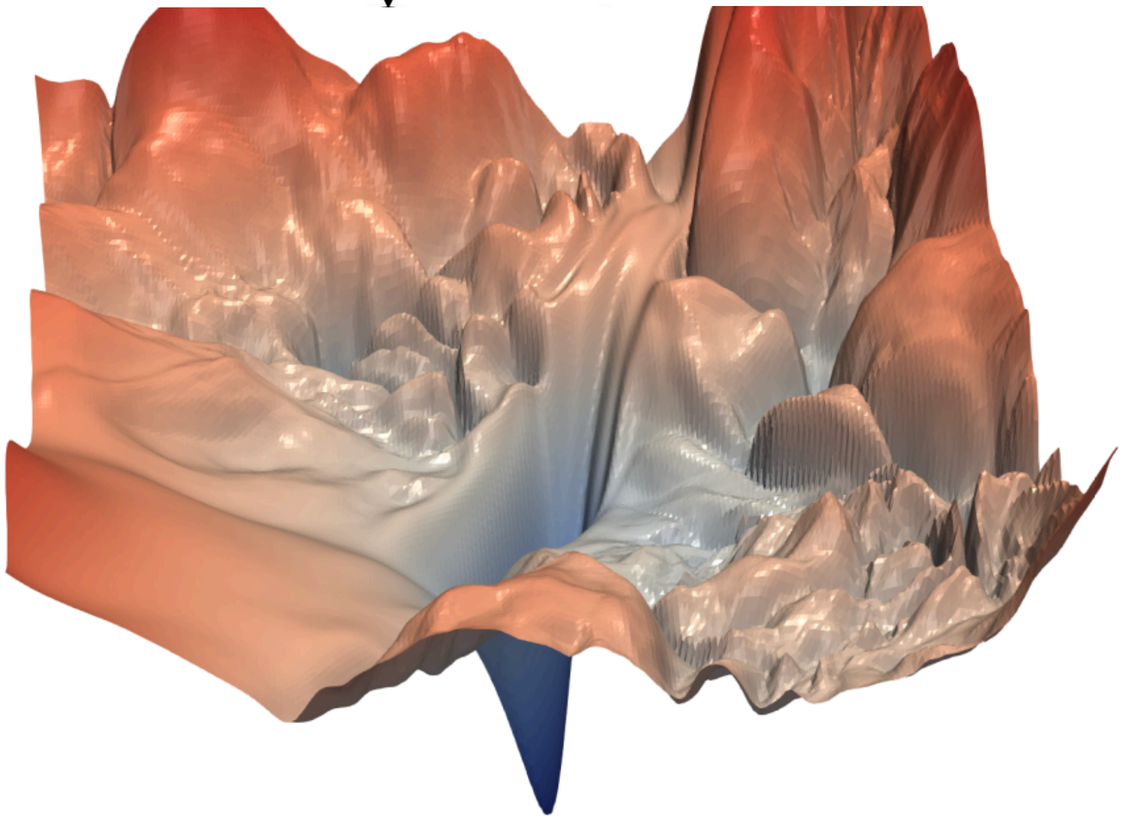
$$y = F(x) + x$$

- Skip connection enabled training of 152 + layers (vs VGG's 19).

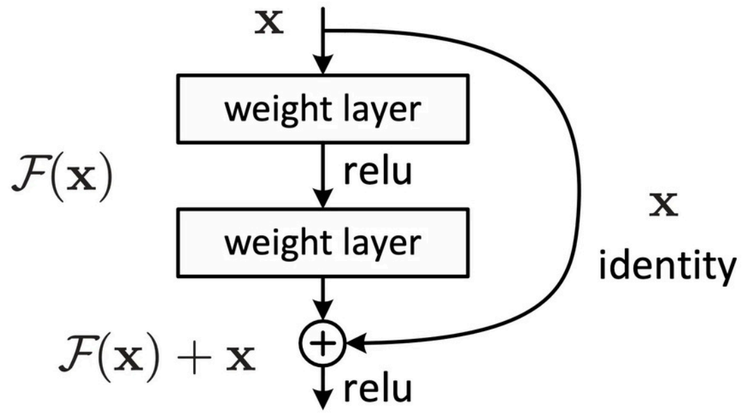
Transformers also use Residual Connections heavily.

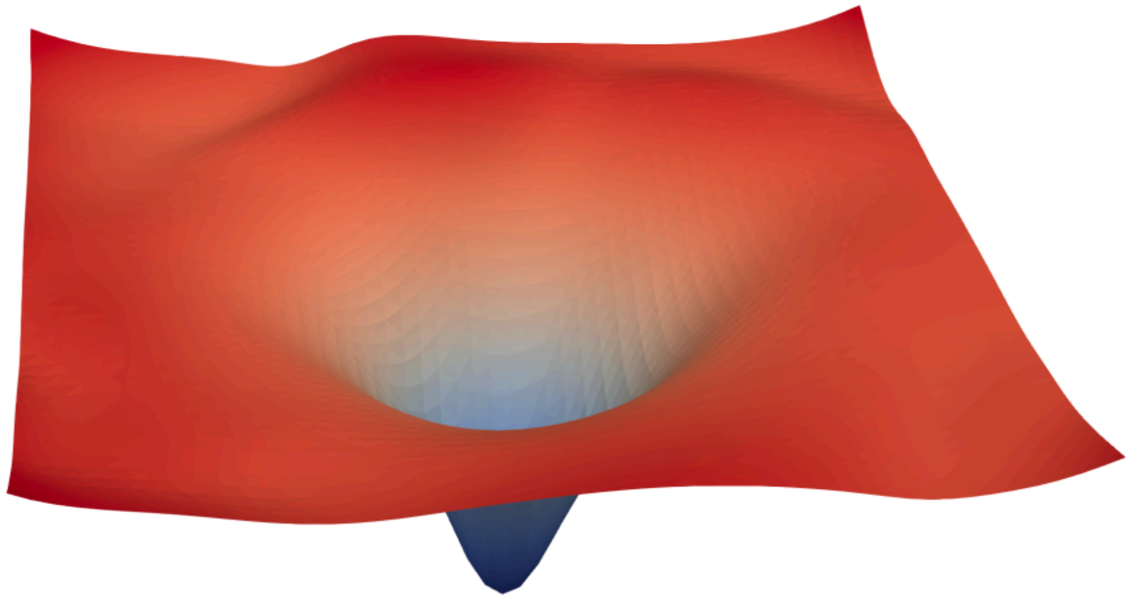
Visualizing the Loss Landscape of Neural Nets





Without Skip Connection

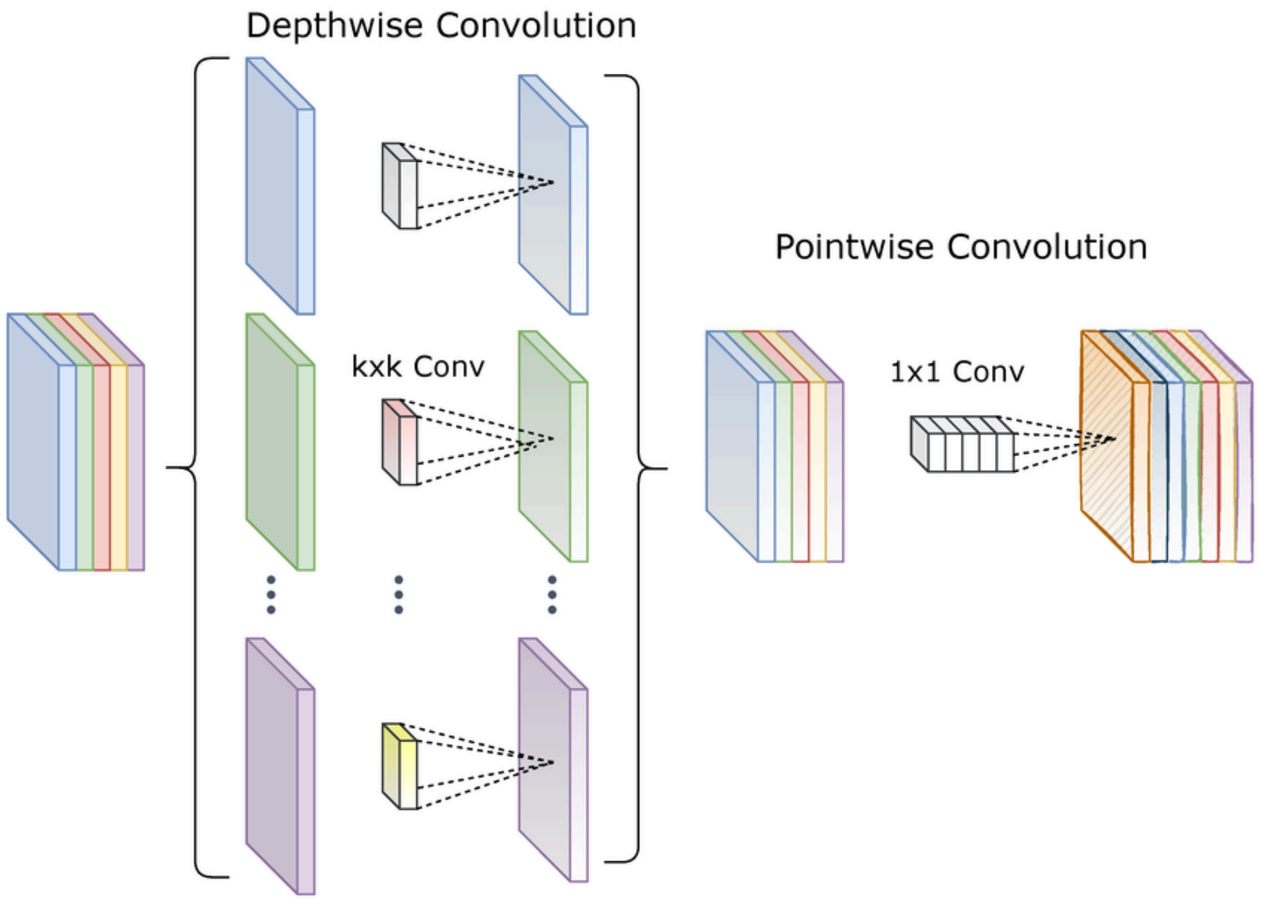




Using Skip Connection

Rumoh Coding

Li, Hao, Zheng Xu, Gavin Taylor and Tom Goldstein. “Visualizing the Loss Landscape of Neural Nets.” *ArXiv* abs/1712.09913 (2017): n. pag.



