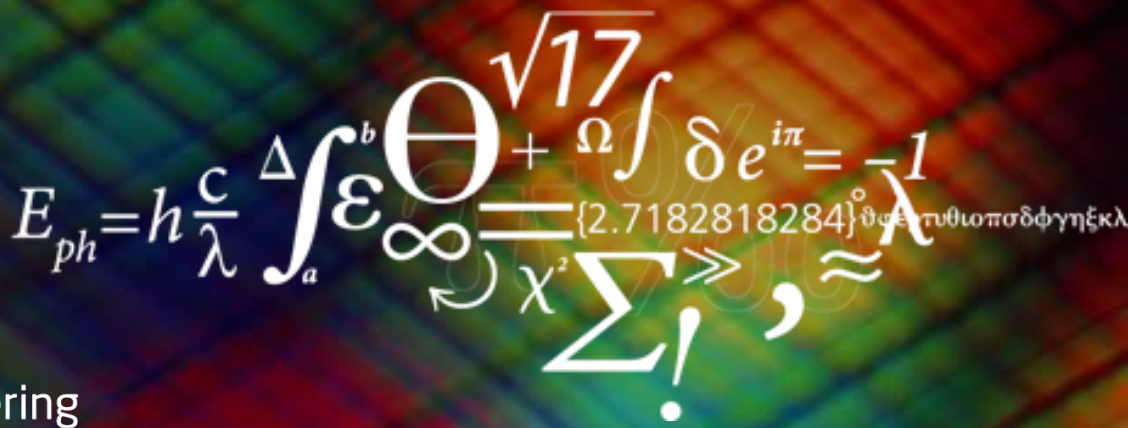
Technical University of Denmark

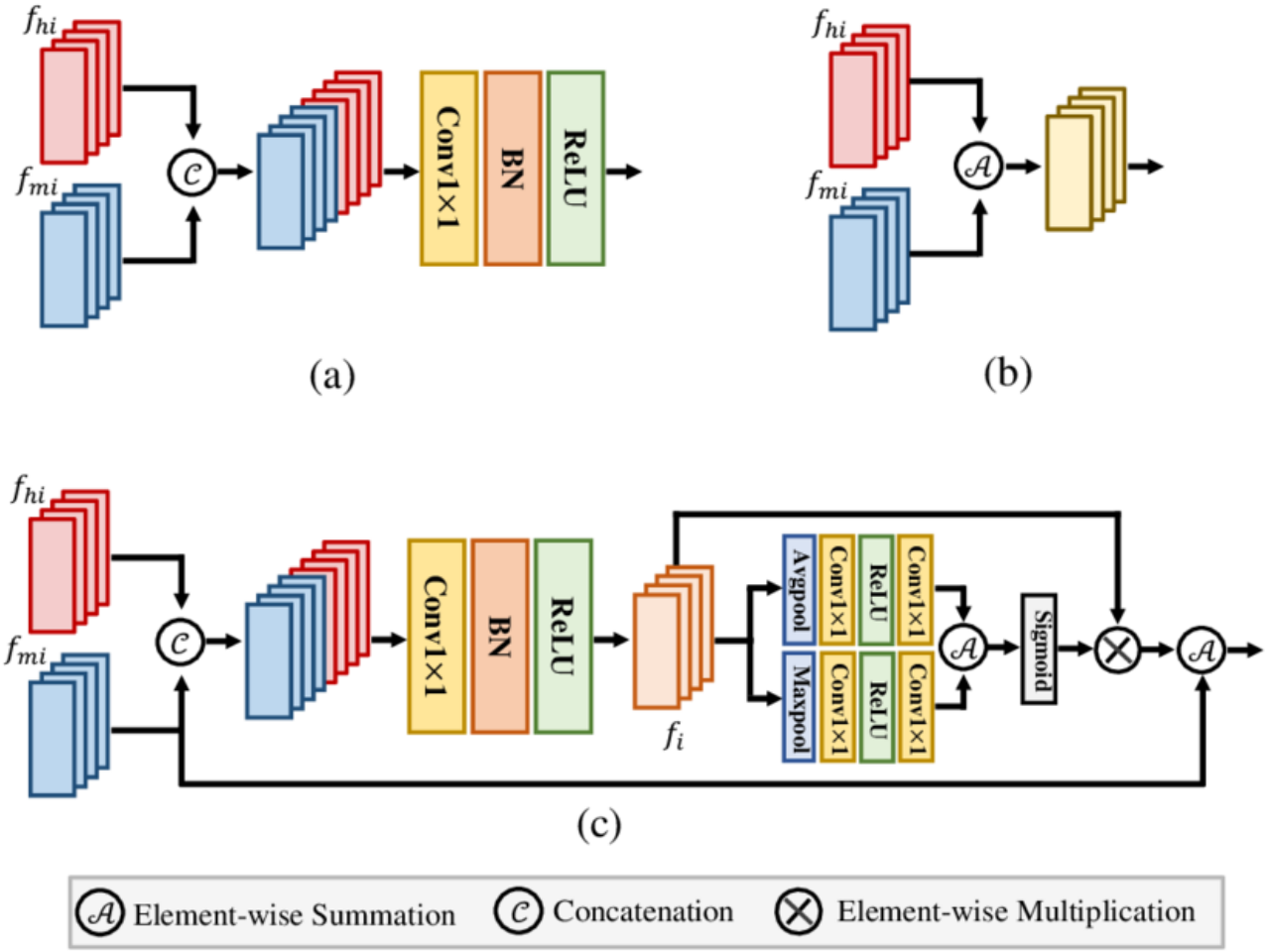
Computational Imaging and Spectroscopy: Deep learning for imaging : Advanced concepts

