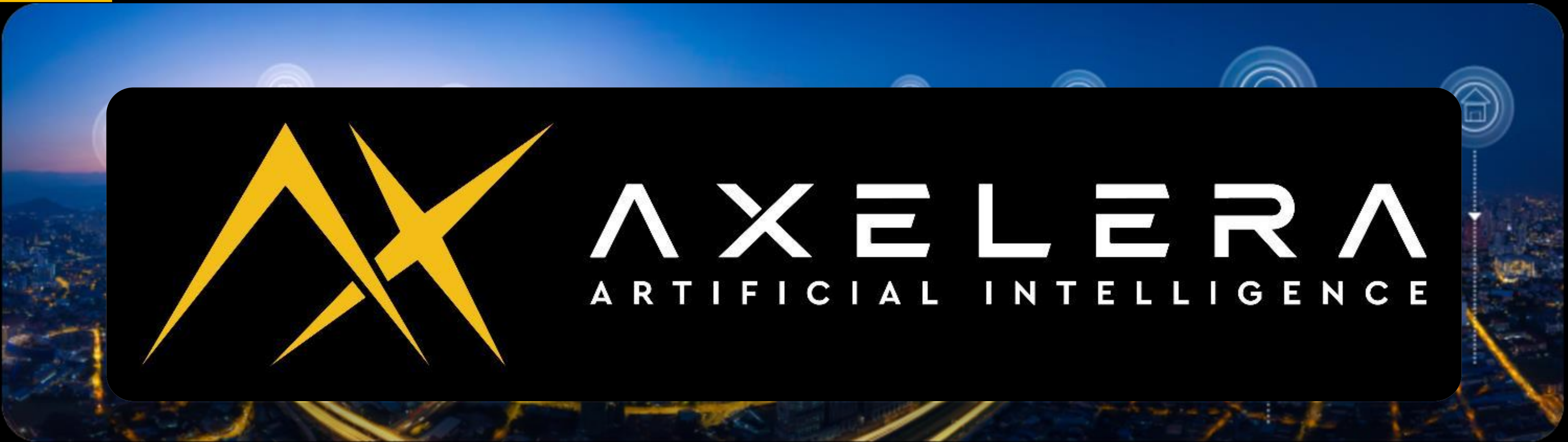
Insights and Trends in Machine Learning for Computer Vision

Bram Verhoef & Martino Dazzi



A powerful, efficient, competitive and user-friendly hardware and software AI platform to accelerate computing vision at the edge

The A-team

MANAGEMENT

Fabrizio Del Maffeo, MsC
CEO & Co-Founder
Axelera AI

- Former Head of AI of the Bitfury Group
- Former Vice-President & Managing Director of AAEON Europe (ASUS group)
- Founder of UP Bridge the Gap, reference platform of Intel IOTG & Movidius

Evangelos Eleftheriou, PhD
CTO & Co-Founder
Axelera AI

- Former Leader of the Neuromorphic Computing Activities at IBM Research
- IBM Fellow (highest honor within IBM)
- 160 patents; 200+ publications; 15,000+ citations

ADVISORS

Prof. Marian Verhelst, PhD – MICAS/KU Leuven
Advisor

Prof. Luca Benini, PhD – ETH Zurich & UniBo
Advisor

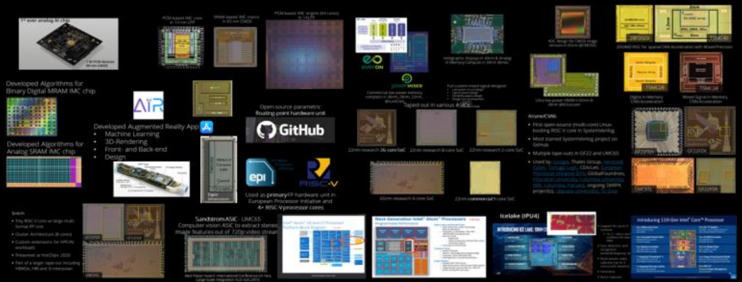
Prof. Torsten Hoefler, PhD – ETH Zurich
Advisor

THE A-TEAM

- **Founded in July 2021 as spin-off of Bitfury AI and IMEC**
- **53 people working remotely and from the offices in Eindhoven (NL), Leuven (BE), Zurich (CH)**
- **>20 people hold a PhD with more than 30.000 citations on AI scientific papers**
- **> 100 people onboard planned by end of 2022**