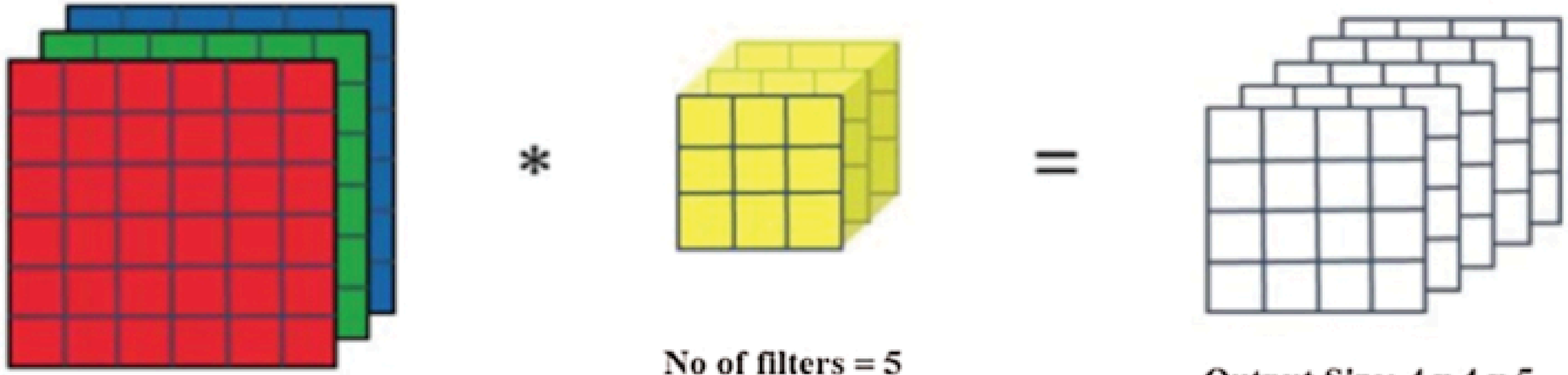
https://www.researchgate.net/figure/Depthwise-separable-convolutions_fig1_358585116

MobileNet – The Efficiency King

- Standard Conv: Filters spatial features and channels simultaneously which is expensive.
- MobileNet use Depthwise Separable Convolutions, consist of:
  - Depthwise Conv: Filters spatial features only.
  - Pointwise Conv (1x1): Mixes channels only.
- Reduces computation cost by ~8-9 times with minimal accuracy loss.

Rumah Coding

Normal Convolution Operation

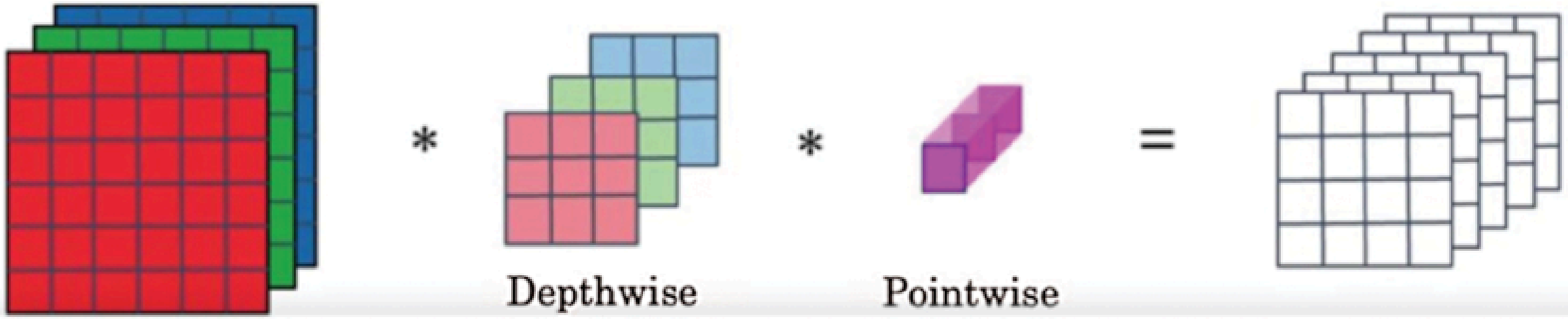


Input Size: 6 x 6 x 3

No of filters = 5
Filter Size = 3 x 3 x3

Output Size: 4 x 4 x 5

Depth Wise Separable Convolution Operation (MobileNet CNN)



Input Size: 6 x 6 x 3

Depthwise

No of filters = 3
Filter Size = 3 x 3

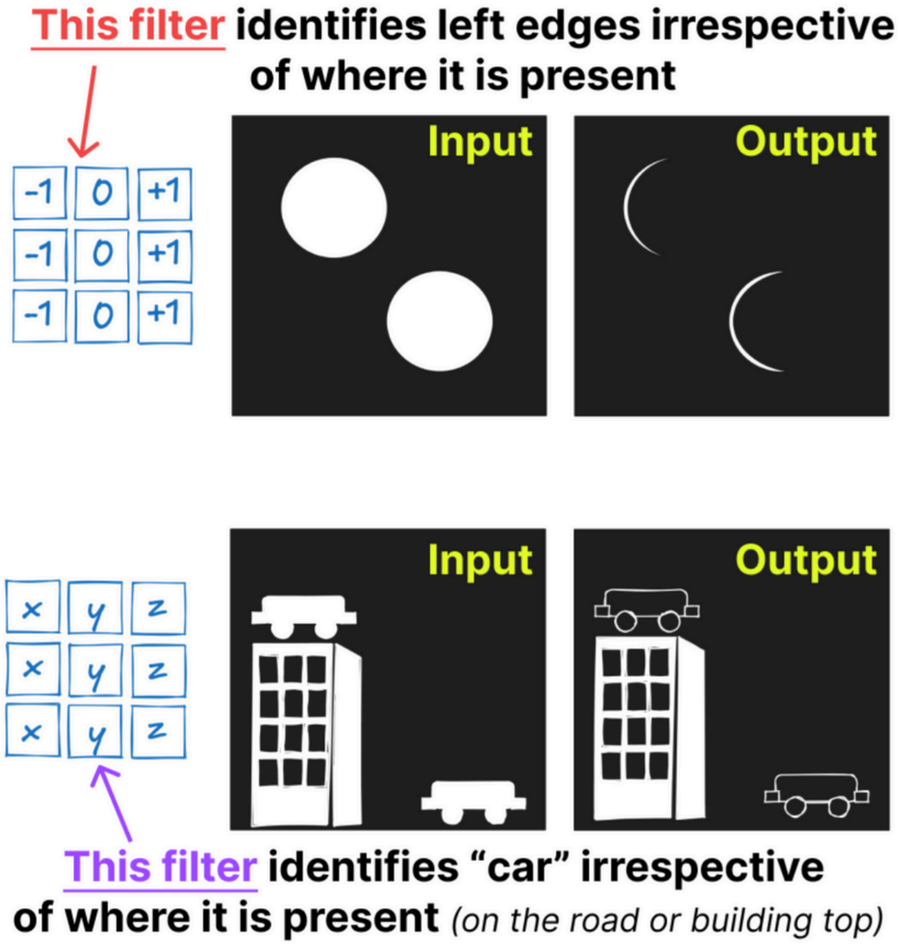
Pointwise

No of filters = 5
Filter Size = 1x 1 x 1

Output Size: 4 x 4 x 5

https://www.researchgate.net/figure/Comparison-between-normal-and-depthwise-separable-convolution-operation_fig4_356837623

Before we deconstruct the Vision Transformer, lets discuss the core concept of CNN

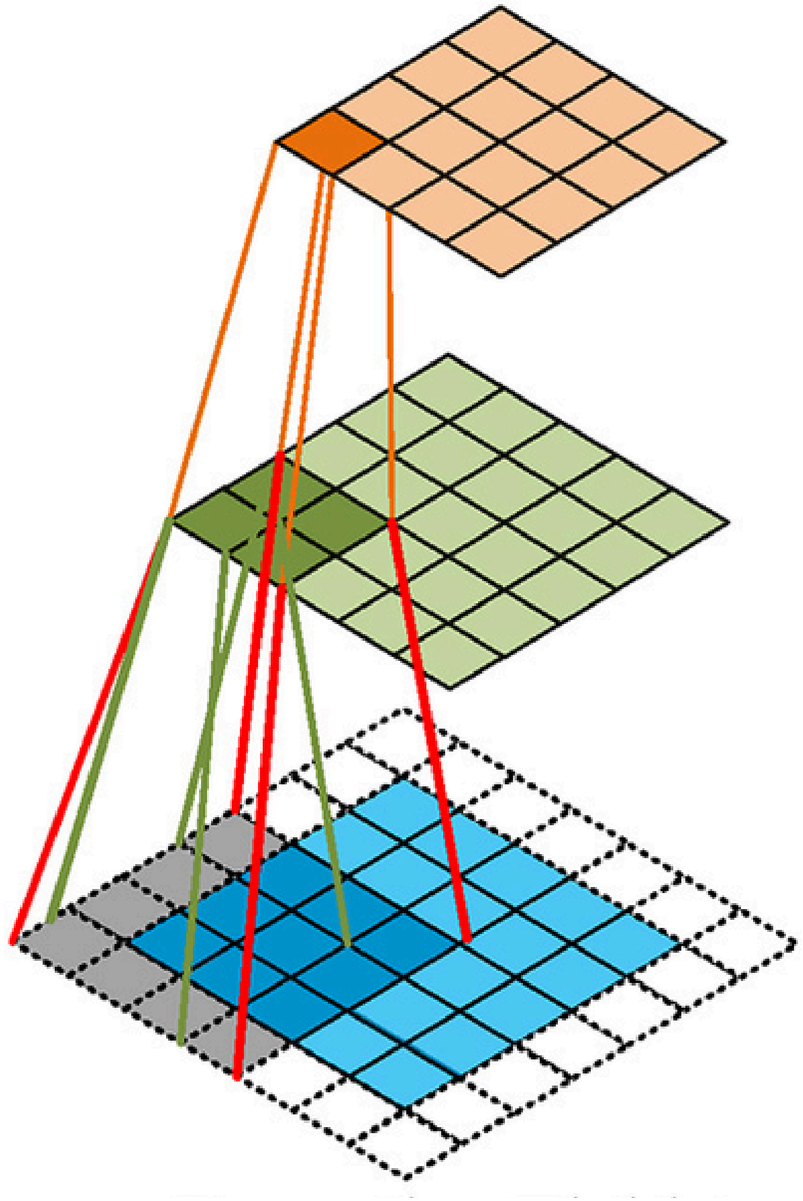


https://www.vizuaranewsletter.com/p/what-exactly-is-inductive-bias-in

The Inductive Bias

- **Inductive Bias** is an assumptions built into the model architecture to help it learn with less data.
  - **Locality**: Pixel correlation is high among neighbors. A pixel's meaning depends on the pixels right next to it.
  - **Translation Invariance**: A cat in the top-left corner is the same as a cat in the bottom-right corner.
- CNN is efficient. It uses weight sharing, the same filter scans the whole image.