Thierry SOREZE
DTU July 2024

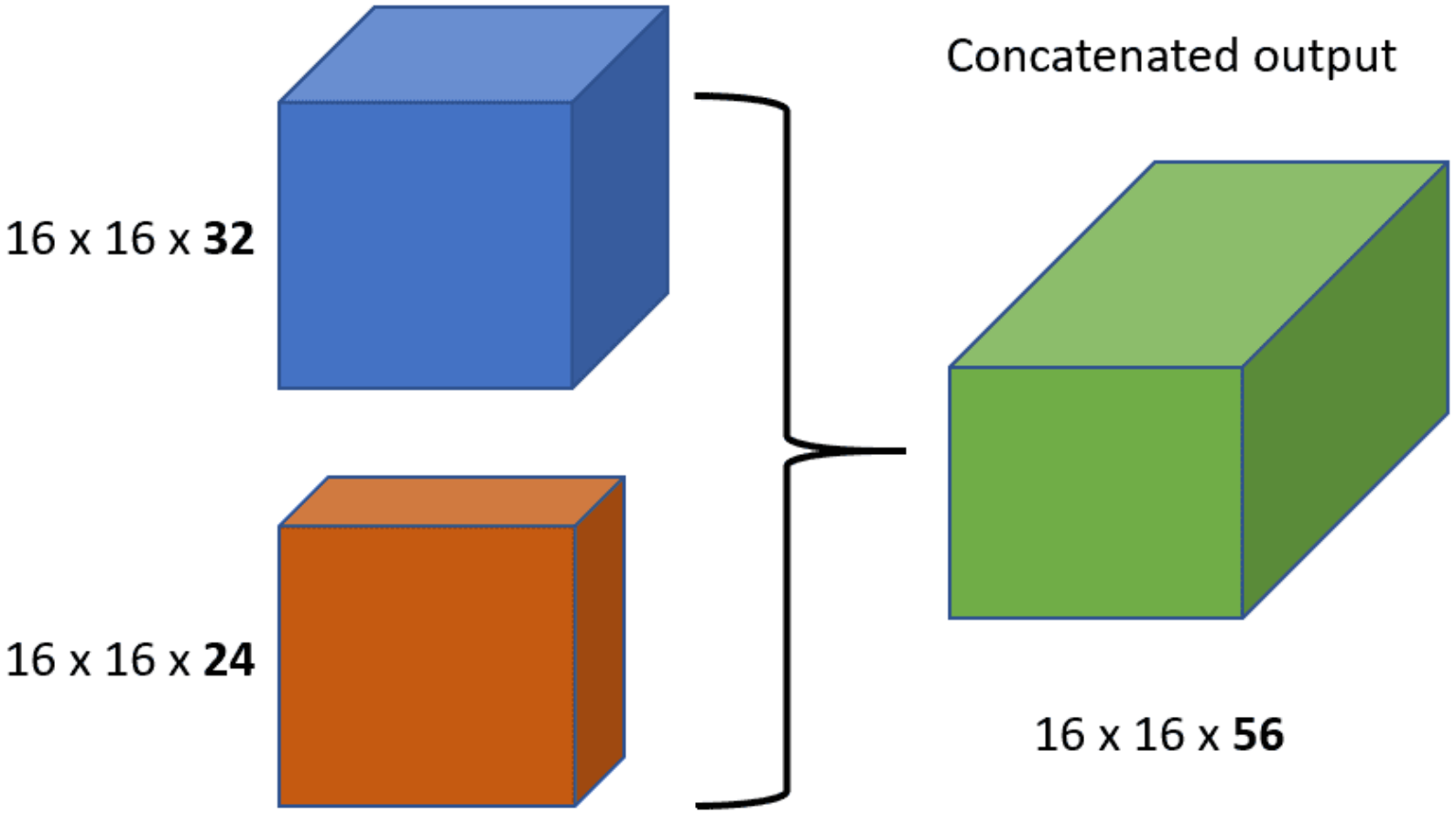


DTU Fotonik
Department of Photonics Engineering

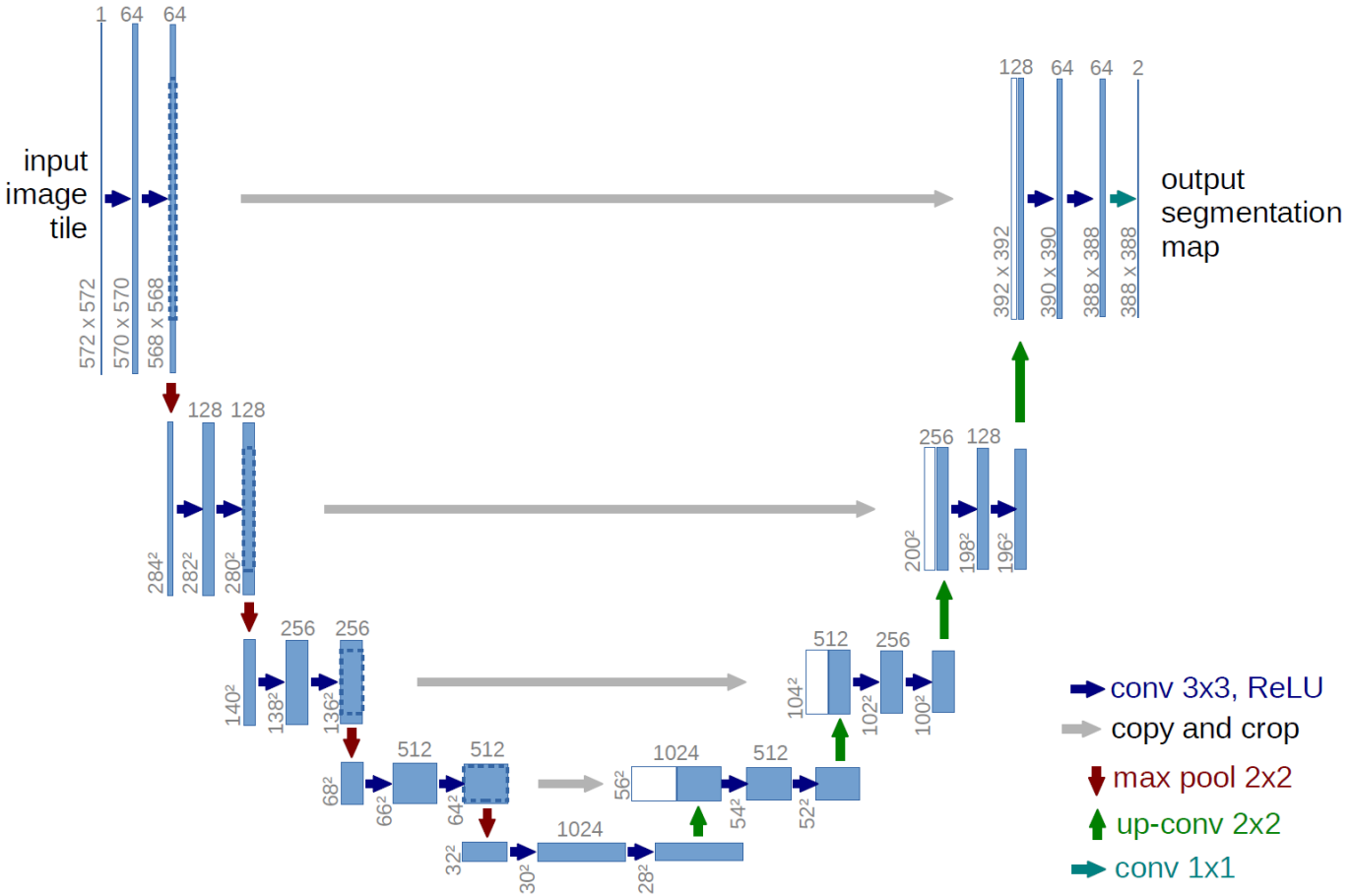
Features fusion methods



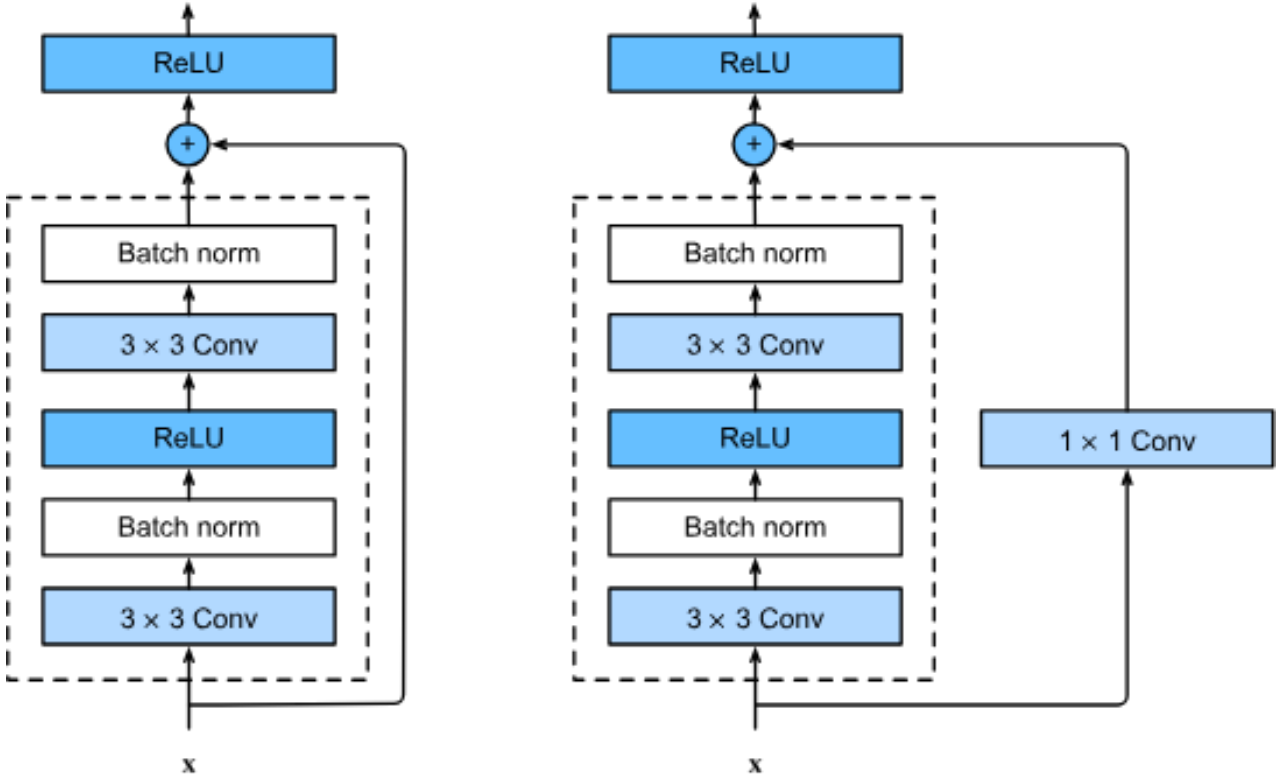
Features fusion methods



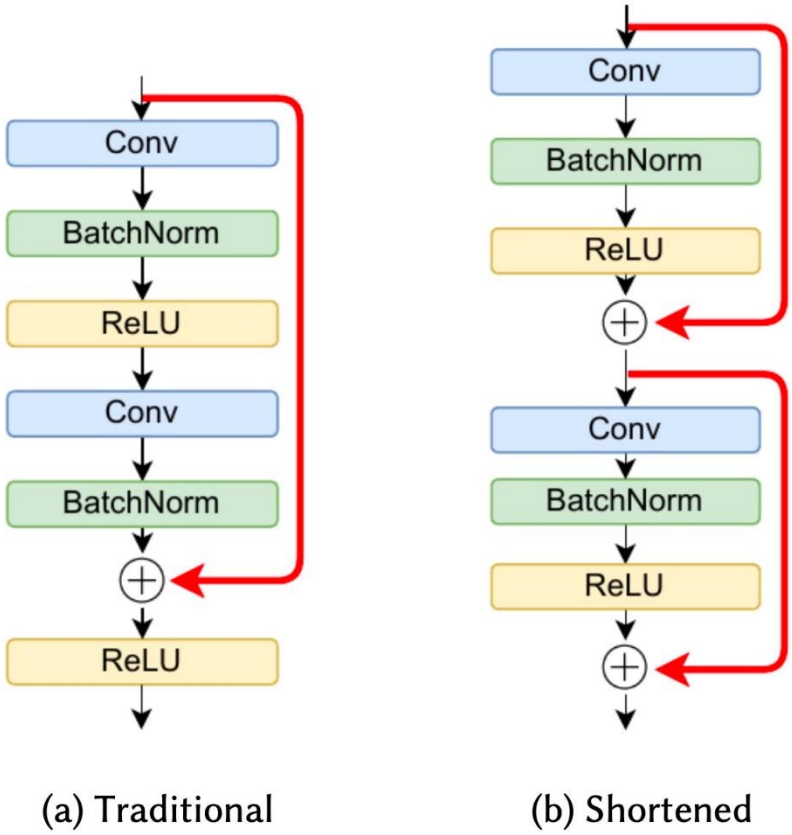
Advanced architecture: Unet



Advanced architecture: Resnet