ASUS intel Google Qualcomm BITFURY IBM imec ETH zürich

TEAM ACHIEVEMENTS



AXELERA ARTIFICIAL INTELLIGENCE

AI is expanding from cloud to edge

Limited Power Budget

Limited computational power

Limited connectivity

Limited computational power scalability

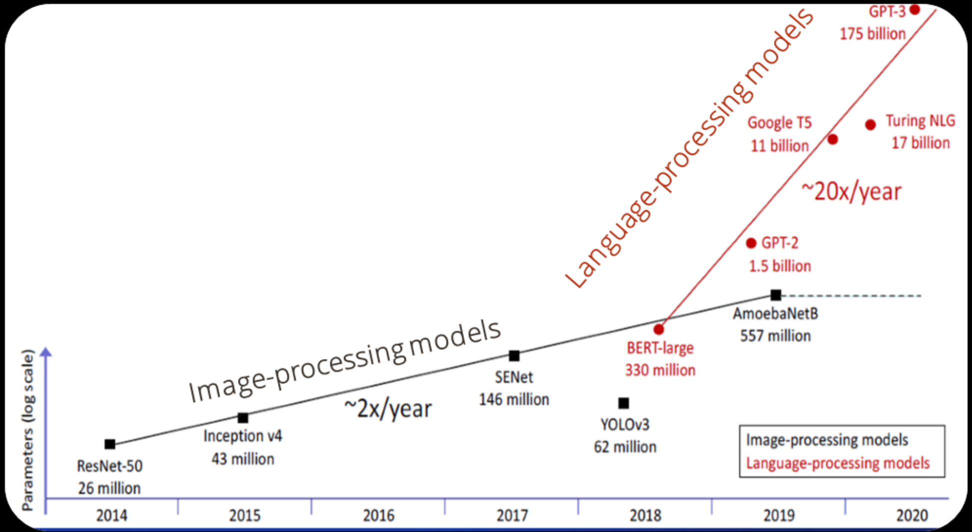
High Cost/Performance ratio

Field/Environmental constraints

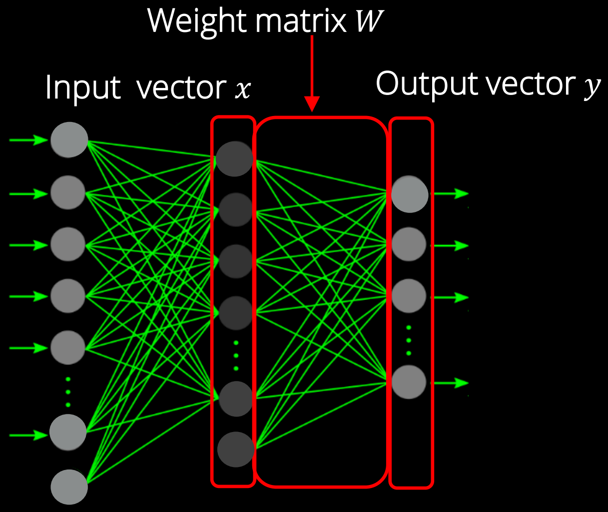
Neural network size is growing exponentially

Compute requirements
to train large neural networks are
doubling every 3.5 months

Unsustainable without
significant hardware and software
innovation

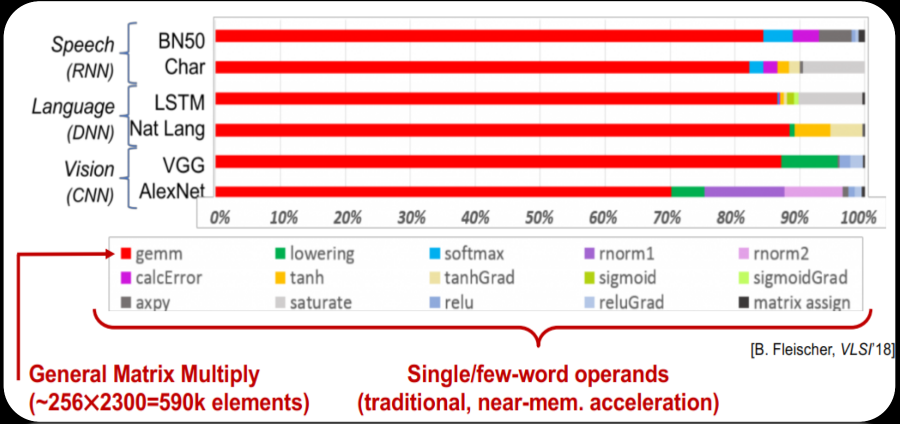


Neural networks are dominated by matrix multiply



$$y = W x$$

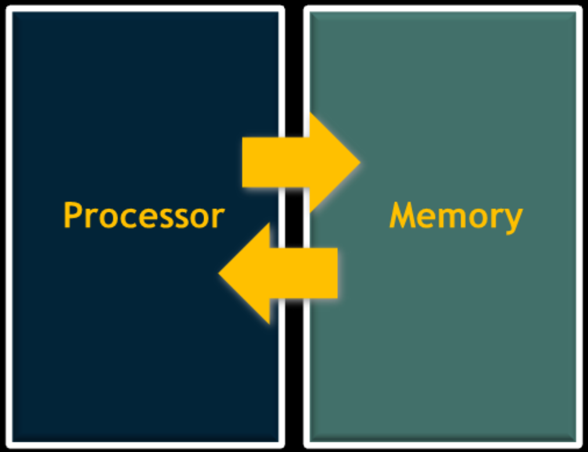
Matrix-vector multiplications constitute 70-90% of the total deep learning operations



Standard computing is not optimized to process billions of multiplications and accumulations

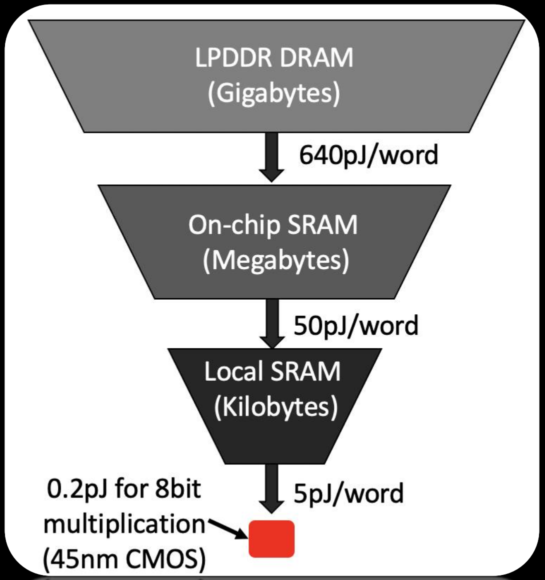
Power consumption is dominated by data movement