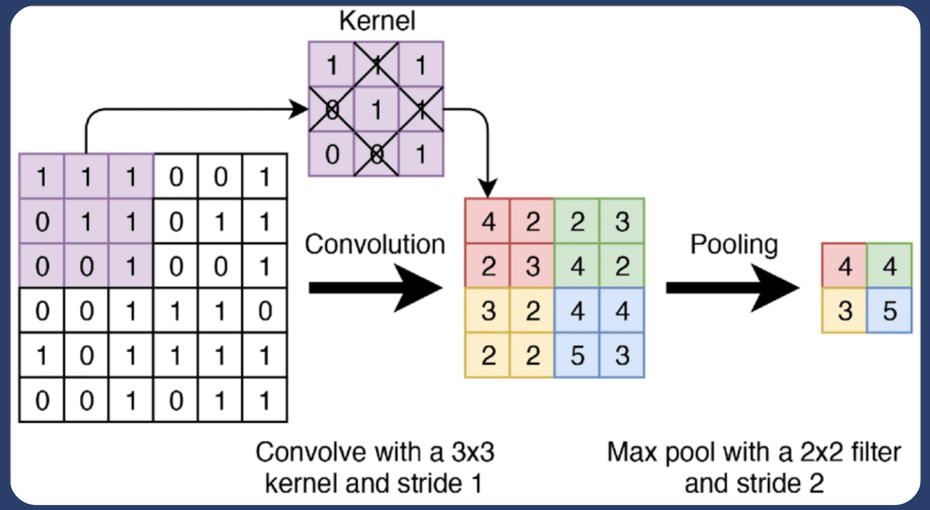
Rumah Coding



https://www.researchgate.net/figure/Receptive-field-a-deep-CNN-A-The-first-convolving-a-3-3-kernel-over-a-5-5-input_fig2_353996855

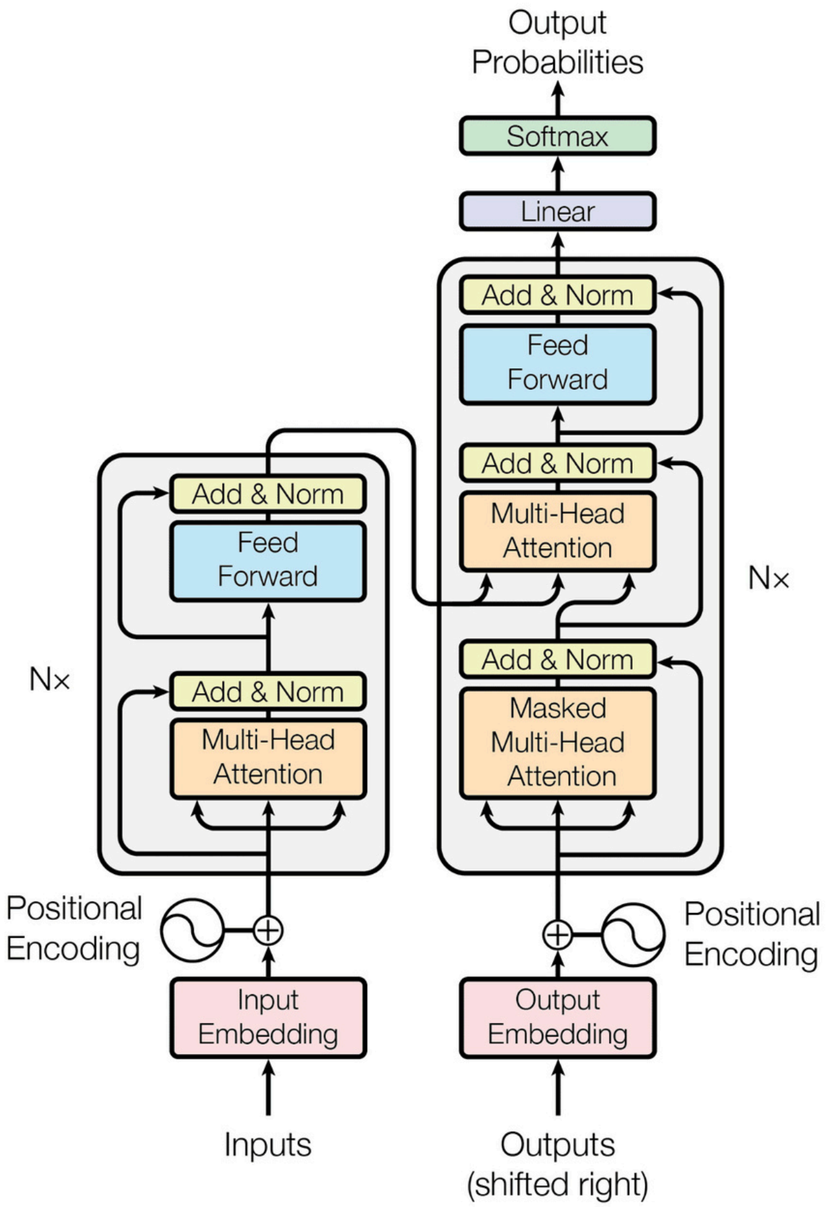
The Global Context Problem

- A single convolution only sees a tiny 3x3 window (Local Receptive Field).
- To understand the relationship between distant pixels, you must stack many layers known as the "Tunnel Vision" Problem.
- Global information is lost or diluted as it passes through down-sampling (Pooling) layers.



https://www.researchgate.net/figure/Schematic-representation-of-a-convolution-and-pooling-layer-in-a-CNN_fig2_331874666

Rumoh Coding



https://machinelearningmastery.com/the-transformer-model/

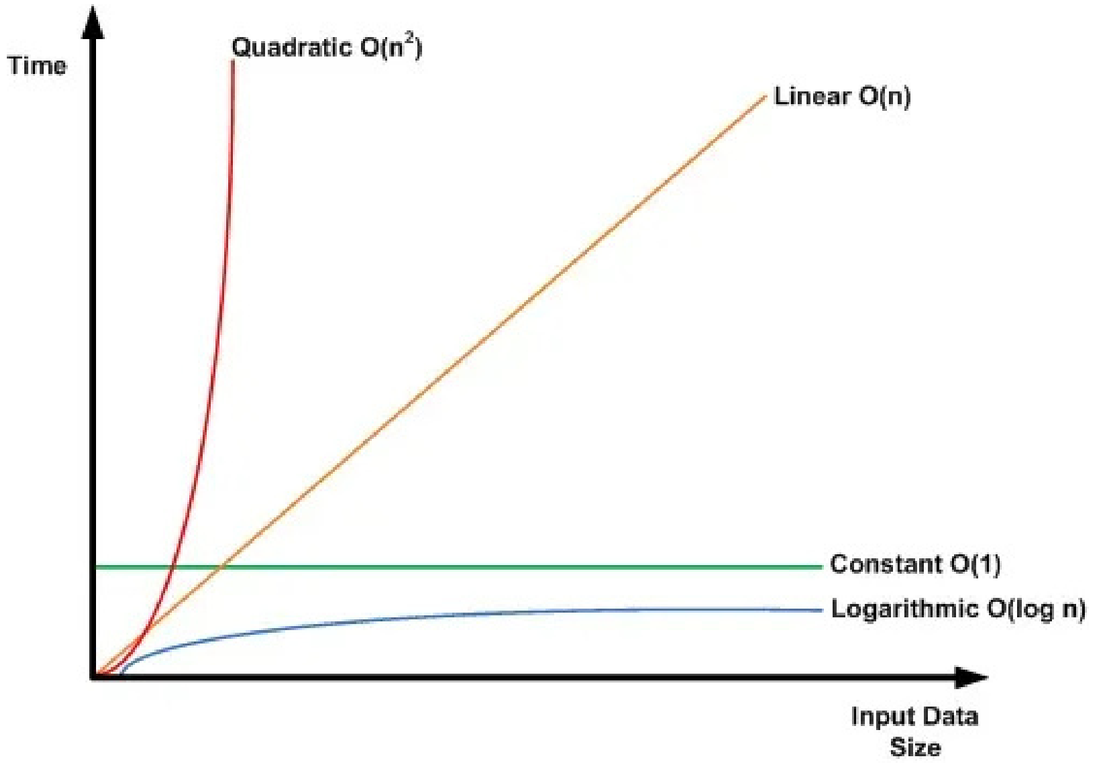
Success in NLP Inspire Computer Vision

- **2017**: "Attention is All You Need" revolutionizes NLP.
- NLP moved from RNNs (Sequential processing) to Transformers (Global parallel processing).
- **Self-Attention**: Every word looks at every other word simultaneously.

If Self-Attention allows words to see the global context immediately... why can't we do the same for pixels?

Rumoh Coding

But, applying self-attention to an image is not straightforward



https://levelup.gitconnected.com/time-complexity-a-guide-to-building-better-software-f8fe633cead

The Computational Hurdle

- Self-Attention has quadratic complexity $O(N^2)$.
- The Math:
  - Text sequence length: ~512 tokens.
  - Image sequence length (pixels): 224 x 224 = 50,176 pixels!
- Calculating attention for individual pixels is computationally impossible for standard hardware.

We need a different approach....

Rumoh Coding

Deconstructing Vision Transformer (ViT)

Rumoh Coding

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

Abstract

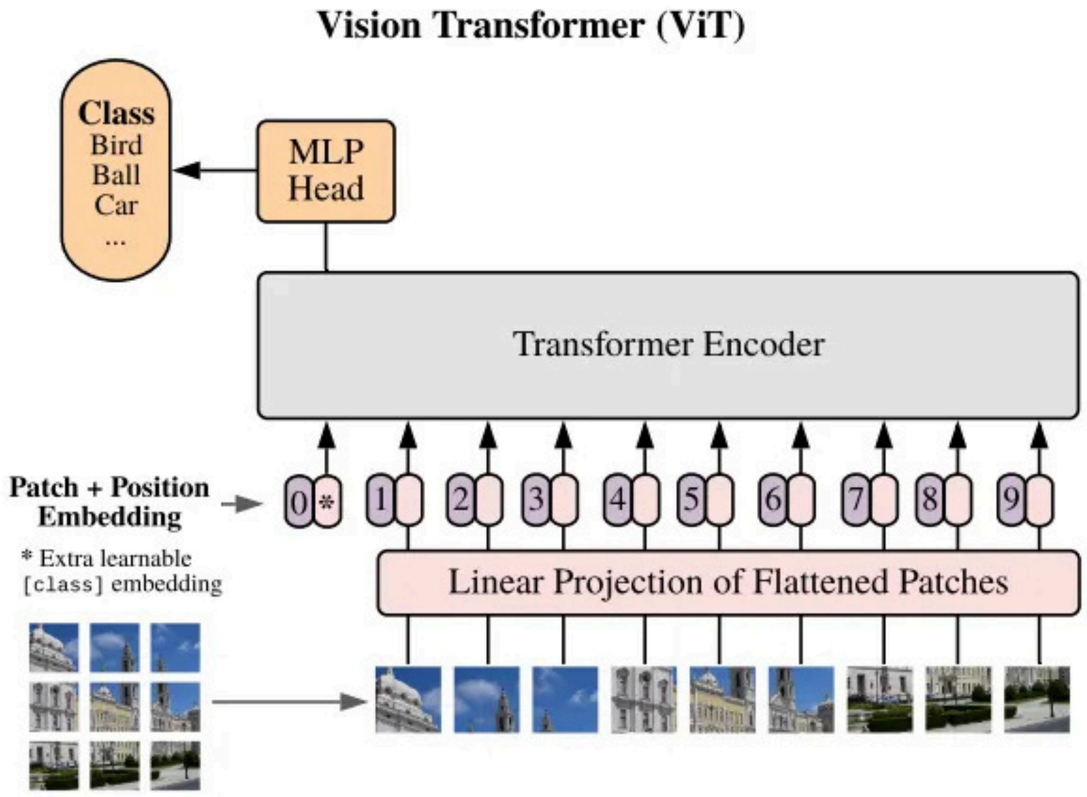
While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

An Image is Worth 16x16 Words

- The paper is published by Google Research (Dosovitskiy et al.) in ICLR 2021.
- It hypothesize: we don't need convolutions. We just need to treat an image as a sequence of sub-images.