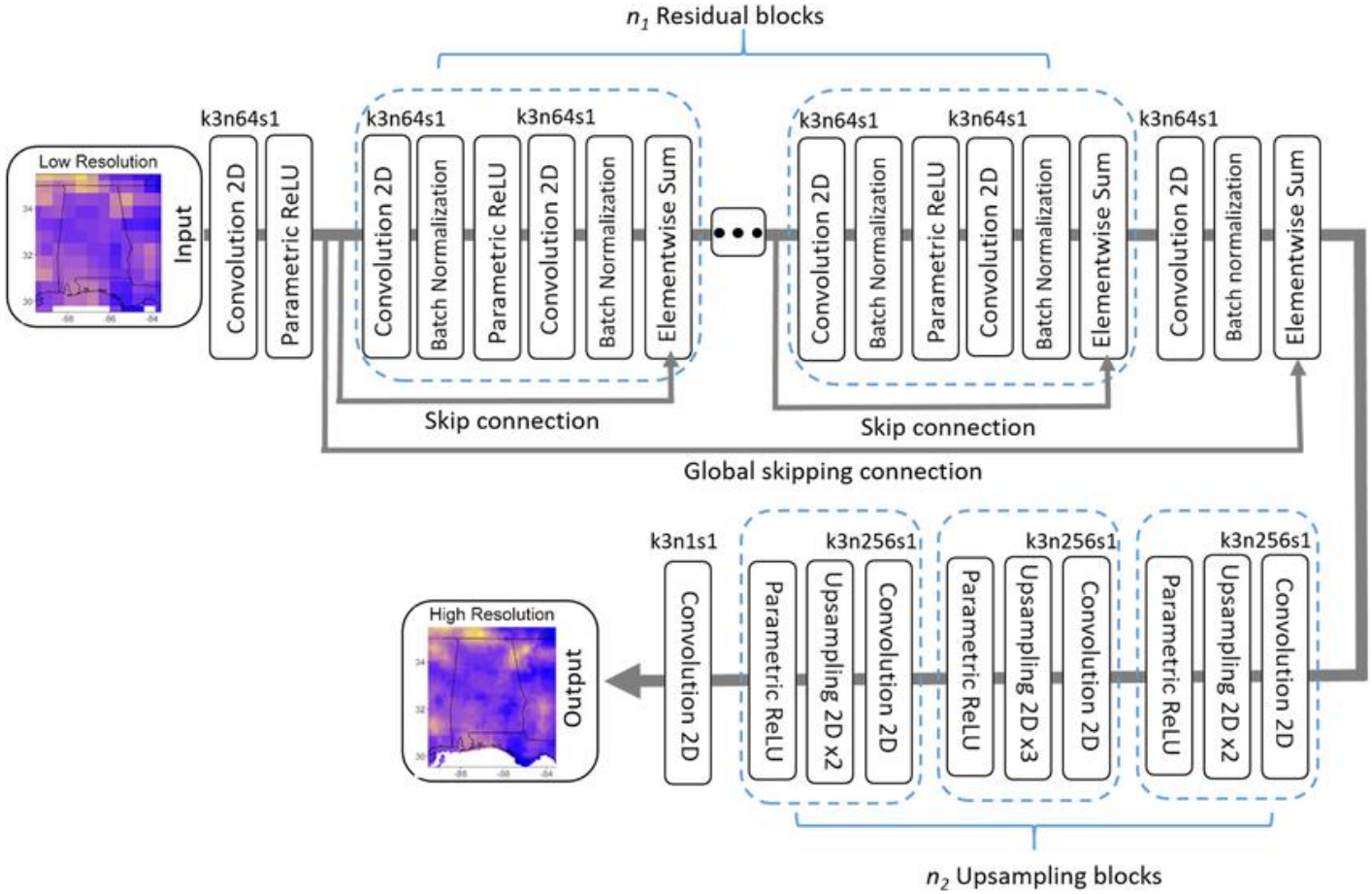


Advanced architecture: Resnet



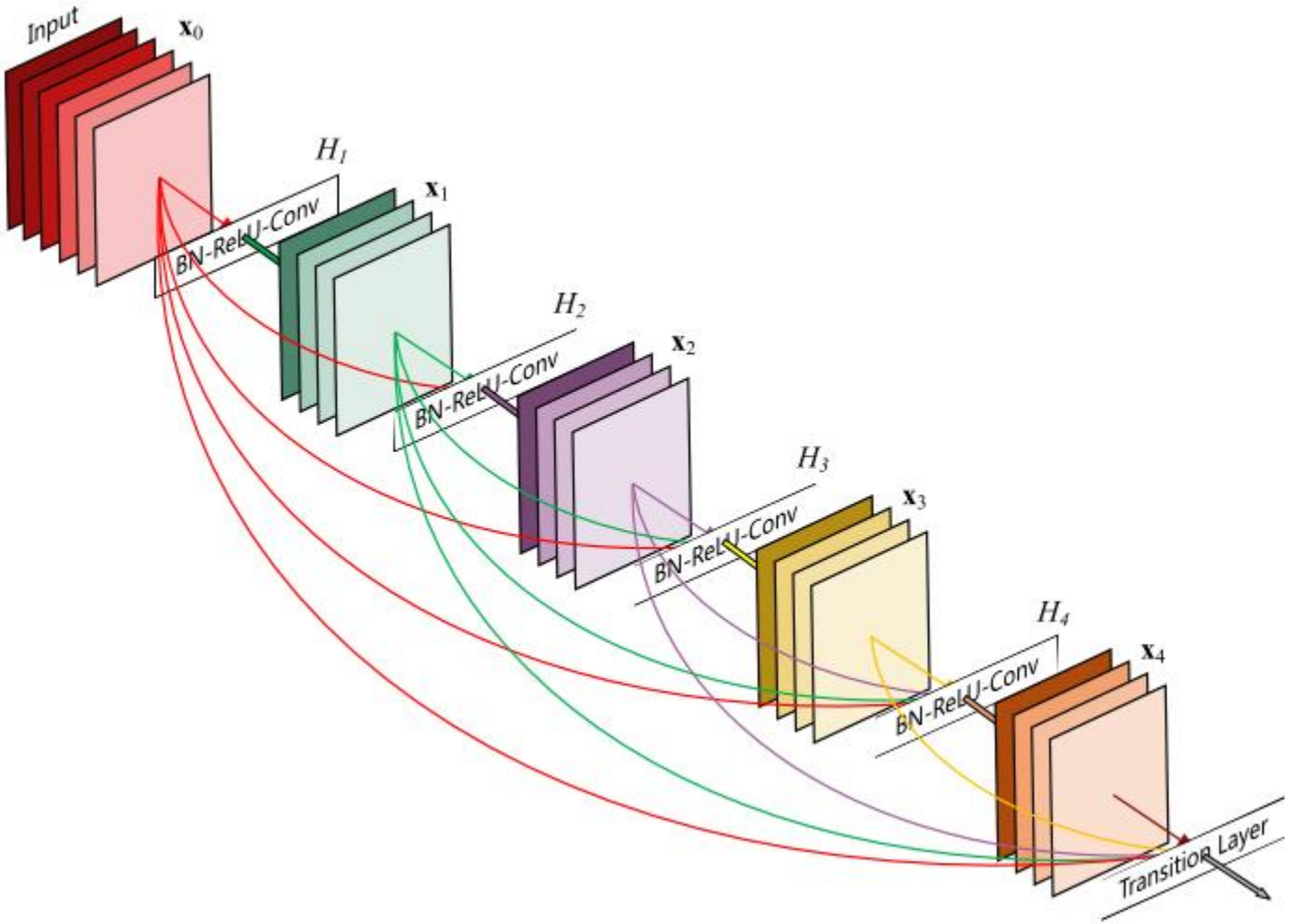
(a) Traditional (b) Shortened

Advanced architecture: Resnet

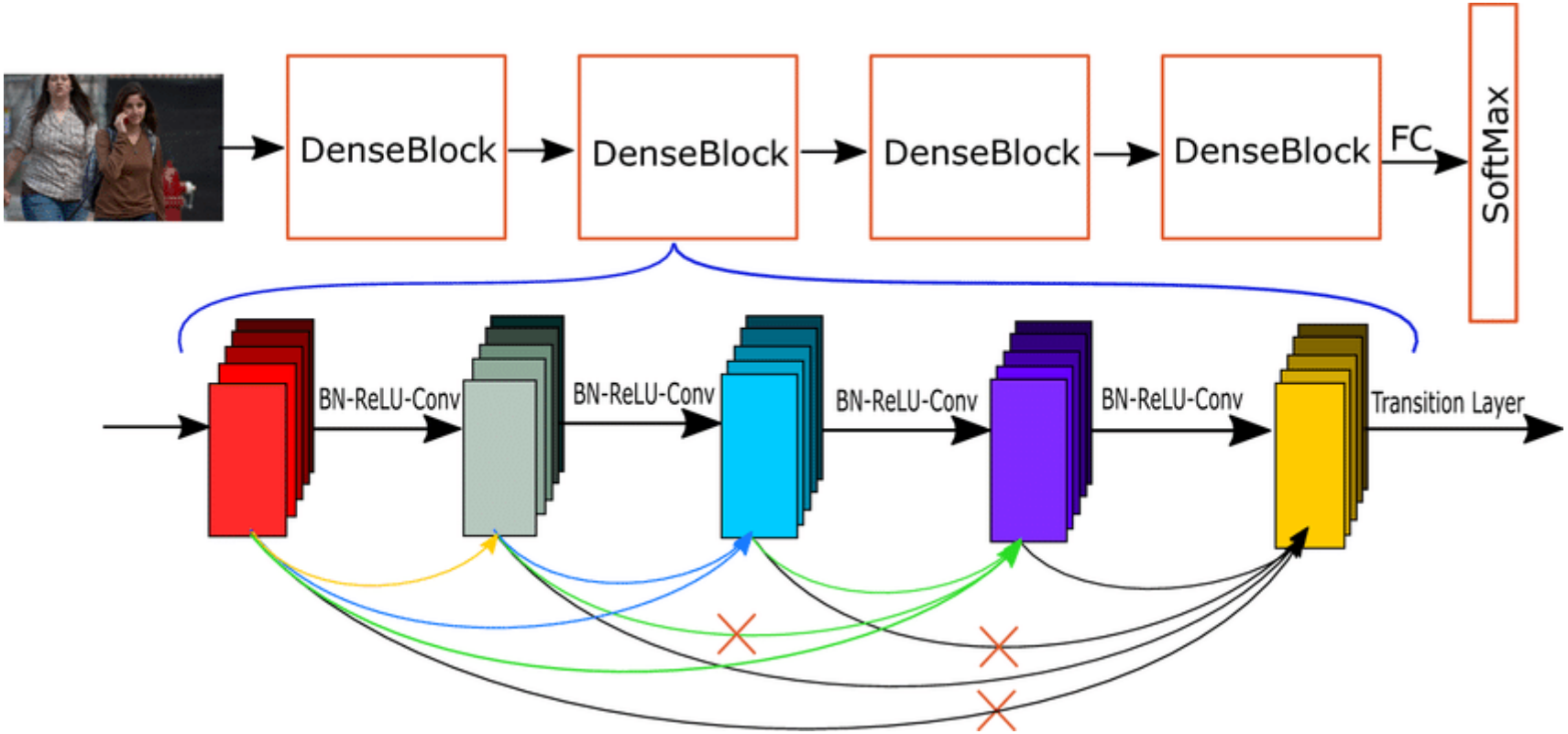


Wang, Fang, et al. "Deep learning for daily precipitation and temperature downscaling." *Water Resources Research* 57.4 (2021): e2020WR029308.