Conventional von Neumann computing architecture



Horowitz, *ISSCC*, 2014

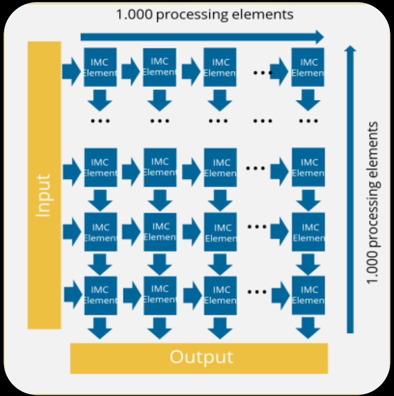
Cost of data transfer and computation



Dally, *ScaledML*, 2019

Our game-changing AI acceleration technology

In-Memory Computing

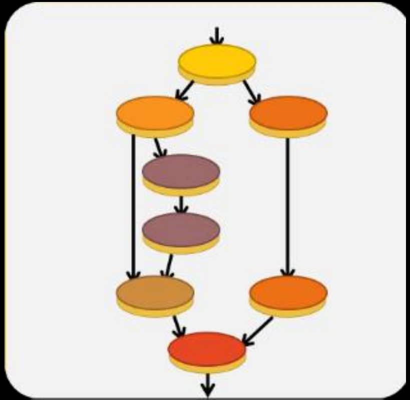


Hundred of Tera Operations per second

+

Highly programmable

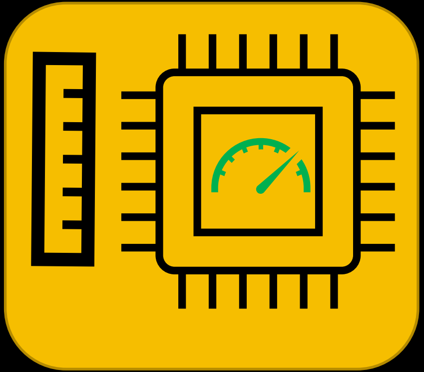
Dataflow



Flexibility to support different Neural Networks

A powerful and green technology

High performance/area



6-20x better than Nvidia

Efficient

5-20 x better than Nvidia

Scalable

From single core to multicore
From 12nm to 5nm or smaller
From edge to end node or cloud

We accelerate AI vision at the edge

The same hardware and the same neural networks can solve multiple problems





The same network fed with different data recognizes different things

We are creating an AI platform to cover the AI value chain

Selected ODM will buy the chip and enable the hardware supply chain

AI Final User

AI Software

PYTORCH

TensorFlow

AI HW System

Click to add text

AI HW Cards

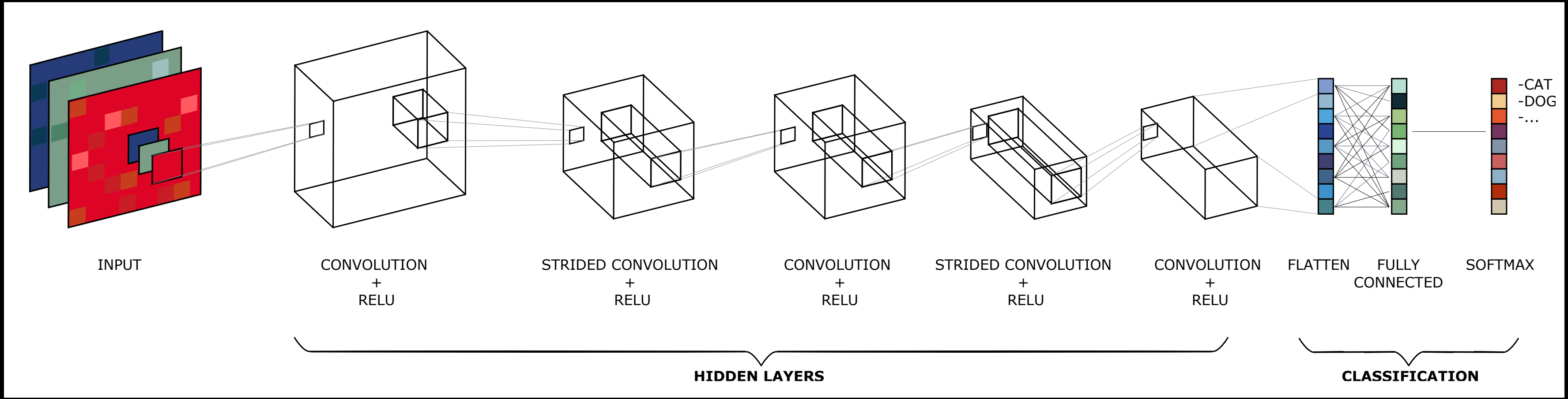
AI Chip

They test, select the technology and they buy or suggest to customer the system to buy

We talk to AI SW companies to support their software and neural networks needs
We work with ODM computing company to deliver AI hardware solutions for field deployment

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs)

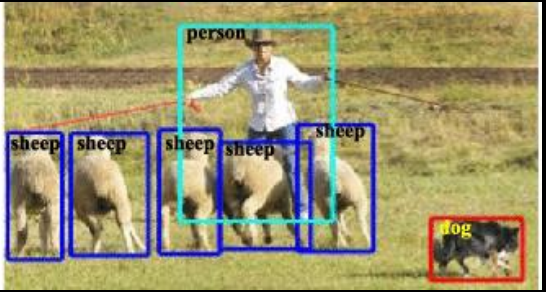


- Neural networks designed for computer vision applications.
- Based on the inductive bias that important information is local => use of convolutions.