NLP: Sentence => Sequence of Words.
ViT: Image => Sequence of Patches.

Rumah Coding



https://medium.com/@fernandopalominocobo/demystifying-visual-transformers-with-pytorch-understanding-patch-embeddings-part-1-3-ba380f2aa37f

Patch Embeddings: The "Tokenization" of Images

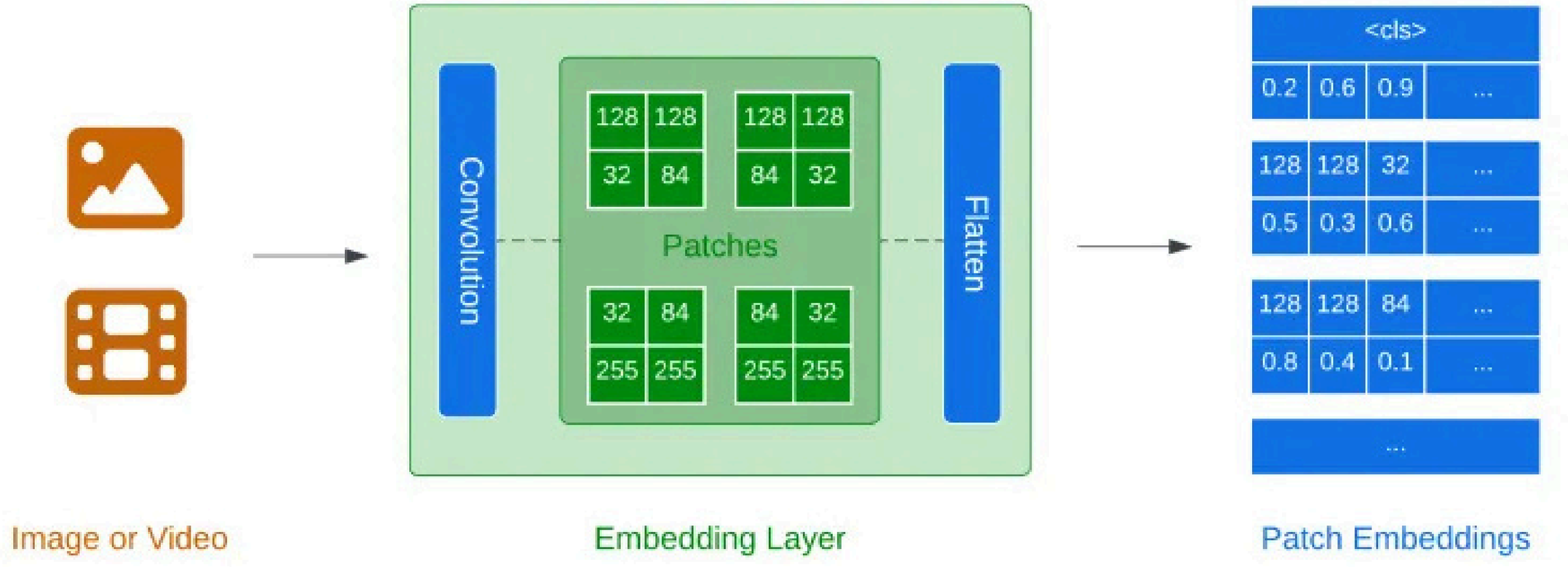
- Step 1 (Patching): Divide image into fixed-size patches (e.g., 16 x 16).
- Step 2 (Flattening):
  - A 16 x 16 RGB patch has 16 x 16 x 3 = 768 values.
  - It is flattened into a 1D vector of size 768.
- Step 3 (Linear Projection): This vector is multiplied by a dense layer (learnable) to create the Patch Embedding.

The image is now a sequence of vectors, exactly like word embeddings in NLP.

Rumoh Coding

Take a Closer Look at Vector Embeddings

Rumoh Coding



https://medium.com/@achronus/an-exploratory-look-at-vector-embeddings-8acadba24f7b

Rumoh Coding

The "Patch Embedding" Layer in Code

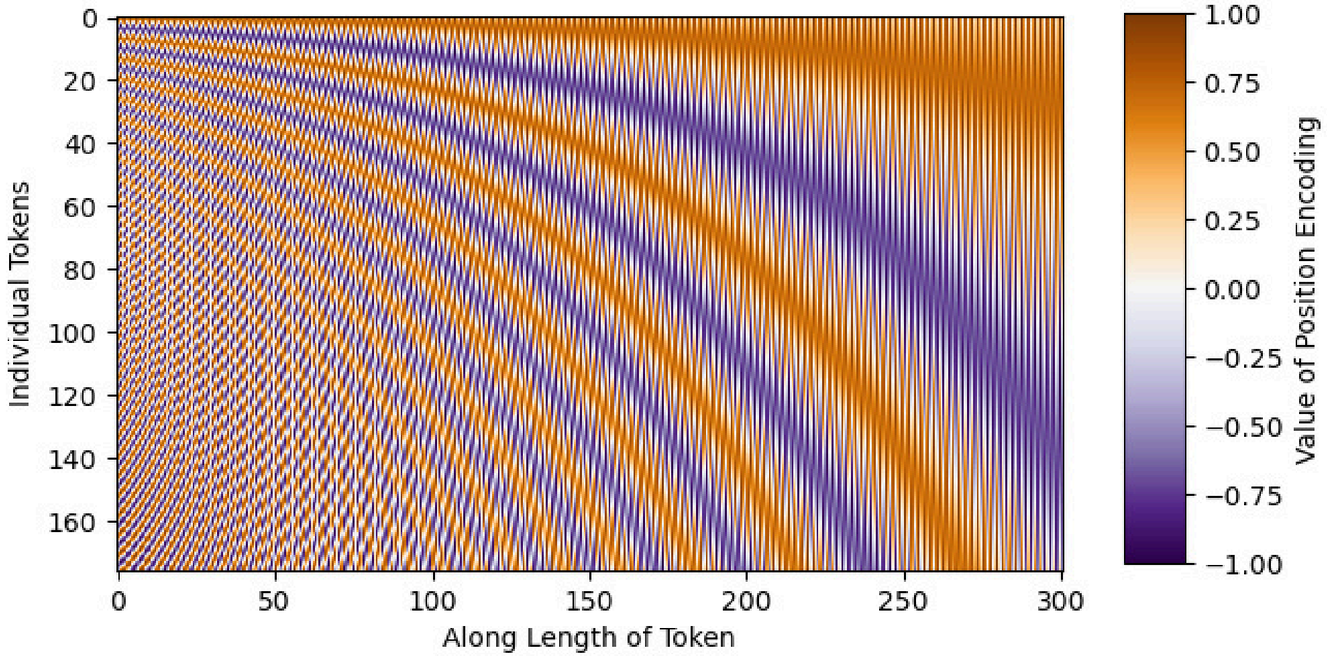
```python
class PatchEmbed(nn.Module):
    def __init__(self, img_size=224, patch_size=16, in_chans=3, embed_dim=768):
        super().__init__()
        self.img_size = img_size
        self.patch_size = patch_size
        self.n_patches = (img_size // patch_size) ** 2

        self.proj = nn.Conv2d(
            in_chans,
            embed_dim,
            kernel_size=patch_size,
            stride=patch_size
        )

    def forward(self, x):
        # x shape: [Batch, Channels, Height, Width]
        x = self.proj(x)        # Output: [Batch, Embed_Dim, H', W']
        x = x.flatten(2)        # Output: [Batch, Embed_Dim, N_Patches]
        x = x.transpose(1, 2)  # Output: [Batch, N_Patches, Embed_Dim]
        return x
```

Example

x => [1, 768, 14, 14]
x => [1, 768, 196]
x => [1, 196, 178]



https://towardsdatascience.com/position-embeddings-for-vision-transformers-explained-a6f9add341d5/