Advanced architecture: Resnet

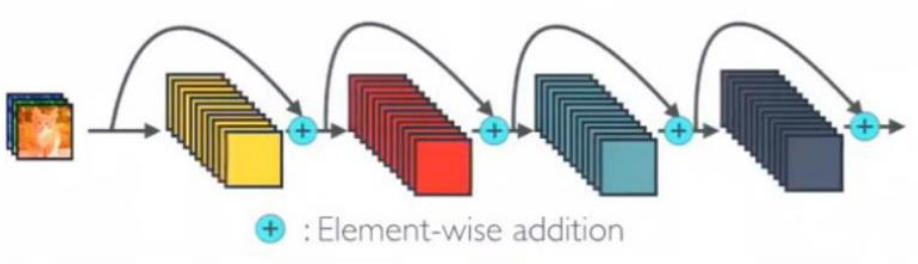


Advanced architectures: Densenet



Advanced architecture: Densenet

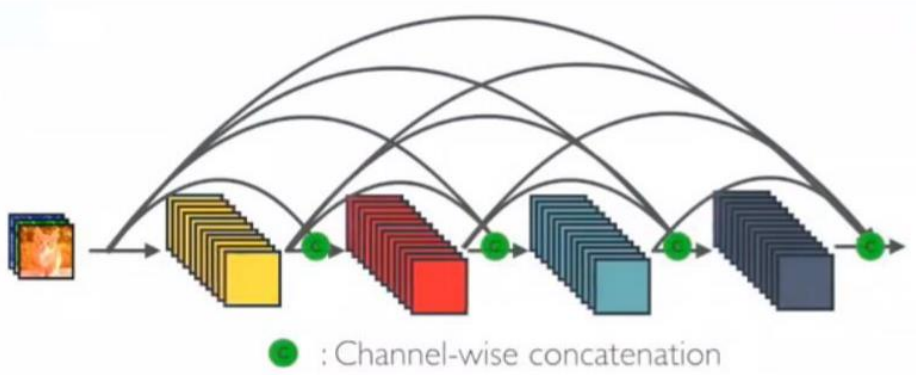
ResNet



Skip Connection

Addition

DenseNet

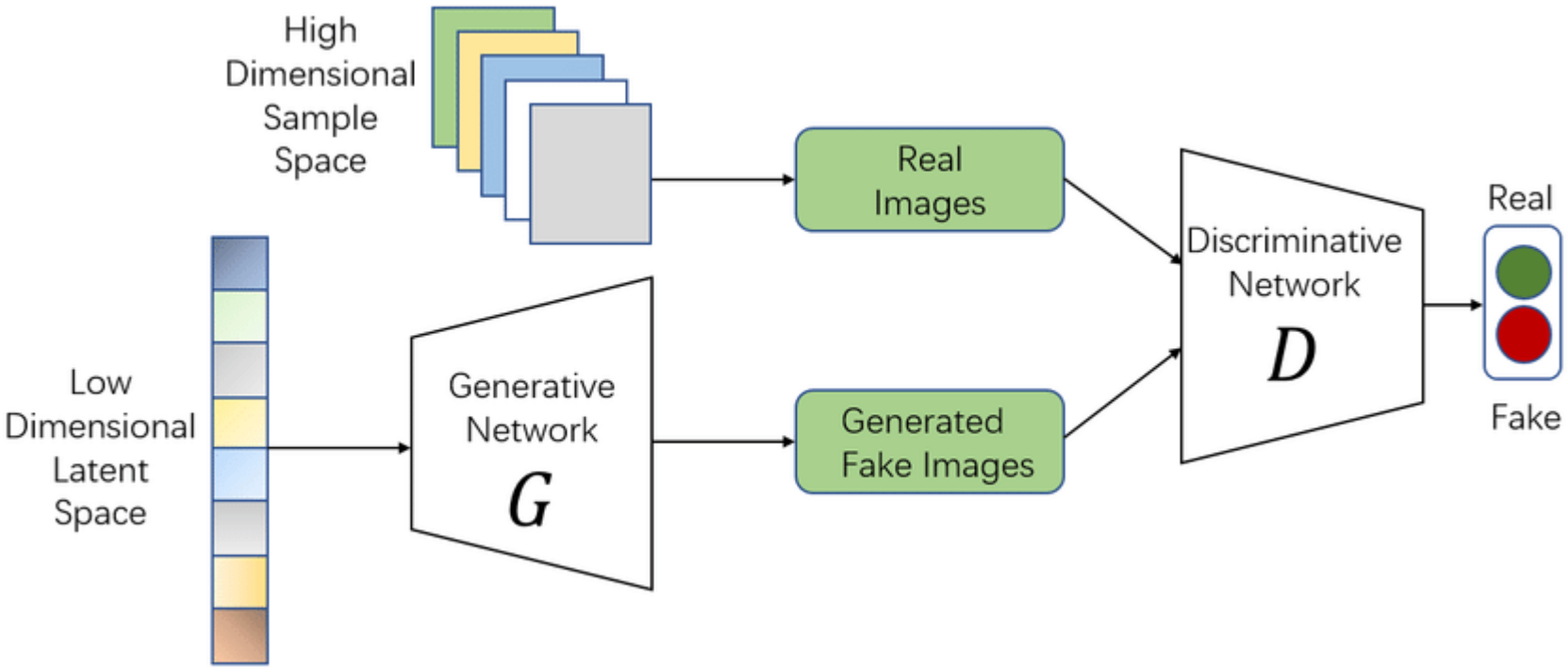


Dense Connection

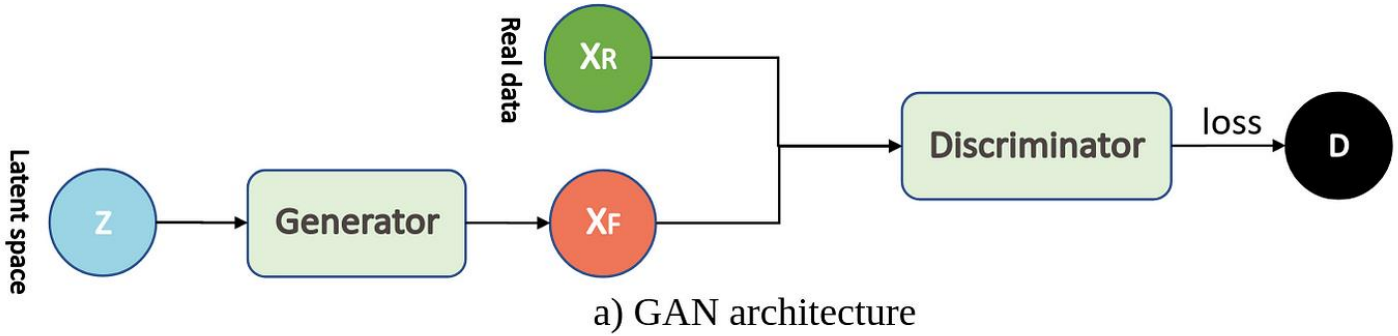
Concatenating

Generative Adversarial Networks (GANs)

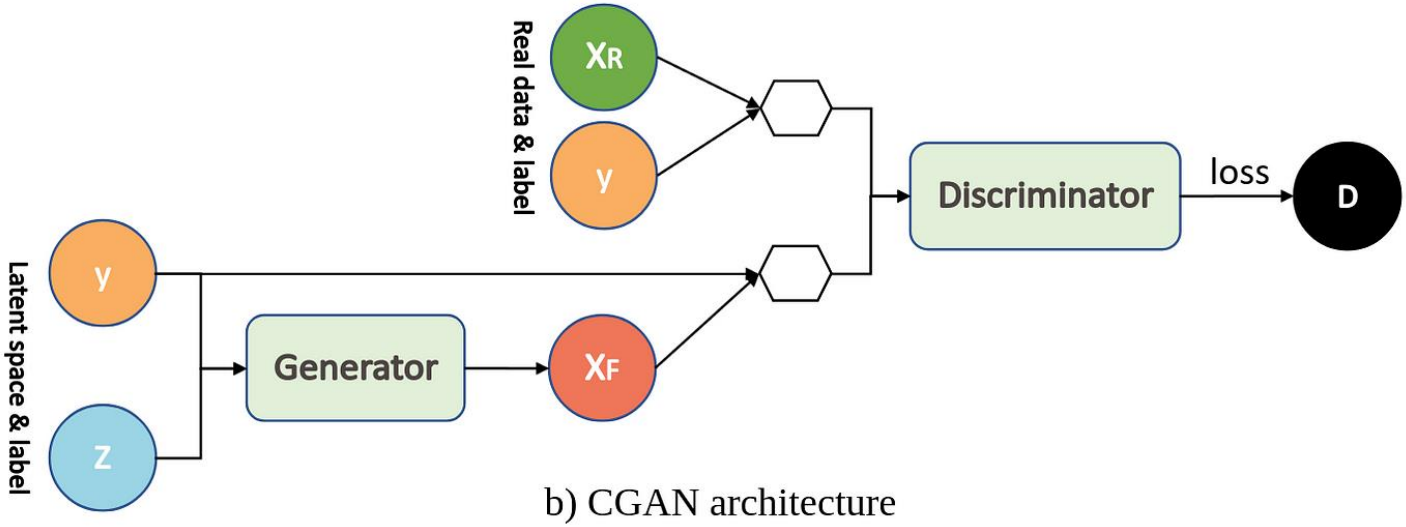
Generative Adversarial Networks: GAN



Generative Adversarial Networks: cGAN



a) GAN architecture



b) CGAN architecture

Generative Adversarial Networks: BCE loss

Binary Cross Entropy loss

$$\mathcal{L}_{\text{GAN}} = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