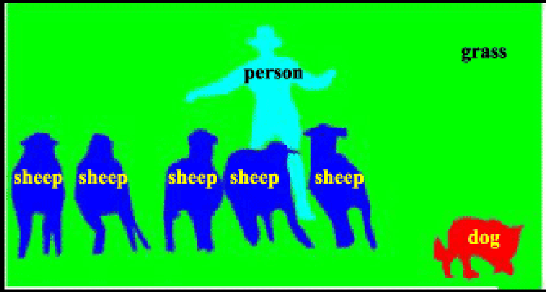
CNNs used at the edge

Object classification

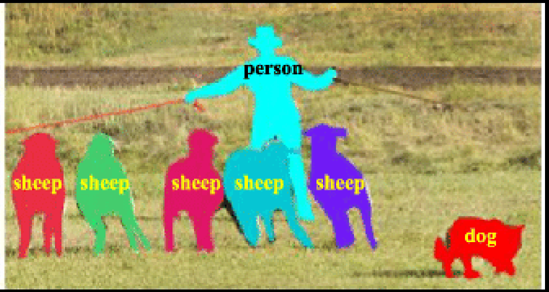
Object detection



Semantic segmentation

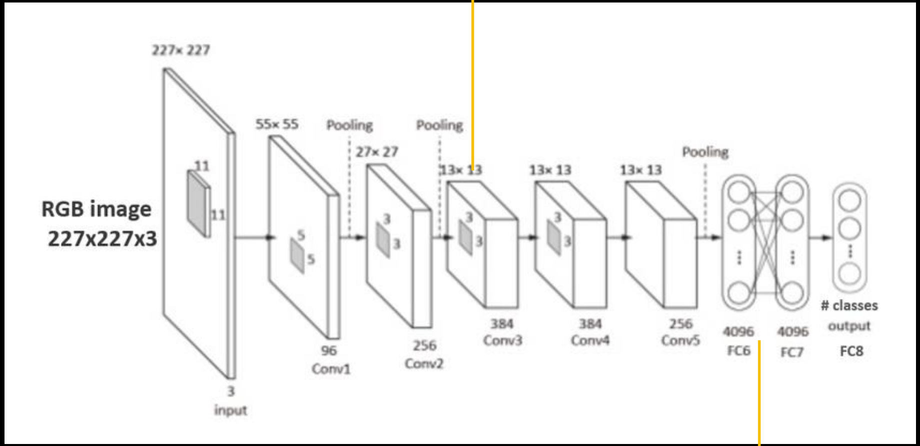


Instance segmentation



Early CNNs

Dense filters



727M FLOPs

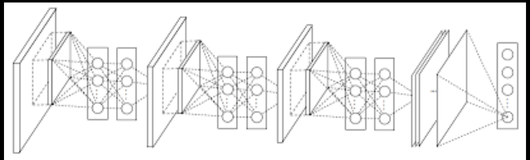
62M parameters

63% top-1 at IM-1K

Large FC layers: ~59M parameters

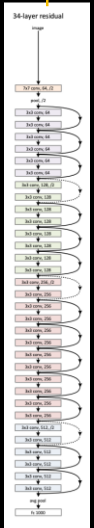
Shallow structure

Early CNNs became efficient and accurate



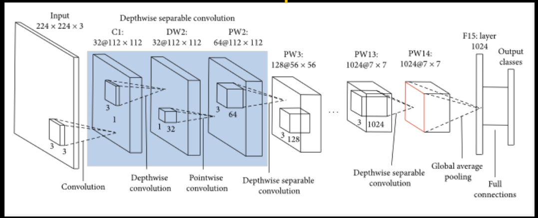
NIN: Global average pooling

4G FLOPs
25M parameters
76.1% top-1 at IM-1K



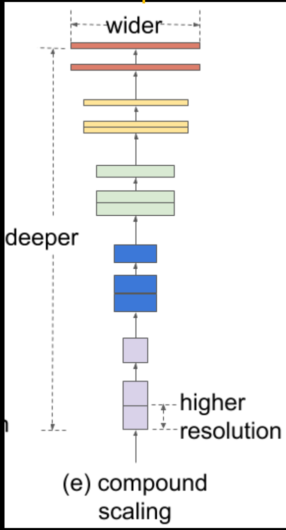
ResNet: Residual connections

500M FLOPs
5.3M parameters
74.4% top-1 at IM-1K



MobileNet: Depthwise-separable convolutions

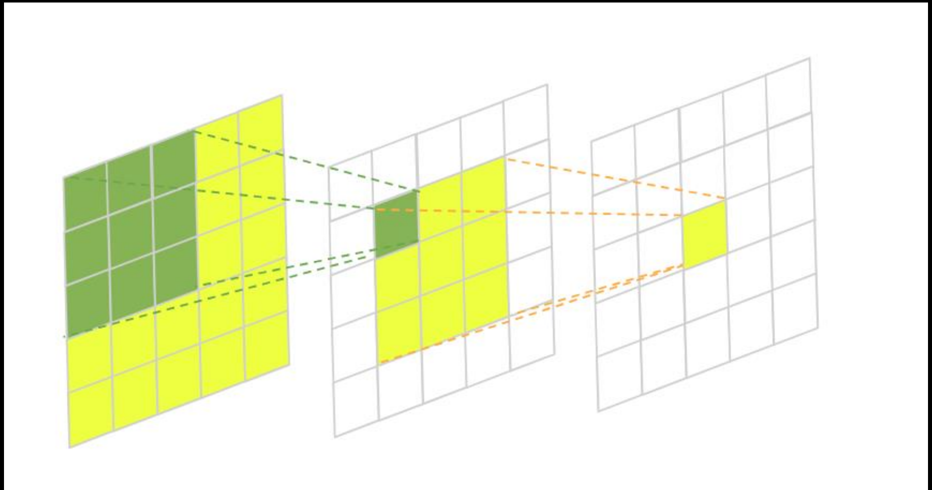
390M FLOPs
5.3M parameters
77% top-1 at IM-1K



EfficientNet: Compound scaling of Depth, width and resolution

[AutoSlim: Towards One-Shot Architecture Search for Channel Numbers , 2019]

Problems with CNNs



- Need many layers in order to gather global information about the image
- Weight sharing induces biases, which may decrease model capacity and applicabilityx

Vision Transformers (ViTs)

What are Transformers?