Positional Embeddings: Giving Context to Chaos

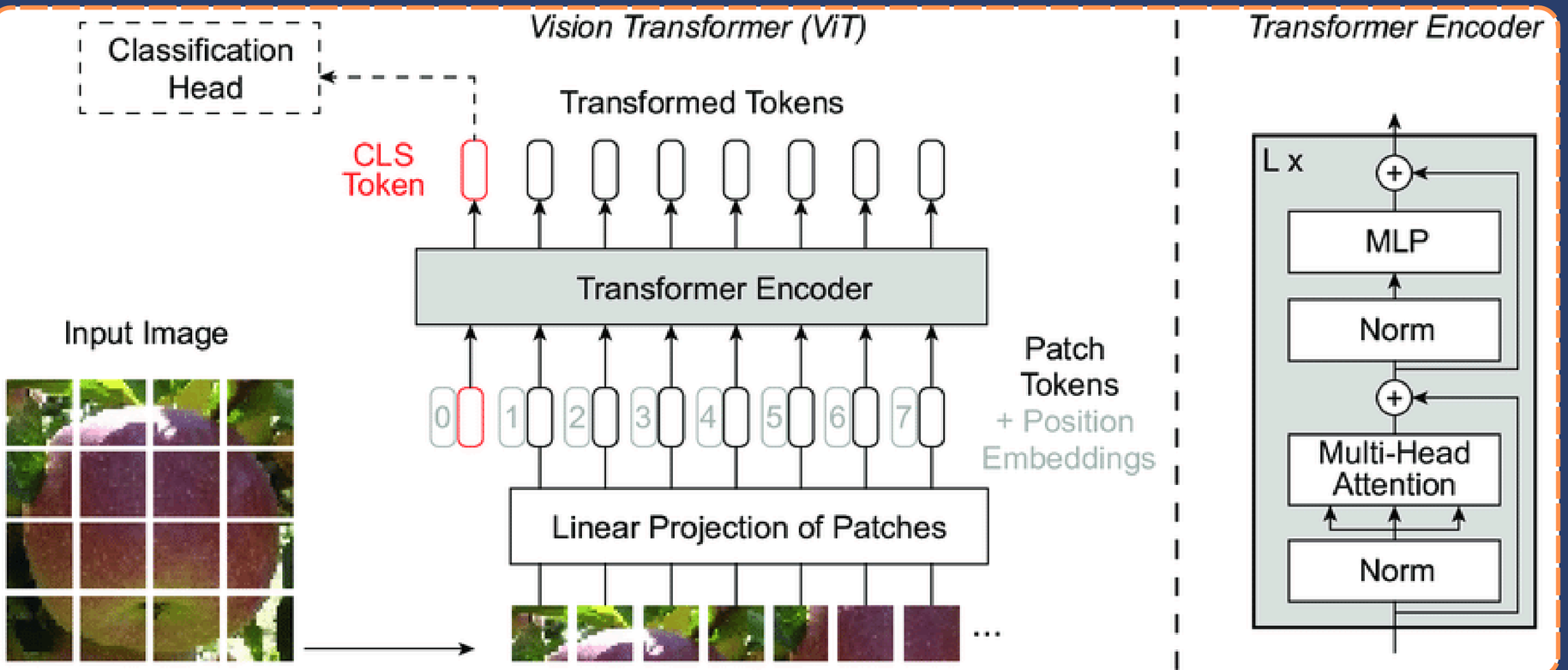
- Standard Transformers are permutation invariant.
  - If you shuffle the words in a sentence without positional encoding, a Transformer wouldn't know the difference.
  - This condition also true form image. If you shuffle the patches of a face, it's no longer a face.
- Add a learnable vector to each patch embedding that represents its location.

Rumah Coding

The [CLS] Token: Where is the Answer?

- In a CNN, we usually Global Average Pool the final feature map to get a classification.
- ViT uses different approach, the [CLS] token (borrowed from BERT).

It serves as a neutral aggregation point for the final classification head.



- **Mechanism:**
  - Prepend a special learnable token ([CLS]) to the start of the sequence.
  - This token interacts with all other patches via Self-Attention.
  - By the final layer, this token has "gathered" information from the entire image.

https://www.researchgate.net/figure/Model-overview-of-a-vision-transformer-An-input-image-is-split-into-patches-and-embedded_fig1_361923153

Rumoh Coding

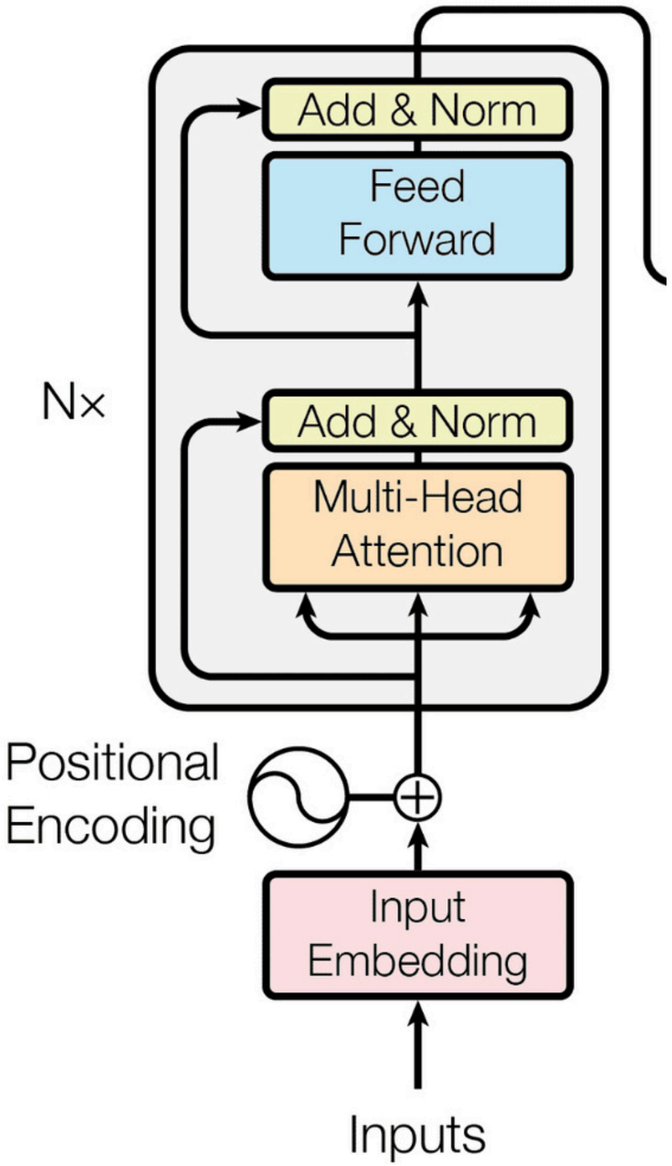
CLS Token + Positional Embeding in Code

```
self.patch_embed = PatchEmbed(img_size, patch_size, 3, embed_dim)

# Class Token (Trainable Parameter)
self.cls_token = nn.Parameter(torch.zeros(1, 1, embed_dim))

# Positional Embedding
self.num_patches = self.patch_embed.n_patches
self.pos_embed = nn.Parameter(torch.zeros(1, self.num_patches + 1,
embed_dim))
# Transformer Encoder Blocks
self.blocks = nn.ModuleList([
  Block(embed_dim, num_heads) for _ in range(depth)
])

# Classification Head
self.norm = nn.LayerNorm(embed_dim)
self.head = nn.Linear(embed_dim, num_classes)
```

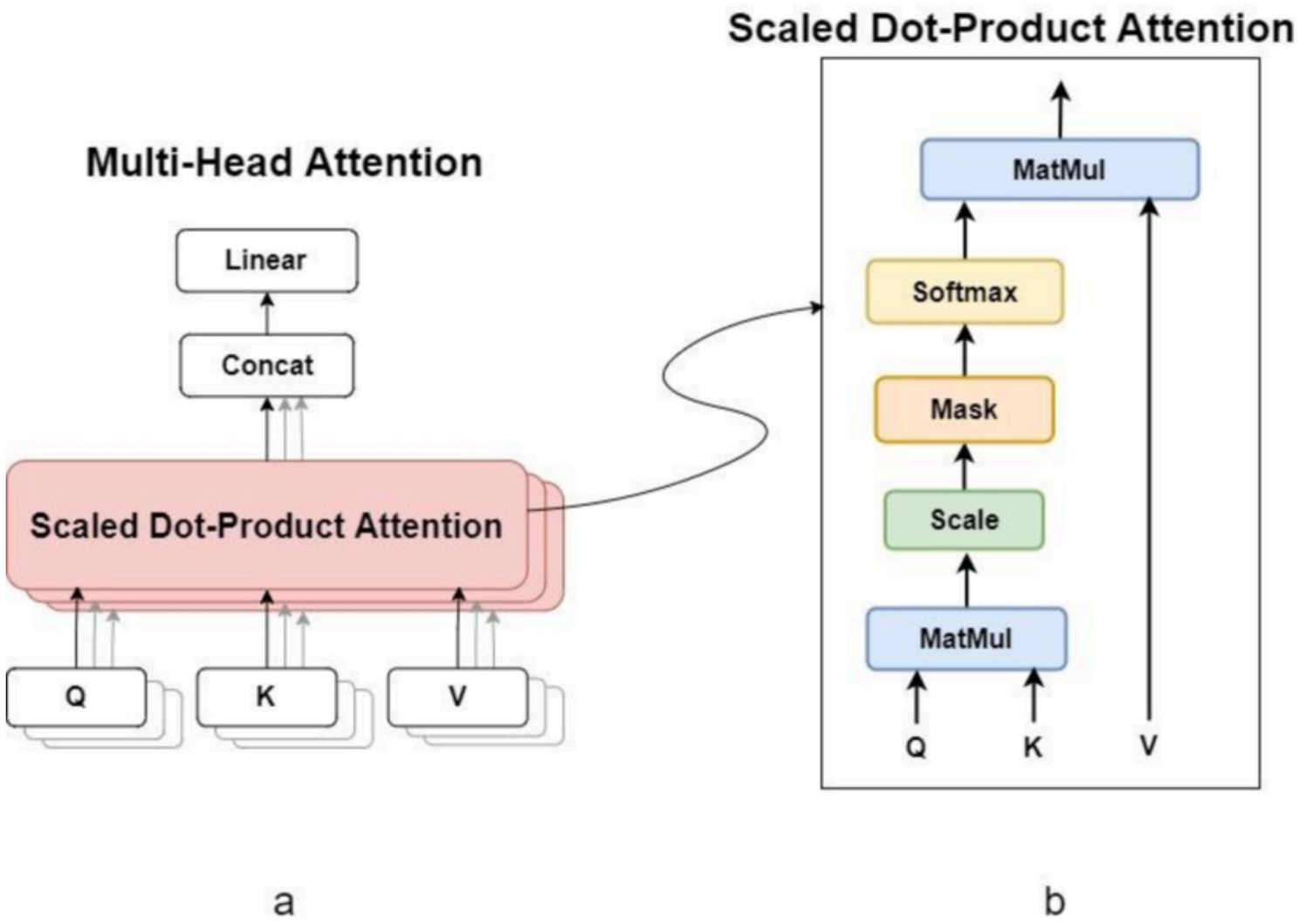


https://machinelearningmastery.com/the-transformer-model/

The Encoder: Don't Reinvent the Wheel

- **ViT** used the exact same encoder used in NLP.
- It consists:
  - **Layer Norm**: For training stability (applied before attention in ViT).
  - **MSA (Multi-Head Self-Attention)**: The "Secret Sauce" (global mixing).
  - **MLP (Multi-Layer Perceptron)**: Per-location processing (GELU activation).
  - **Residual Connections**: To allow deep training.