1. Unstable training
2. Overfitting
3. Gradients vanishing/Explosion
4. Mode collapse

Generative Adversarial Networks: Wass loss

Wasserstein distance loss + Spectral Normalization

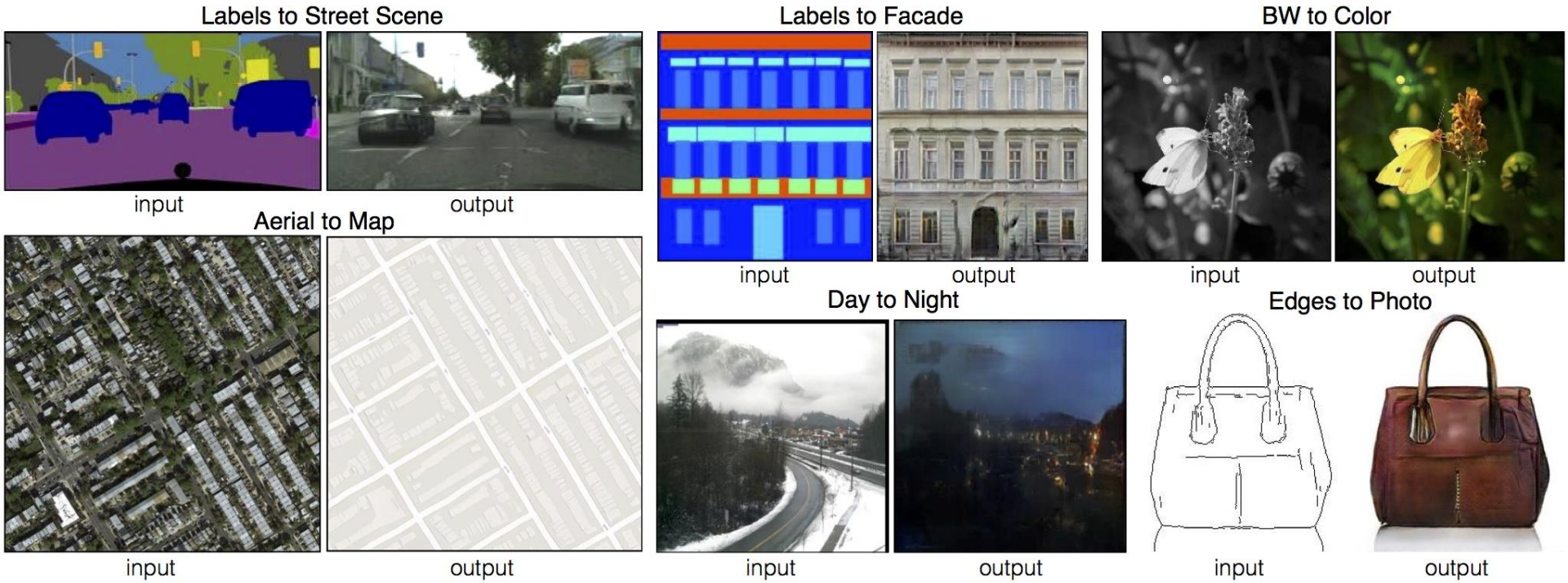
$$W(P,Q) = \inf_{\pi \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\pi}[\|x-y\|]$$

$$\mathcal{L}_{\text{WGAN}} = \max_{\theta_D} \left(\mathbb{E}_{x\sim p_{\text{data}}}[D(x)] - \mathbb{E}_{z\sim p_z}[D(G(z))]\right)$$

$$\sigma(W) = \frac{\|Wv\|_2}{\|v\|_2}$$

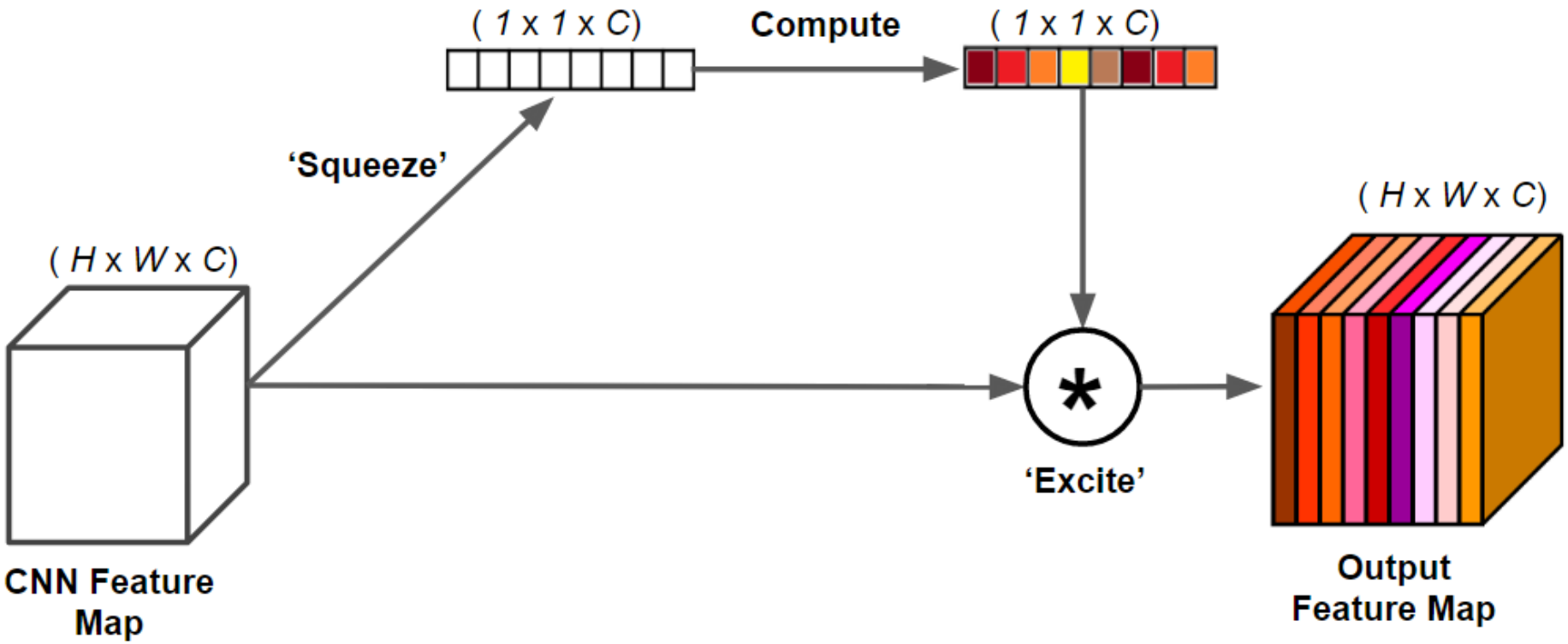
$$\text{Spectral Normalization}(W) = \frac{W}{\sigma(W)}$$

GAN Backbone: Pix2Pix



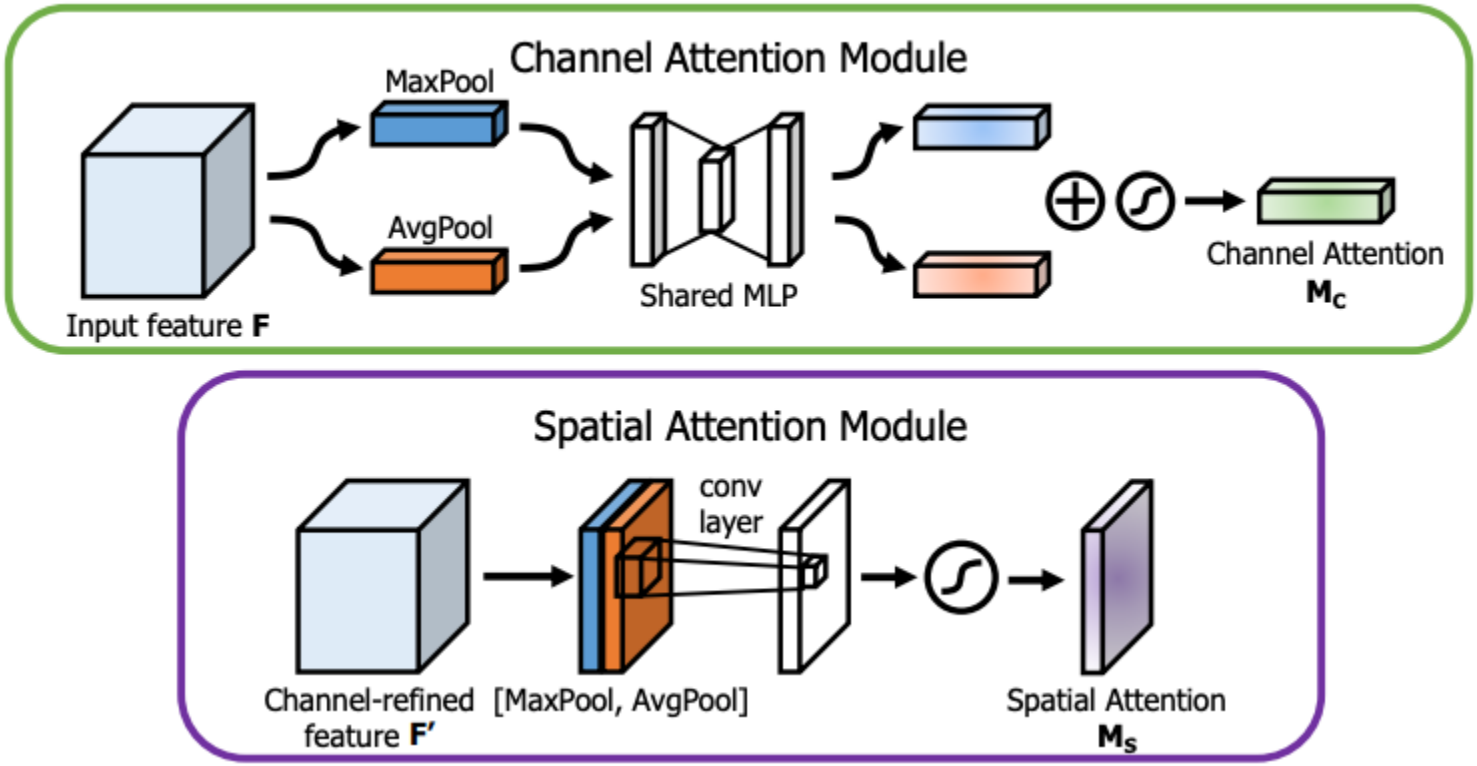
Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Attention mechanisms: SEN