Initially proposed to replace RNNs
In NLP

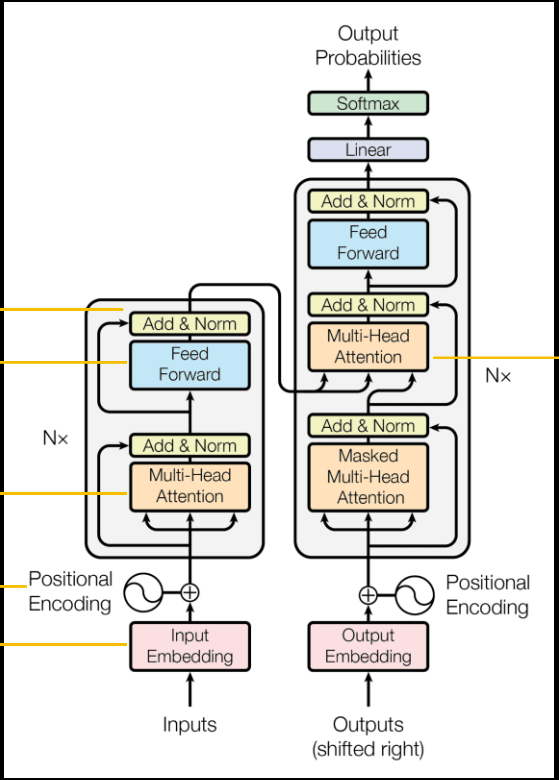
Residual connection + layer normalization

Pointwise MLP

Self-attention

Positional encoding

Input vectors/tokens



Encoder Decoder

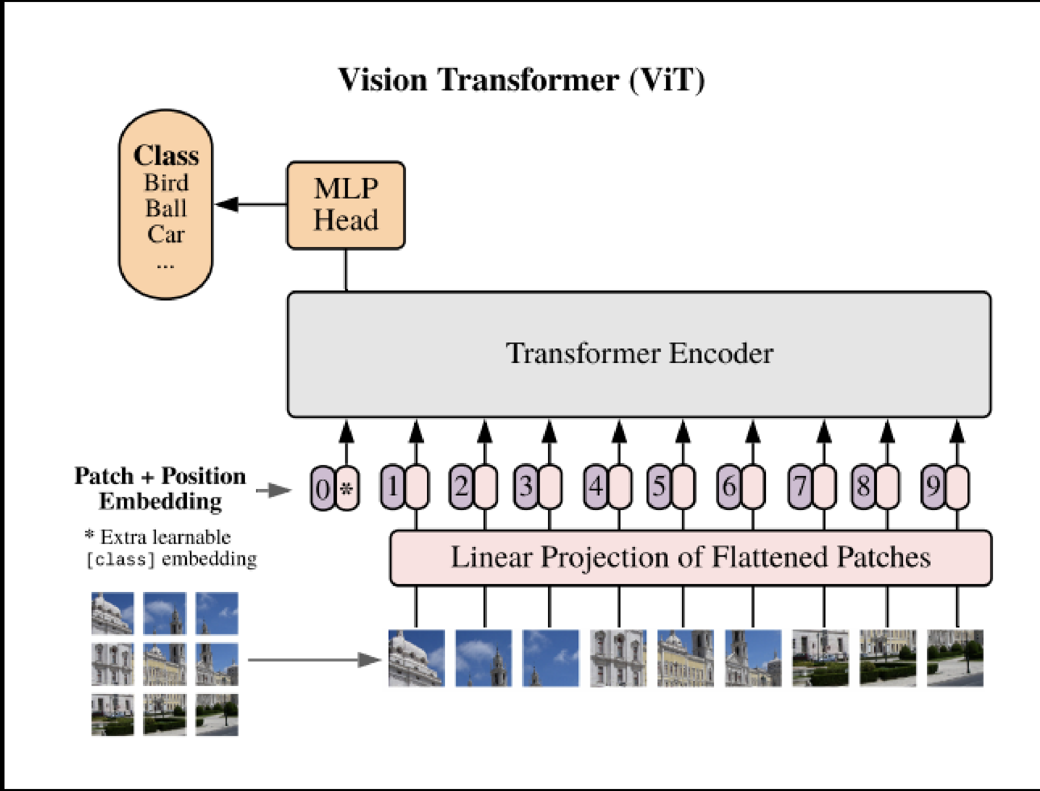
Cross-attention

Vision transformer (ViT)

ViT-/16B:
17.6G FLOPs
86M parameters
77.9% top-1 at IM-1K

87.76% when ViT-L/16 (307M params) is pre-trained on JFT-300M

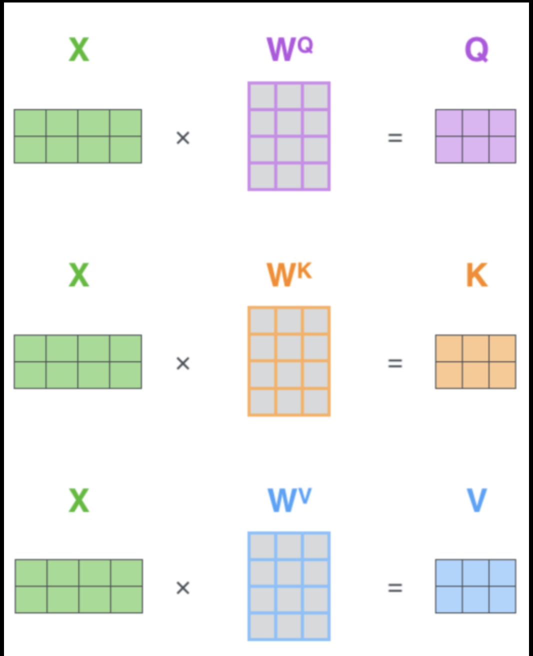
Vision transformer (ViT)