Rumoh Coding



https://www.researchgate.net/figure/Multi-head-attention-mechanism-In-the-encoder-and-decoder-multiple-attention-heads-are_fig2_372074536

Multi-Head Attention: Many Perspectives at Once

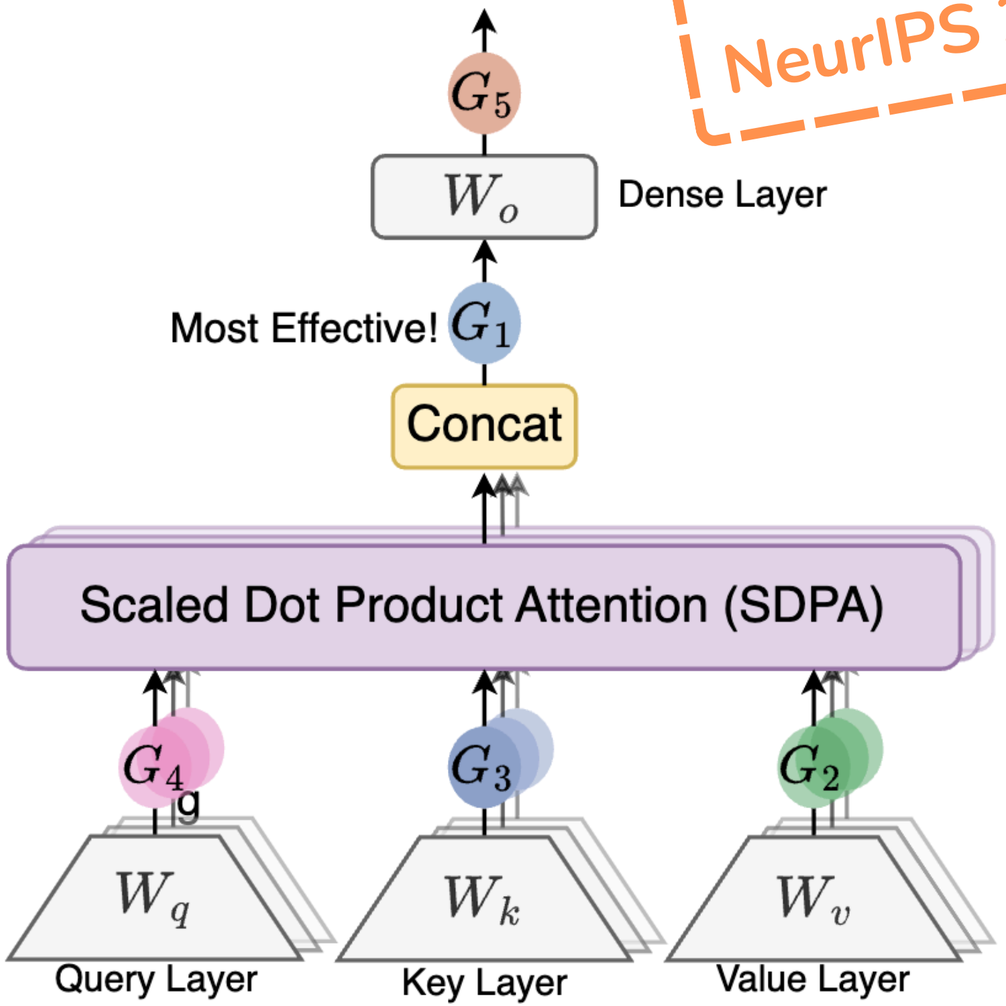
- A single attention process might get stuck focusing on just one thing (e.g., grouping pixels only by color).
- Run the attention mechanism multiple times in parallel!
  - Head 1 might learn geometry.
  - Head 2 might learn texture.
  - Head 3 might learn long-range semantic relationships.

The results are concatenated, giving a rich, diverse understanding of the image

Rumoh Coding

BREAKING NEWS

Best Paper
NeurIPS 2025



Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free

- The authors apply head-specific sigmoid gate applied immediately after the Scaled Dot-Product Attention (SDPA).
- This method applies query-dependent sparse gating scores to modulate the attention output.

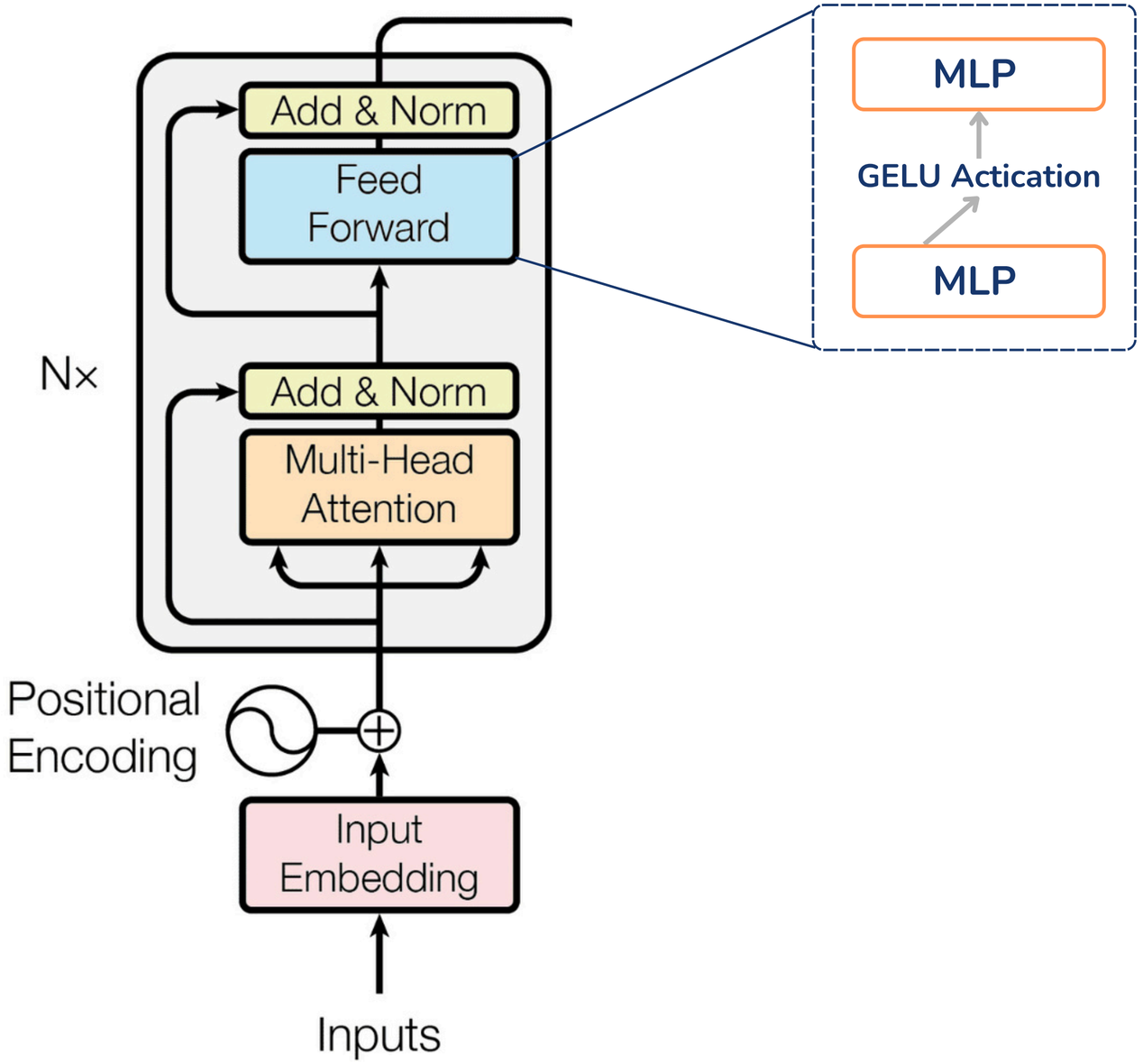
$$Y' = Y \odot \sigma(XW_\theta)$$

- The result are Performance Improvements, Training Stability and Context Length Extension.

It enhances non-linearity and introduces sparsity that removes attention sinks, leading to "attention-sink-free" models that are more stable and easier to scale.

Qiu, Zihan, et al. "Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free." arXiv preprint arXiv:2505.06708 (2025).

Rumoh Coding

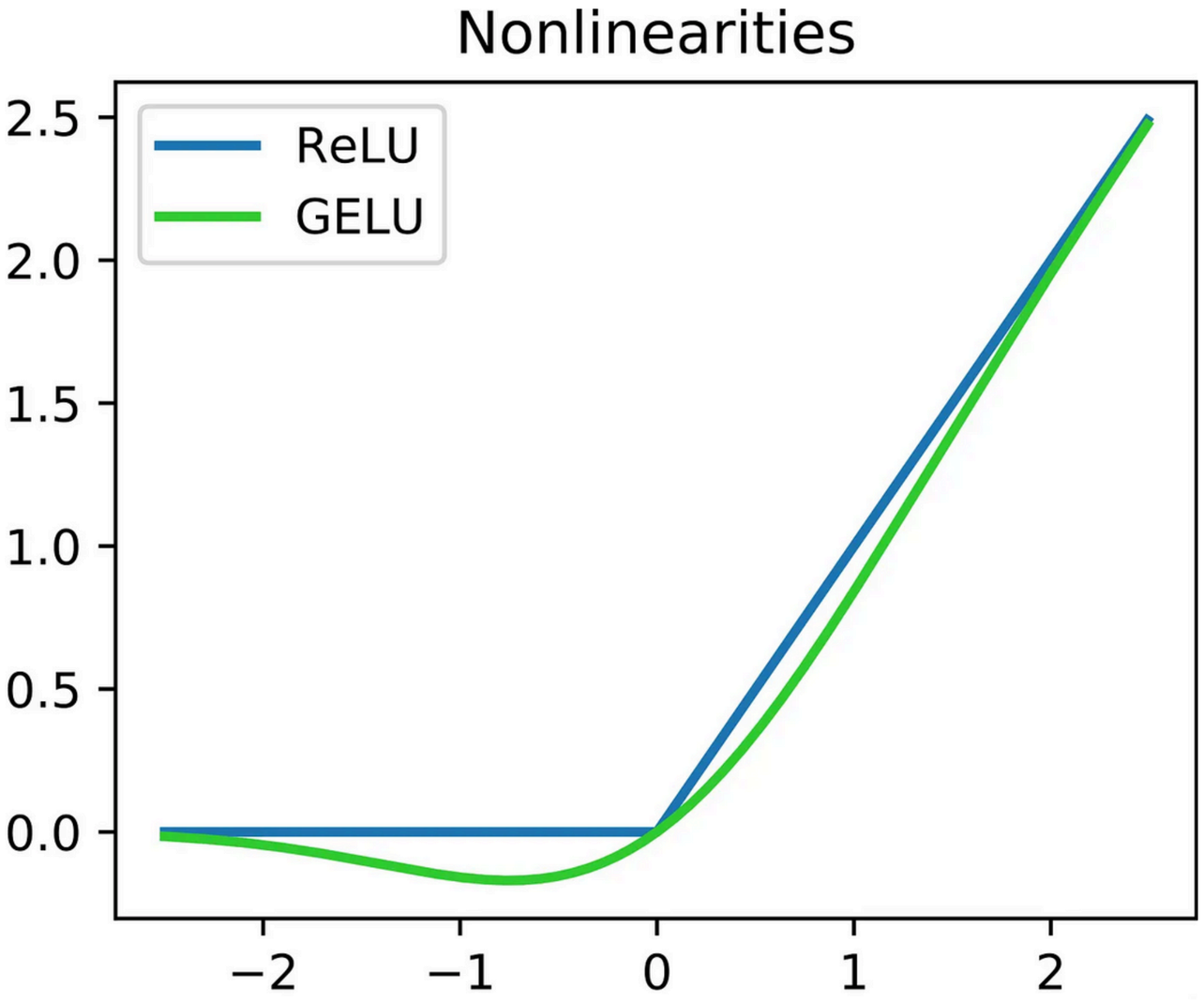


https://machinelearningmastery.com/the-transformer-model/

The MLP – The "Brain" of the Patch

- Self-Attention allows patches to talk to each other and share context, then MLP processes that shared information individually for each patch.
- Feed Forward consists of two Linear Layers with a GELU activation in between.
- It projects the data to a higher dimension (usually 4x larger) and then back down e.g 768 => 3072 => 768.
- Why expand? To give the model "space" to untangle complex features and relationships learned during Attention.