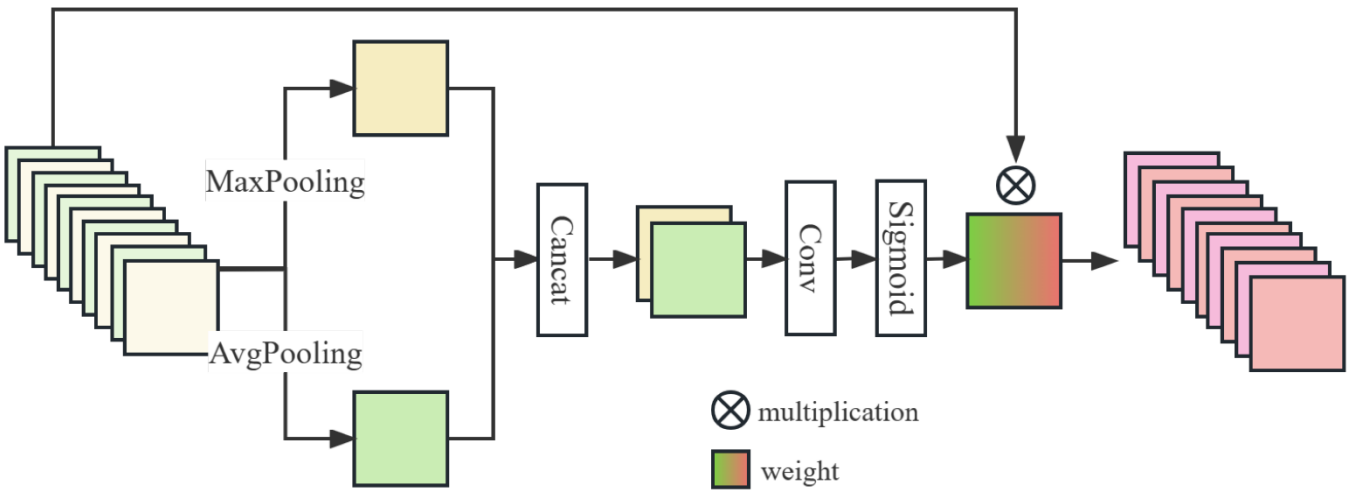
Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Attention mechanisms: BAM and CBAM

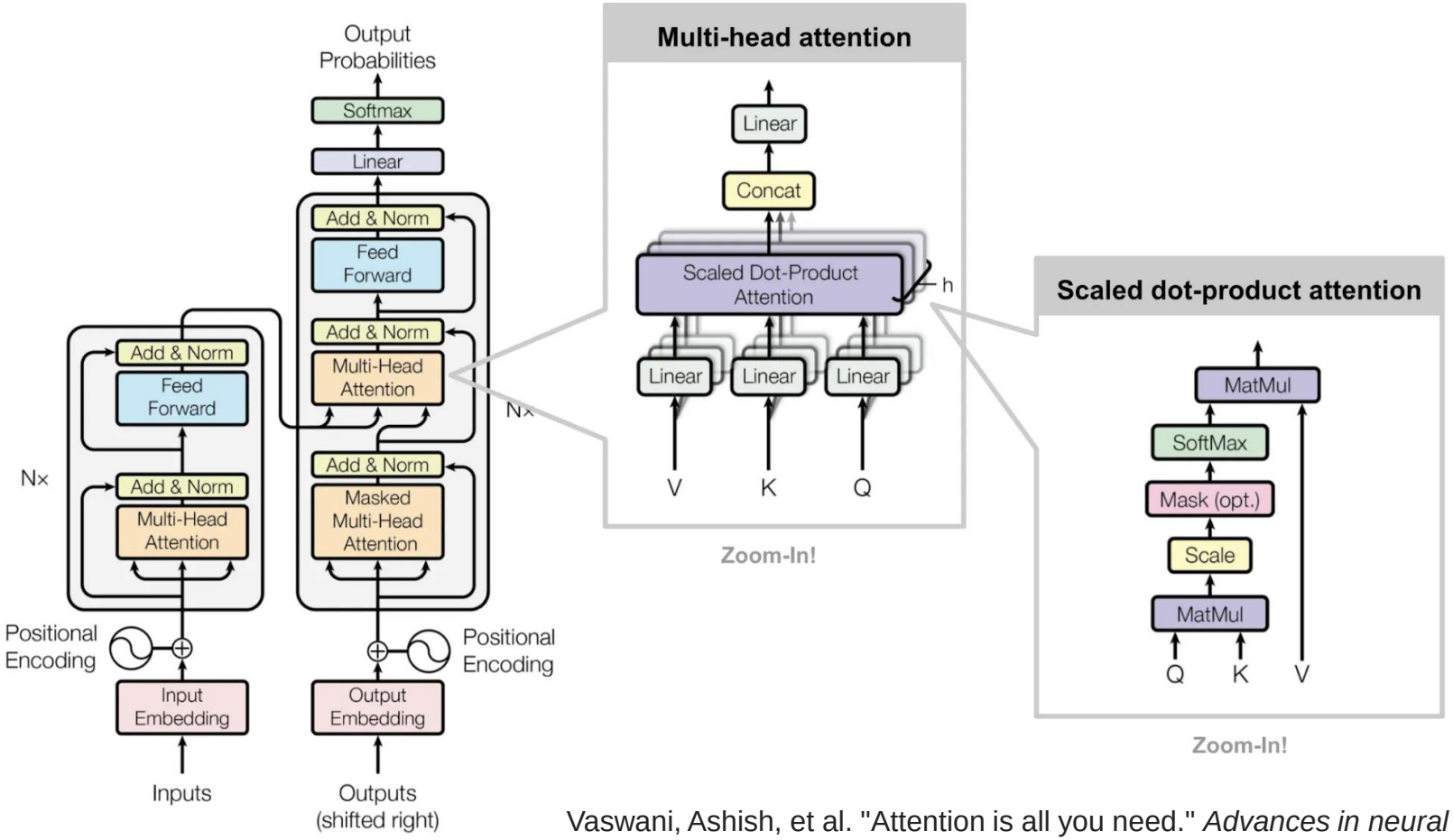


Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Attention mechanisms: Channel Attention ex.

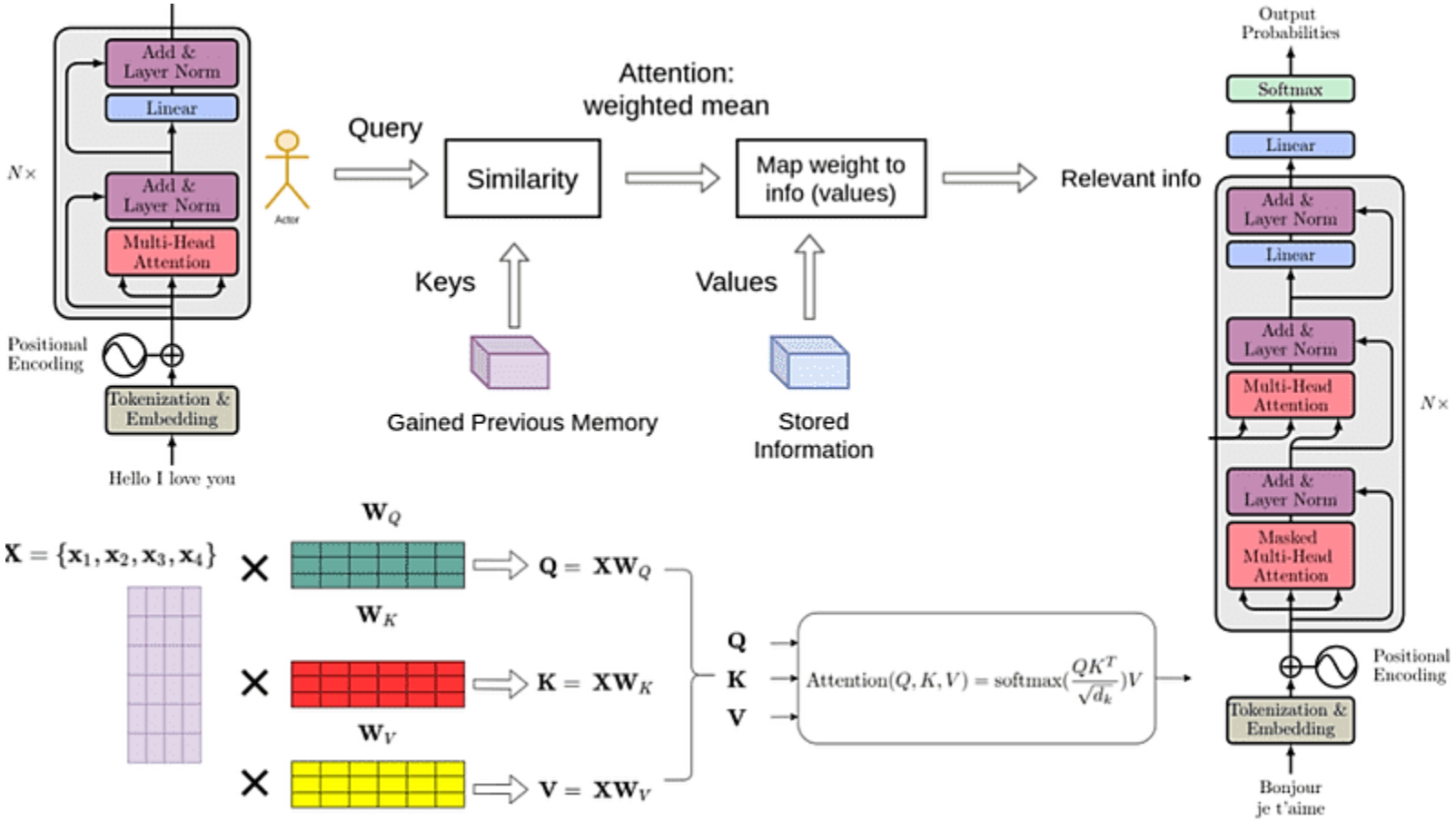


Attention mechanisms: Self-Attention

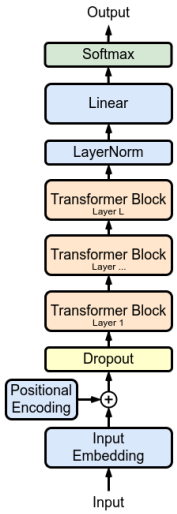


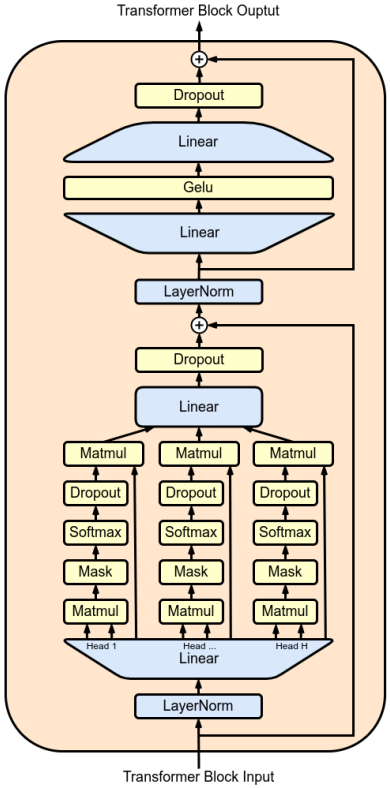
Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Transformers