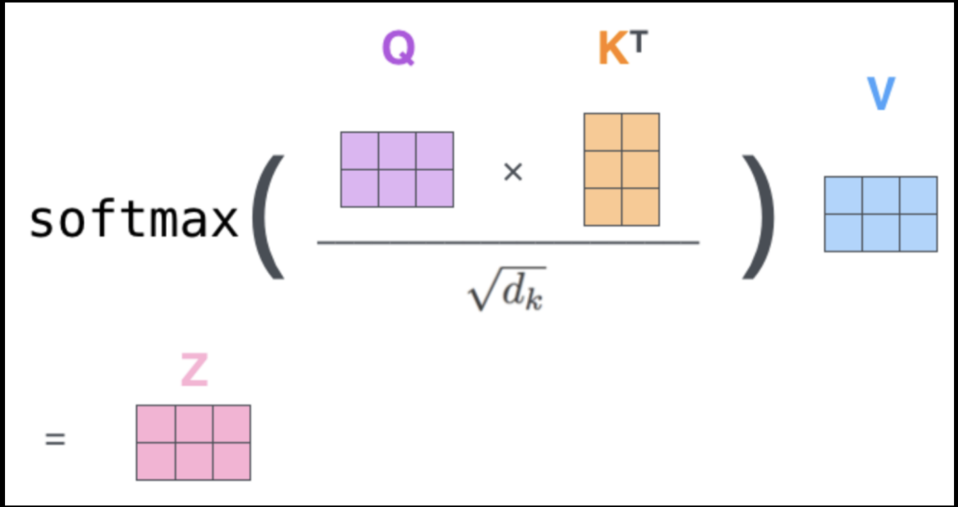


ViT-/16B:
17.6G FLOPs
86M parameters
77.9% top-1 at IM-1K

87.76% when ViT-L/16 (307M params) is pre-trained on JFT-300M

What is self-attention?





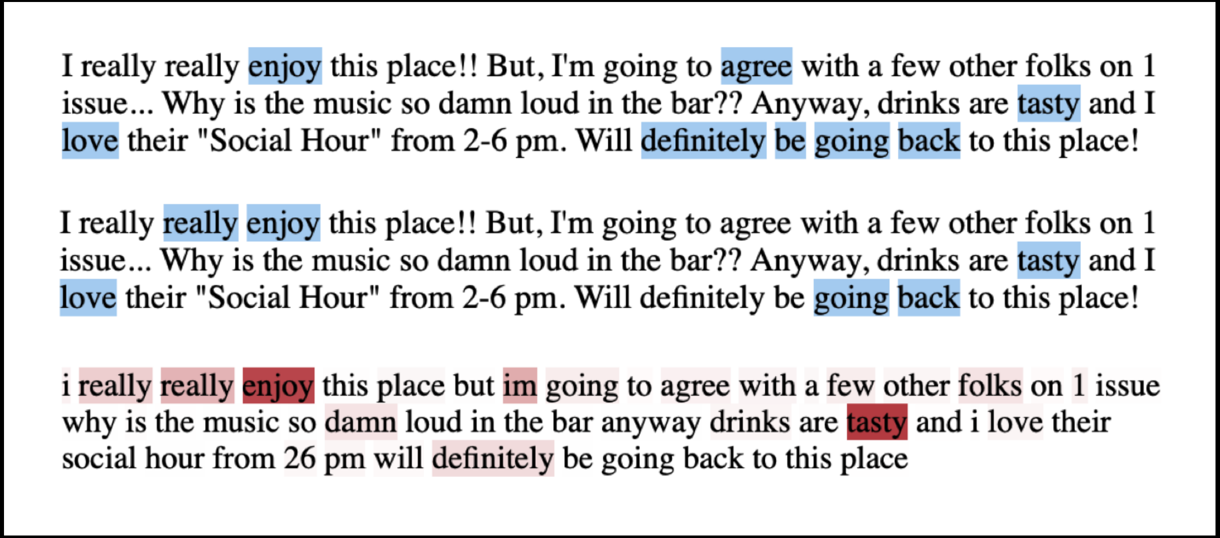
Every token is compared to all other tokens to compute attention map → Quadratic complexity

What is attention?

- Self-Attention represents the relative importance of each token with respect to any other token.
- In ViT -> First token represents the class => Attention represents the relative importance of every tokens with respect to the target task.

[Rao et al., *"DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification"*]

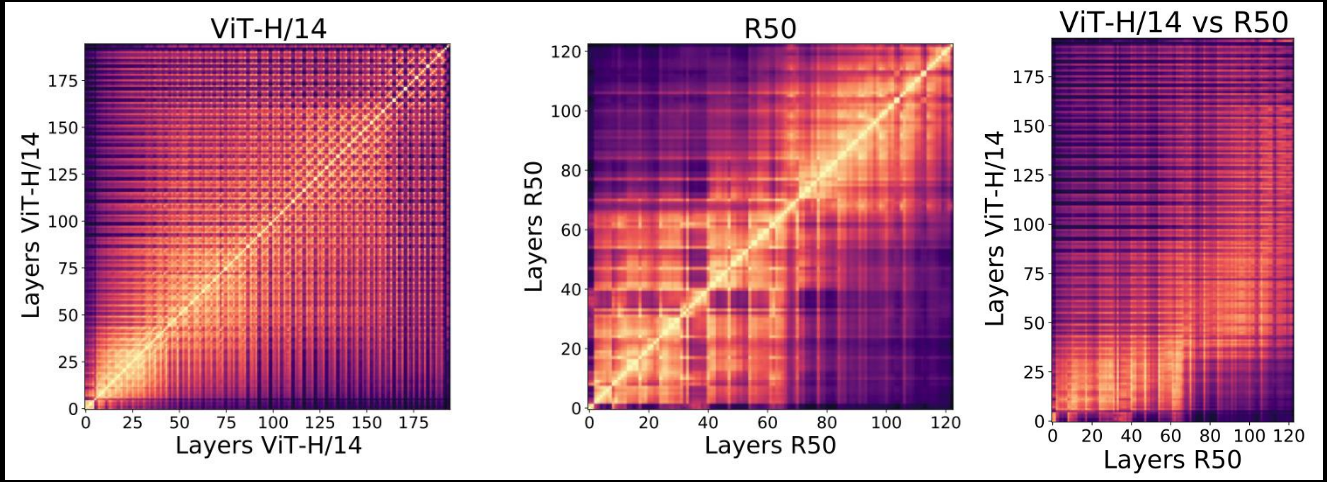
What is attention?



- As the name suggests, it is a measure of which elements the network should pay attention to for a specific task.
- Very similar to how humans pay attention: in the example, humans (blue) and transformer (red).