Rumoh Coding



Why GELU? (The Smoother ReLU)

- GELU is Gaussian Error Linear Unit.
- RELU is harsh. If a value is negative, the gradient dies completely ("Dead ReLU").
- GELU is smooth and probabilistic. It allows small negative values to pass through briefly.
- The smoothness helps optimization in very deep networks (like Transformers).

https://medium.com/@coffee_and_notes/deep-learning-gelu-gaussian-error-linear-unit-activation-function-56168dd5997

Rumoh Coding

ViT proves that Architecture implies Priors is not strictly necessary. A general-purpose architecture can learn vision if given enough data.

The Clash

CNN vs. ViT

Rumoh Coding

Tunnel vs. Eagle Eye

CNN

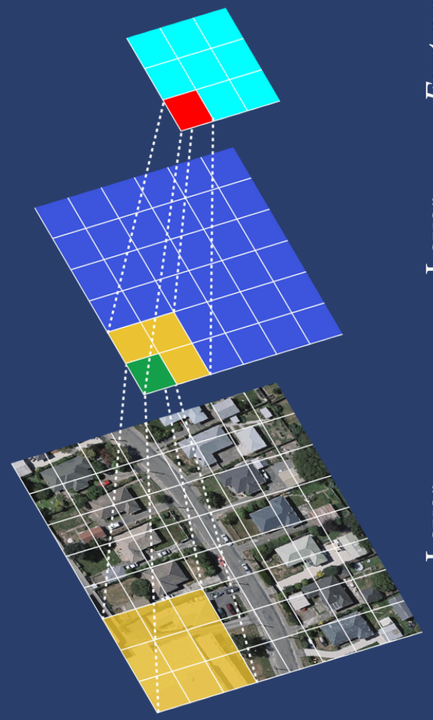
- Starts with a **small** receptive field (local features like edges/corners).
- Slowly **expands** to global context through pooling and stacking layers.
- It is like reading a book through a magnifying glass.

ViT

- Global Attention from Layer 1.
- The top-left pixel can attend to the bottom-right pixel immediately.
- It is like seeing the whole page at once.

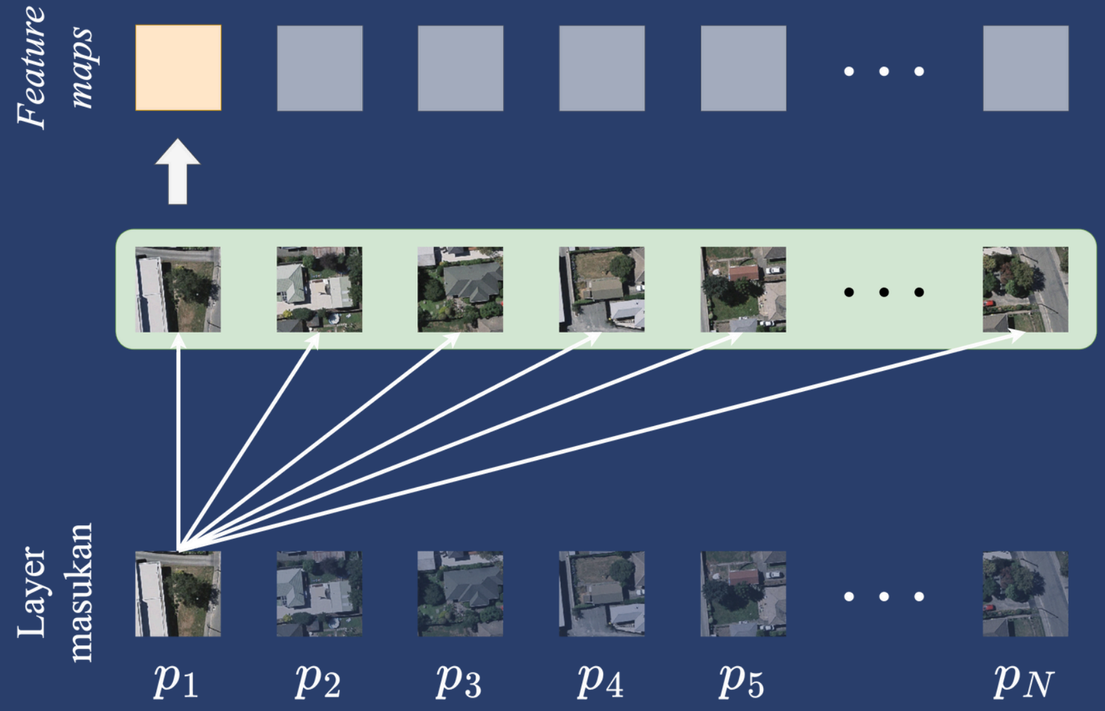
CNN

$$r_i = r_{i-1} + (k-1)\prod_{k=1}^{i-1} s_k$$



ViT

$$r_i = N$$



Rumah Coding

Built-in Knowledge vs. Freedom

CNNs have High Inductive Bias

- They are "hard-coded" to assume **locality** and **translation** invariance.
- Great for **small** data; trains faster.
- The model is **constrained** by these assumptions.

ViTs have Low Inductive Bias

- They make almost **no assumptions** about image structure.
- Higher ceiling. If given enough data, it can **learn** patterns that CNNs cannot (e.g., relationships between distant objects).
- Harder to train; prone to **overfitting** on small data.

ViT is a "Data Hungry" beast.

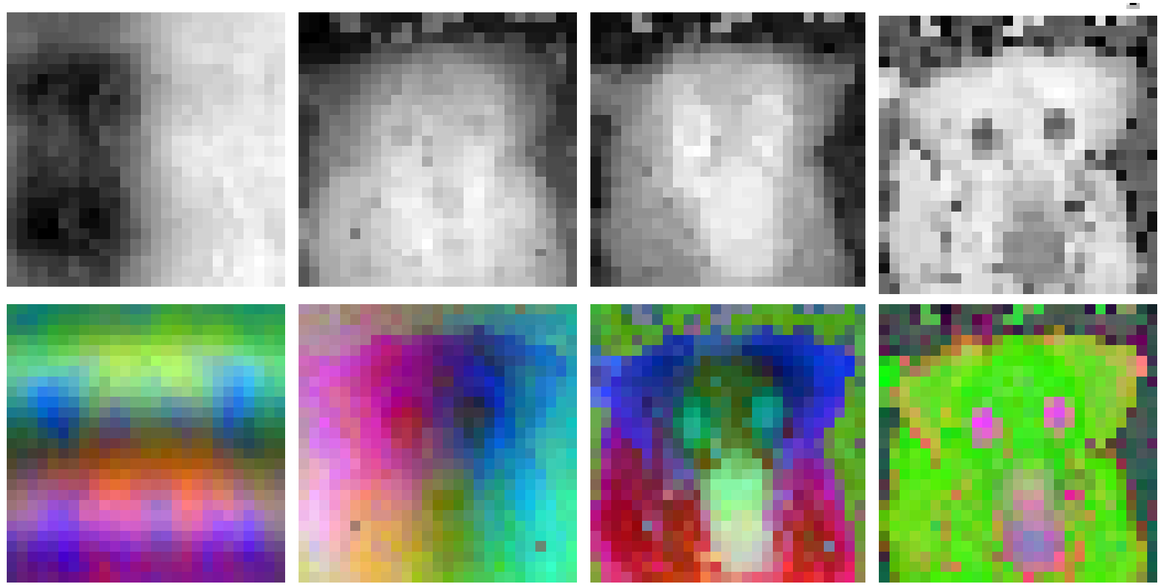
When does ViT win?

- Small Datasets (ImageNet-1k / CIFAR):
  - CNNs outperform ViTs.
  - ViT overfits because it lacks the inductive bias to regularize learning.
- Massive Datasets (JFT-300M):
  - ViT overtakes CNNs.
  - With enough data, ViT learns better "rules" than the ones hard-coded into CNNs.

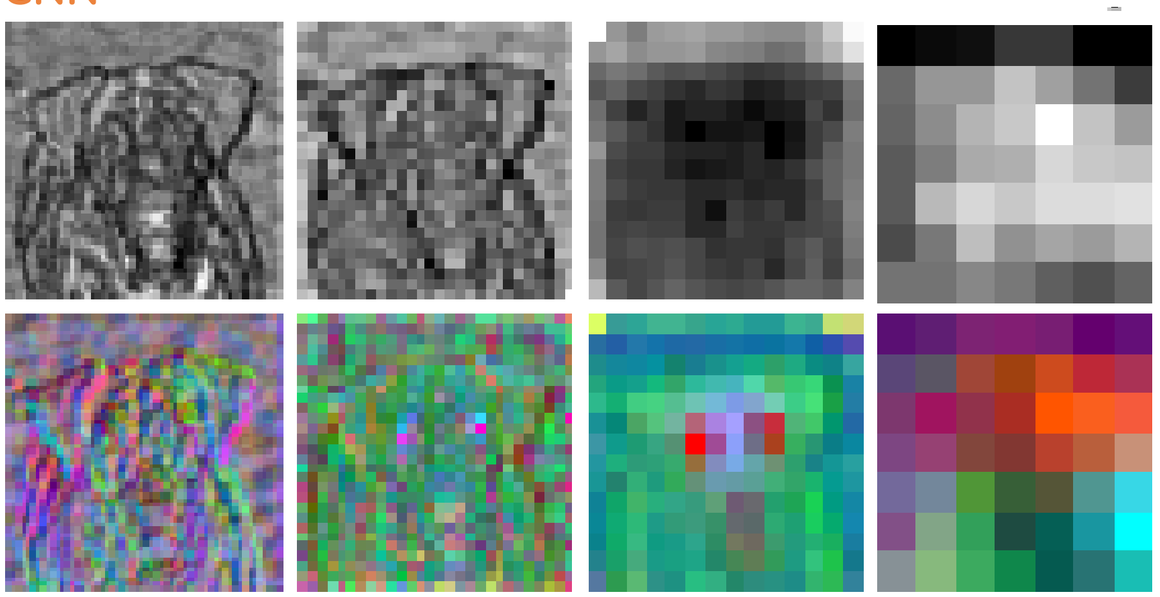
https://datasets-dev.10web.me/docs/ml/datasets/cifar-10-dataset/

Rumoh Coding

ViT



CNN



Shallow → Deep

How Do They "Think"?