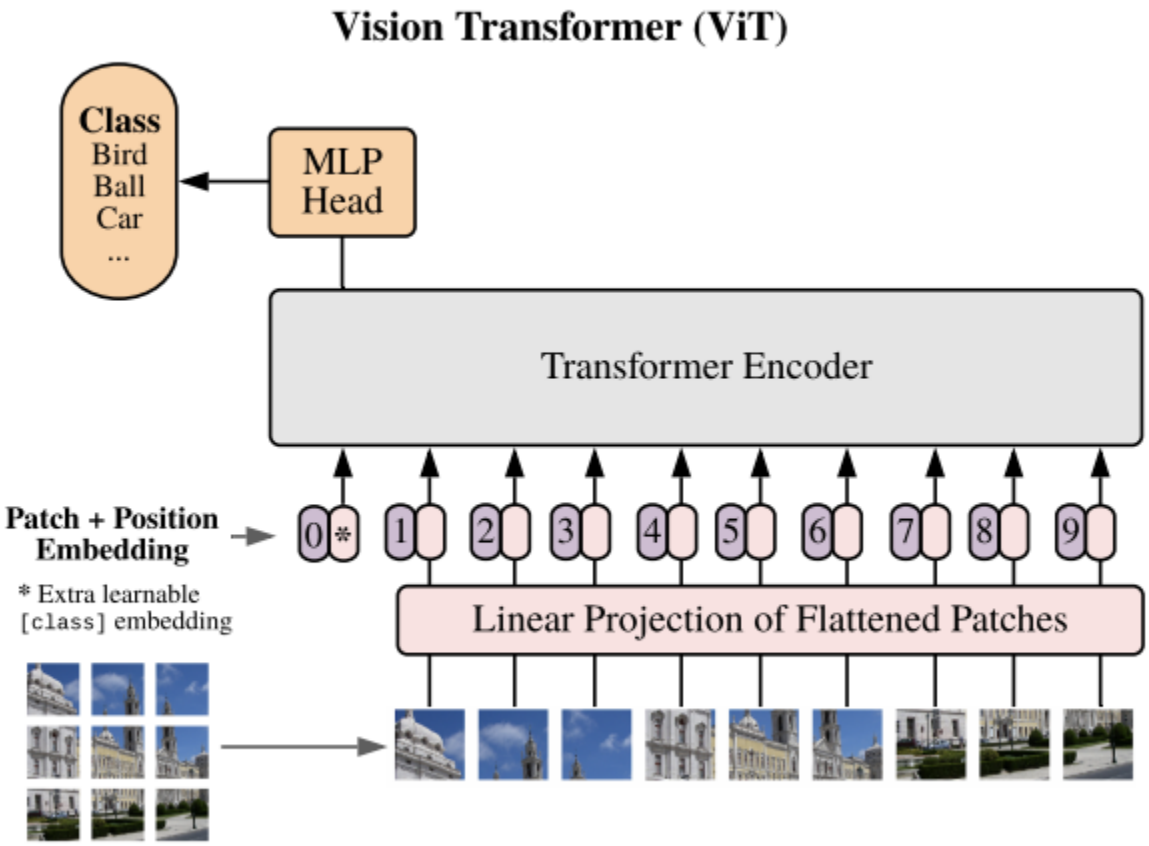


Transformers: GPTransformers

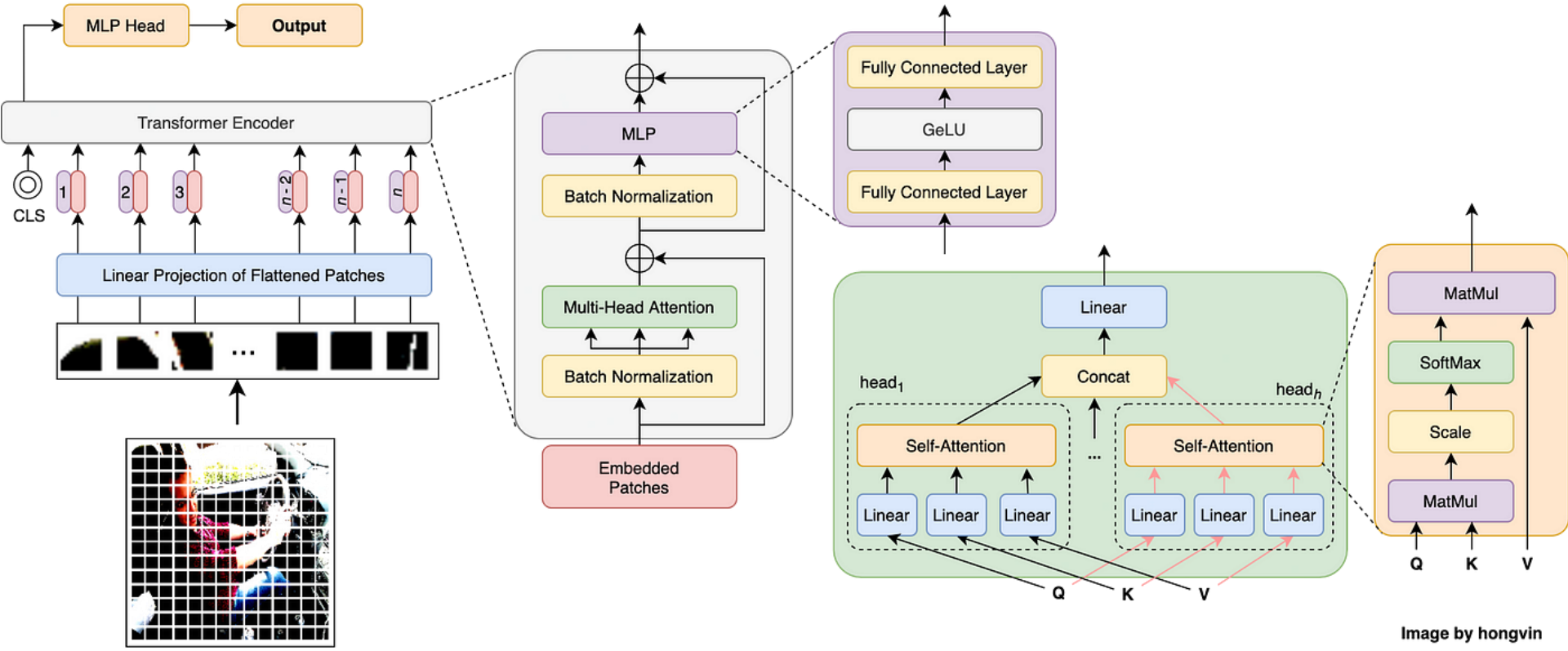




Transformers: : Vision transformer

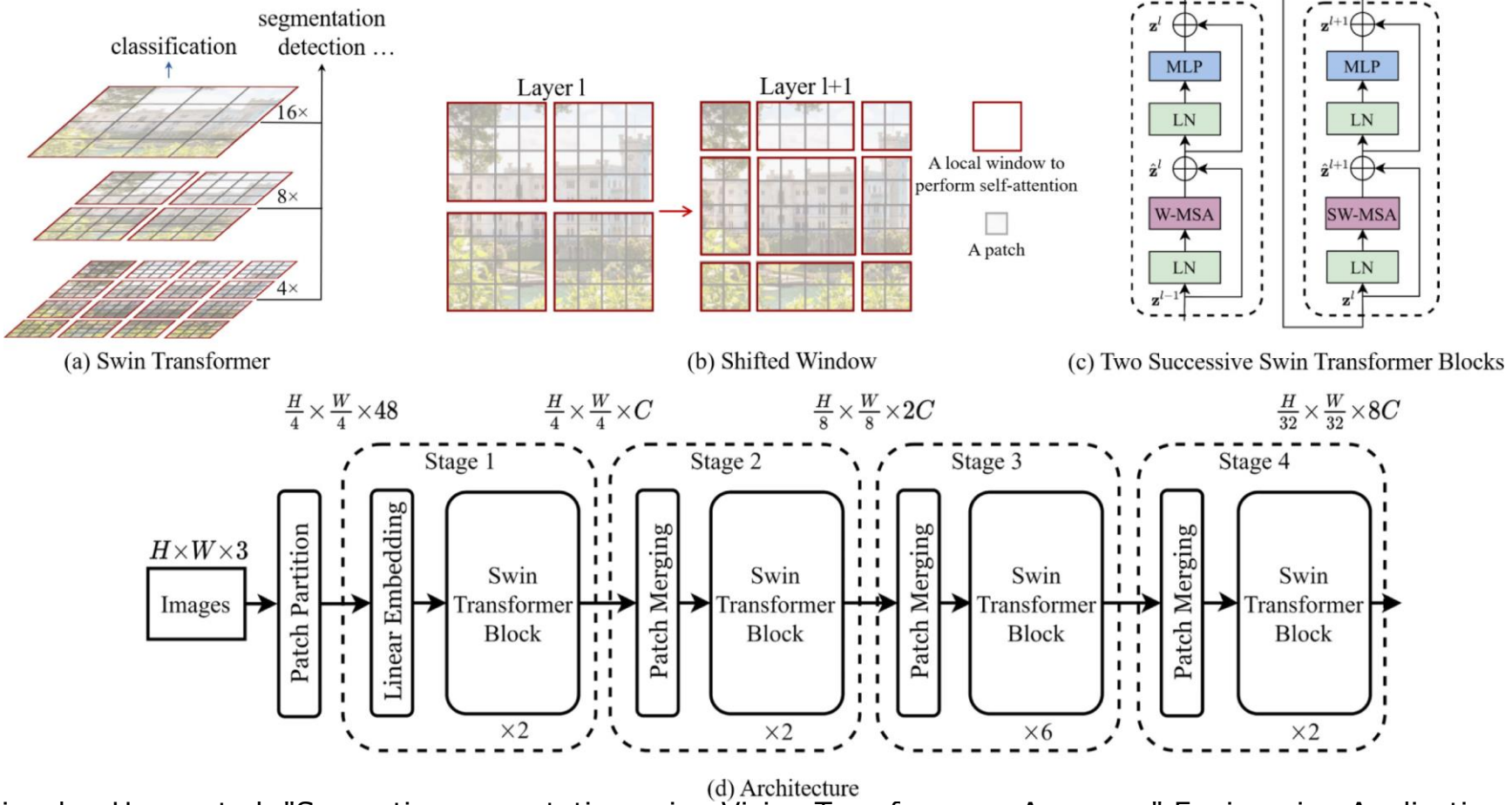


Transformers: : Vision transformer



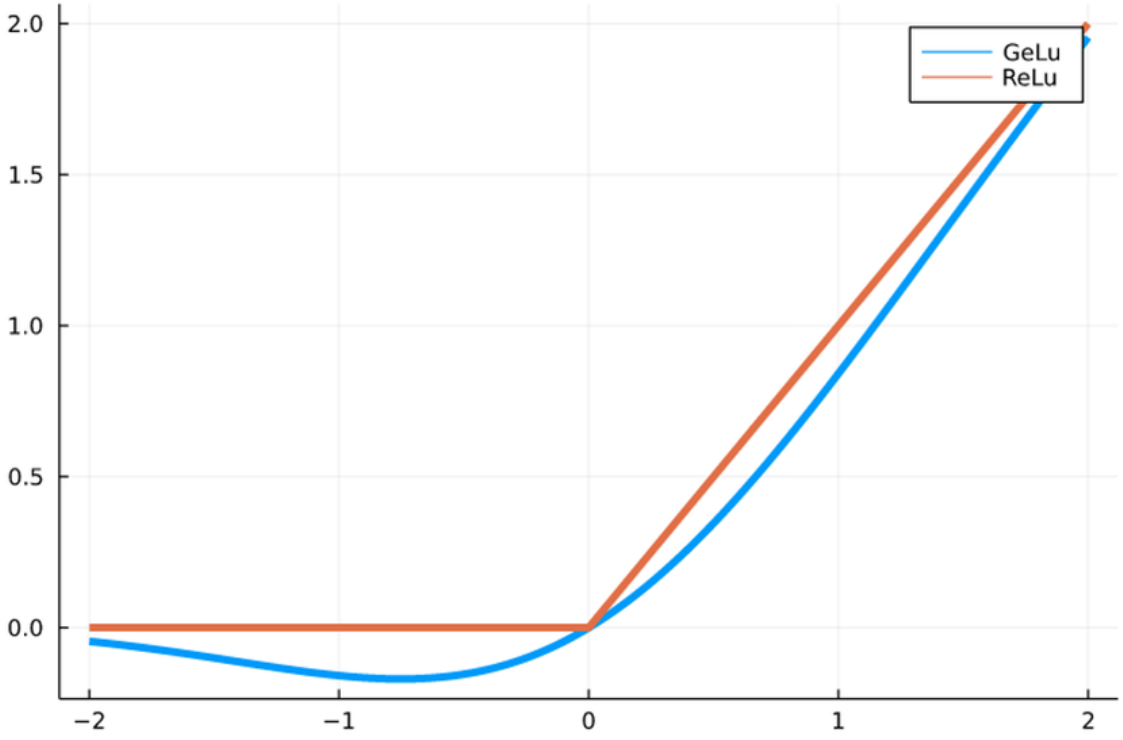
https://khvmaths.medium.com/vision-transformer-understanding-the-underlying-concept-83d699d71180

Transformers: : Swin transformer



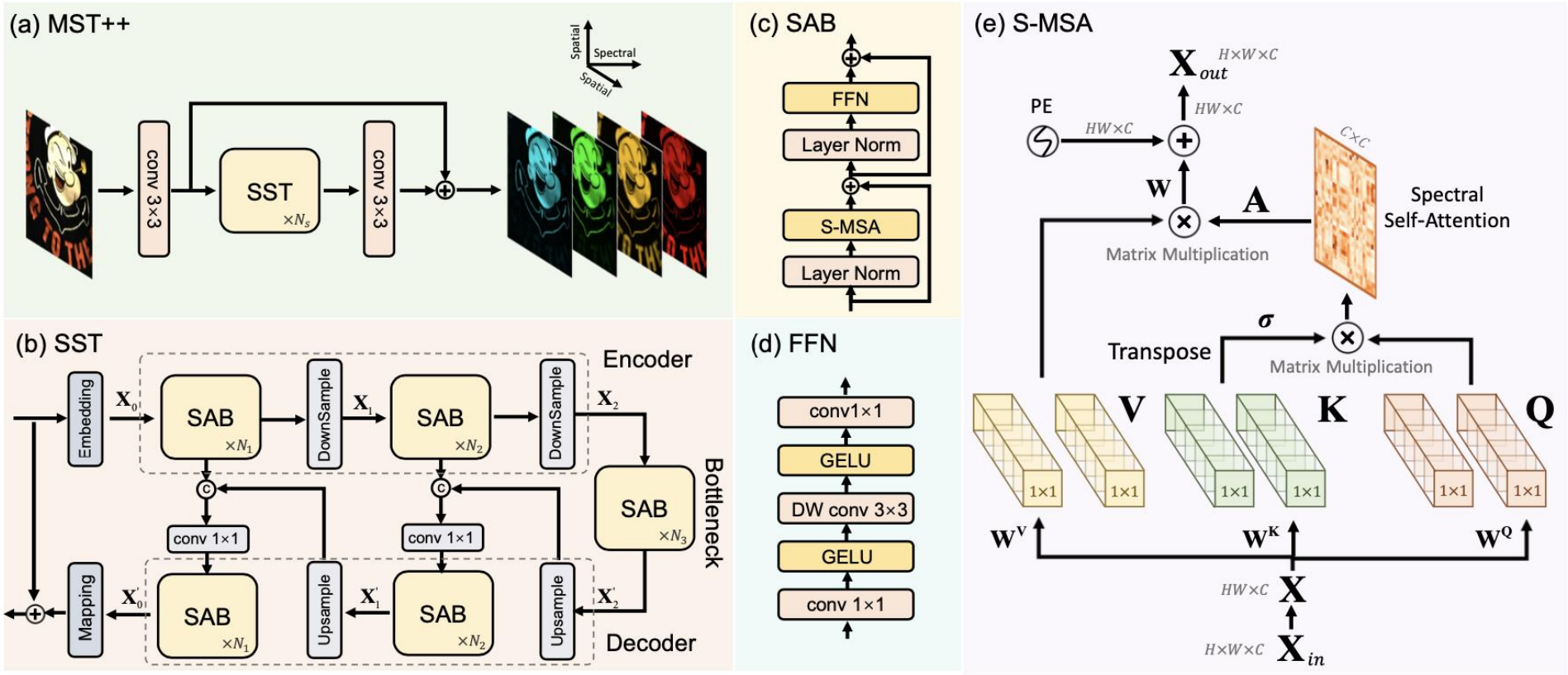
Thisanke, Hans, et al. "Semantic segmentation using Vision Transformers: A survey." Engineering Applications of Artificial Intelligence 126 (2023): 106669.

Transformers: : GELU activation function



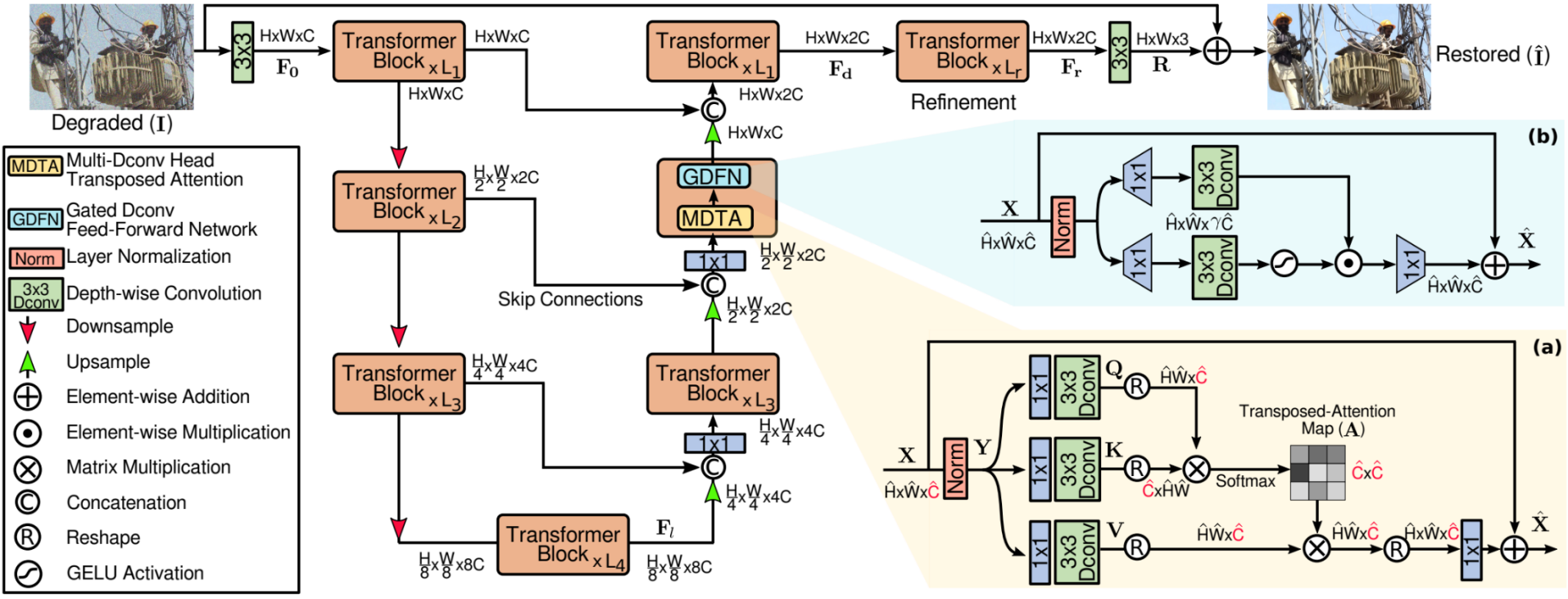
$$f(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right)$$

Backbone/SOTA model HSI: MST++



Cai, Yuanhao, et al. "Mst++: Multi-stage spectral-wise transformer for efficient spectral reconstruction." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Backbone/SOTA Transformer: Restormer



Zamir, Syed Waqas, et al. "Restormer: Efficient transformer for high-resolution image restoration." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

Hands on: GANs

- ✓ https://www.tensorflow.org/tutorials/generative/dcgan
- ✓ https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- ✓ https://github.com/tensorflow/gan/blob/master/tensorflow_gan/examples/colab_notebooks/tfgan_tutorial.ipynb