[Sen et al., *"Human Attention Maps for Text Classification: Do Humans and Neural Networks Focus on the Same Words?"*]

Attention vs Convolution



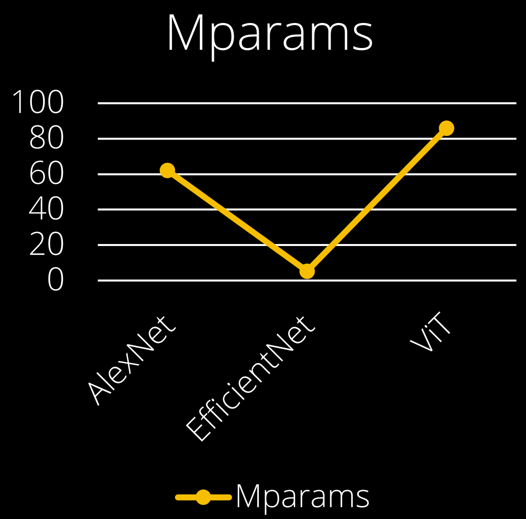
- CKS similarity between activations: In ViT, activations look similar throughout, in ResNet50, difference between first and last set of layers.
- Cross-similarity between networks: First half of RN-50 activations similar ro first quarter of ViT activations, no similarity with the last layers of ViT.

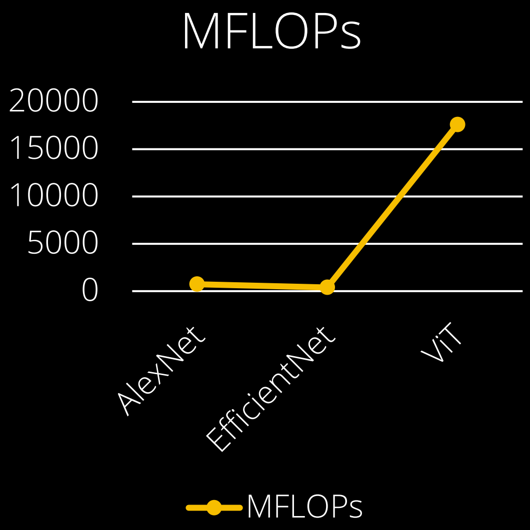
[Raghu et al., *"Do Vision Transformers See Like Convolutional Neural Networks?"*]

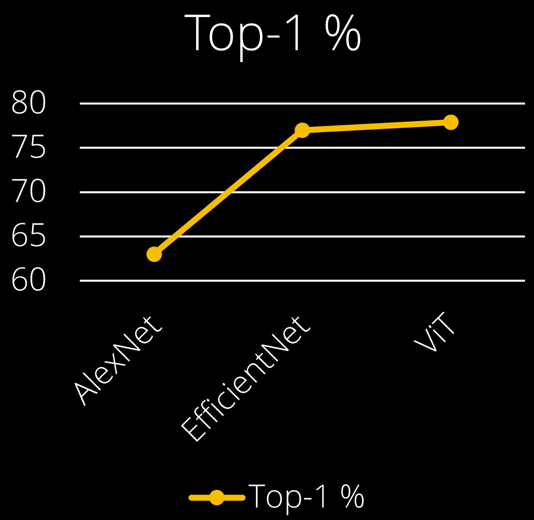
CNNs vs Transformers

- CNNs build on an inductive bias that suggests local information is important; ViTs process the whole information in one go.
- Lack of inductive bias => Model has to learn it itself => more parameters, more data augmentation.
- Learning paradigm of attention-based models works also for computer vision => surpasses CNNs when trained on very large datasets (e.g. JTF-300M) and fine-tuned on smaller tasks.

Alexnet -> EfficientNet -> ViT







Problems with ViT

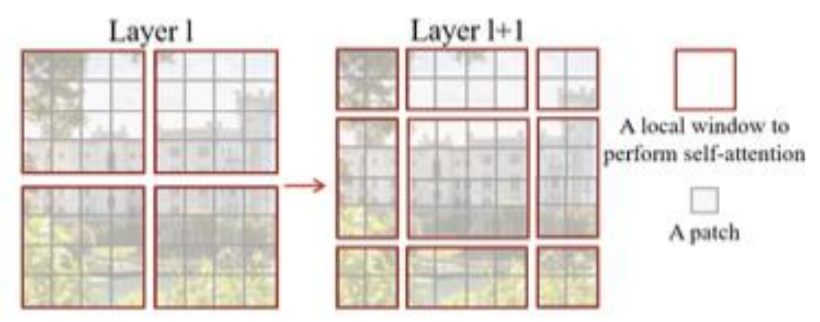
- Quadratically complex self-attention
- Difficult to train
- Low accuracy compared to CNNs when trained on “small” datasets

Problems with ViT: Complex Self Attention

- Problems:
  - Increases # computations
  - Increases memory requirements
  - Limits the patch size/resolution of image processing
  - Little inductive bias -> Models are difficult to train

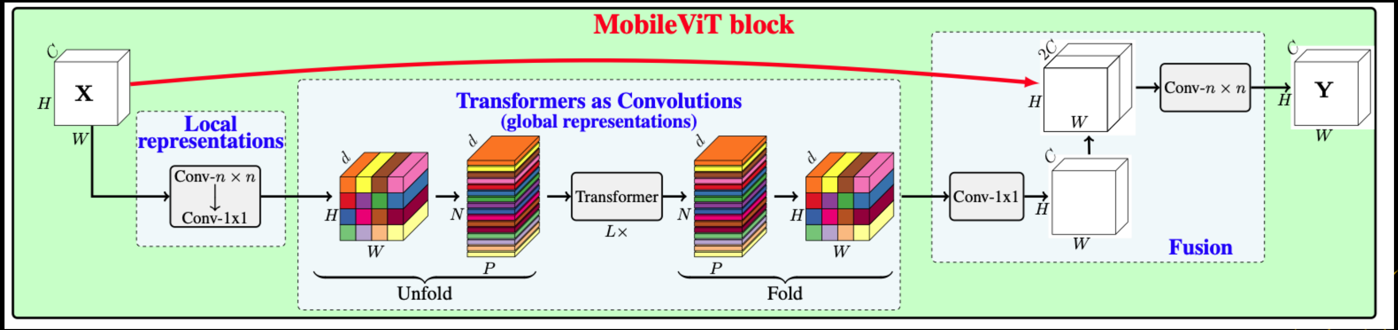
Problems with ViT: Complex Self Attention