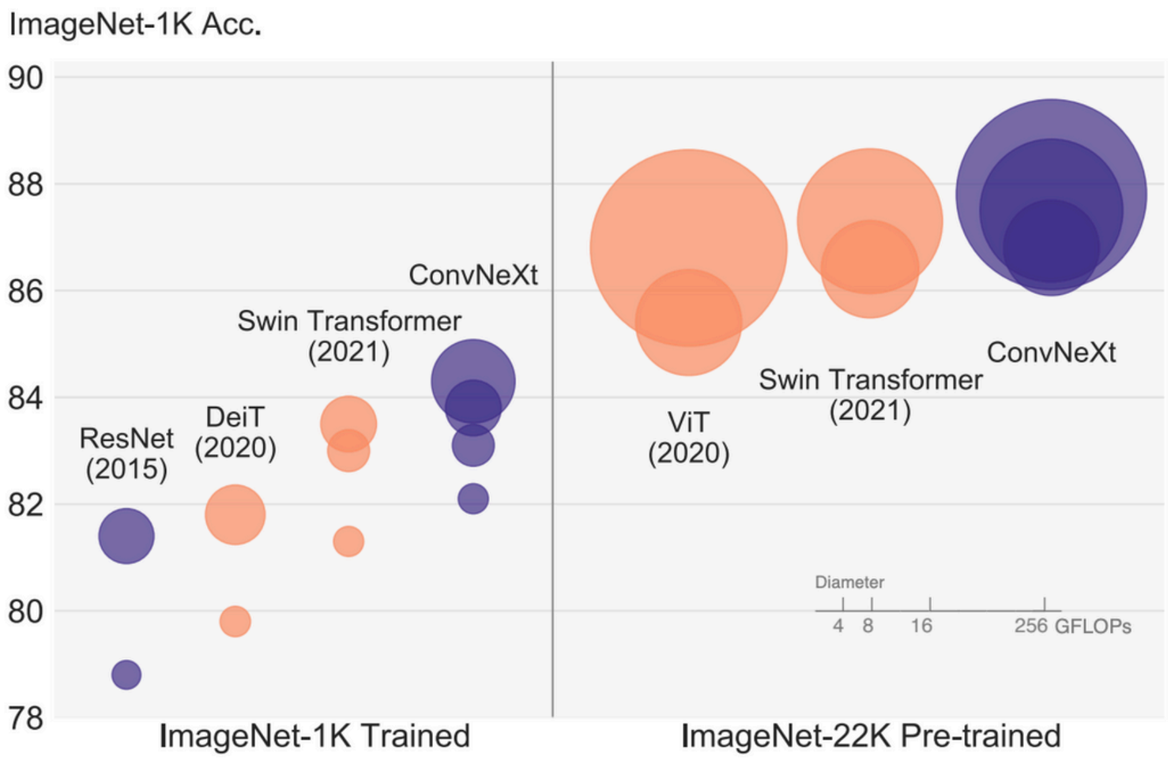
- Attention Distance:
  - In deep layers, both CNN and ViT look globally.
  - In Layer 1, ViT already has heads that look at the whole image, mixing global context immediately.
- What they focus on:
  - CNNs tend to focus on textures and high-frequency patterns first.
  - ViTs tend to focus on semantic shapes and objects much earlier in the network.

Attention maps allow us to visualize exactly which parts of the image contribute to the decision.

https://www.researchgate.net/figure/Deep-features-visualization-via-PCA-Applied-on-a-ViTs-and-b-CNN-ResNet-models-each_fig2_357014516

Rumoh Coding

The Verdict & The Future



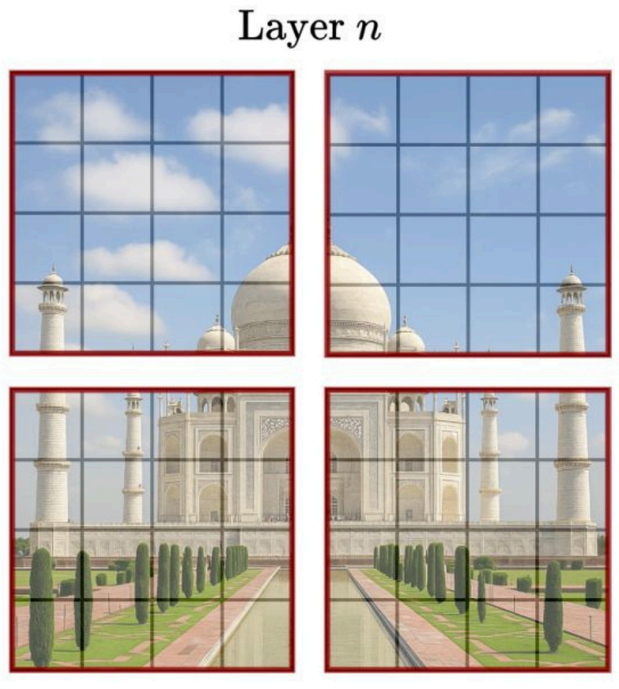
https://github.com/facebookresearch/ConvNeXt

Is Convolution Dead?

- Researchers asked: "What if we modernize a standard ResNet using Vision Transformer training techniques?"
- The result is ConvNeXt. A pure CNN that matches or beats Swin Transformers on ImageNet.

The "magic" wasn't just attention; it was also modern training recipes (data augmentation, optimizers, etc.) that ViT introduced. CNNs are still highly relevant.

Rumoh Coding



The Future is Hybrid

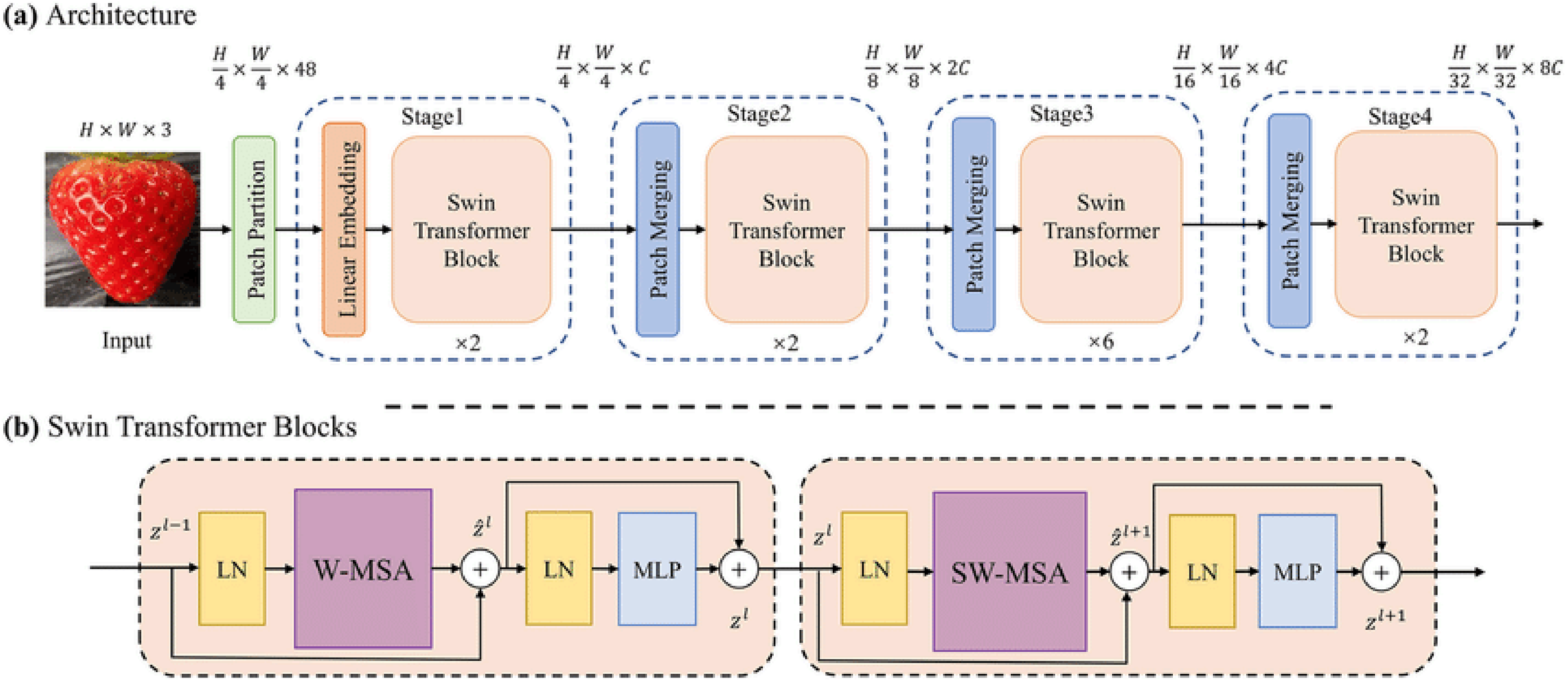
- We need both.
  - ViT's strength: Global context and scaling.
  - CNN's strength: Local feature extraction and data efficiency.
- Swin Transformer (Hierarchical ViT)
  - Re-introduces the concept of "Windows" (local attention) back into Transformers.
  - Processes images hierarchically (like a CNN pyramid) rather than flat patches.

The future isn't "ViT OR CNN." It is architectures that combine convolution in early layers for efficient feature extraction, and attention in later layers for global context.

https://www.linkedin.com/posts/kavishka-abeywardhana-01b891214_swin-transformer-in-the-original-vision-activity-7392862546585767936-v2jh/

Rumoh Coding

Swin Transformer Architecture



https://www.researchgate.net/figure/a-Swin-Transformer-architecture-b-Swin-Transformer-blocks_fig3_360004380

Rumoh Coding

Conclusion

Scenario / Constraint	Recommended Architecture	Why
Limited Data (< 1M images, no pre-training)	Modern CNN (e.g., EfficientNetV2, ConvNeXt)	Inductive bias helps learn faster with less data.
Massive Data (JFT-300M scale)	Vision Transformer (ViT, Data-efficient DeiT)	Higher ceiling; capacity to learn complex non-local patterns.
Edge Deployment (Mobile/IoT, low latency)	MobileNet / EfficientNet-Lite	CNNs are still far more optimized for current mobile hardware.
Dense Prediction Tasks (Segmentation, Detection)	Hybrid / Swin Transformer	Hierarchical structure is essential for multi-scale understanding.

Link to ViT Demo
(Google Colab)

s.id/VIT123

Talk is cheap, show me the code

https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010

Rumah Coding

Terima Kasih

Lhuqita Fazry