- Problems:
  - Increases # computations
  - Increases memory requirements
  - Limits the patch size/resolution of image processing
  - Little inductive bias -> Models are difficult to train
- Solutions:
  - Local self-attention within windows of patches, hierarchical structure (e.g. Swin Transformer)



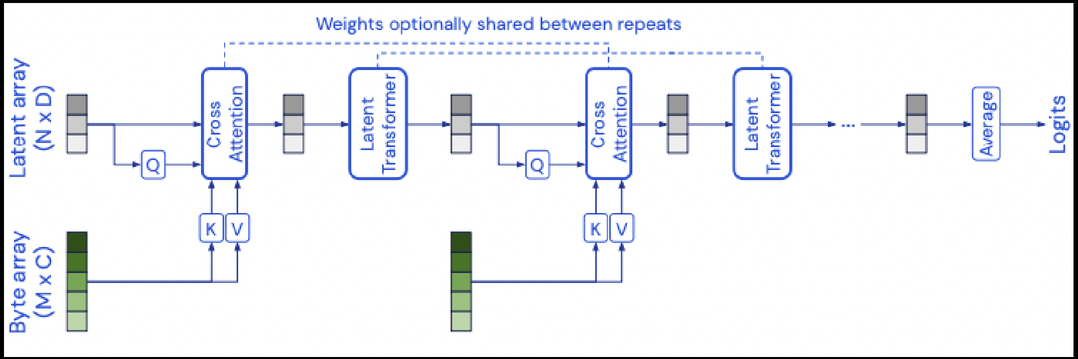
Problems with ViT: Complex Self Attention

- Problems:
  - Increases # computations
  - Increases memory requirements
  - Limits the patch size/resolution of image processing
  - Little inductive bias -> Models are difficult to train
- Solutions:
  - Wrap attention between convolutions (e.g. MobileViT)



Problems with ViT: Complex Self Attention

- Problems:
  - Increases # computations
  - Increases memory requirements
  - Limits the patch size/resolution of image processing
  - Little inductive bias -> Models are difficult to train
- Solutions:
  - Replace/Modify classical self-attention with alternatives (e.g. Perceiver, AFT, VOLO, V-MLP)



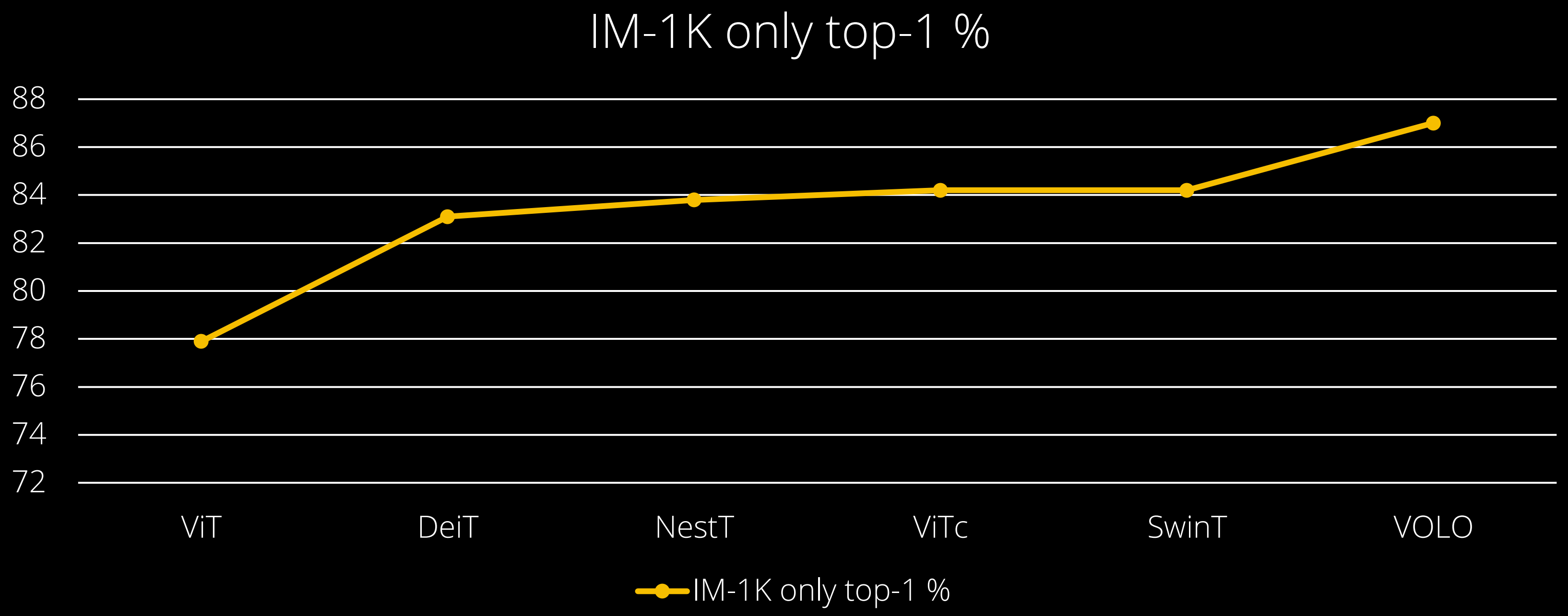
Problems with ViT: Complex Self Attention

- Problems:
  - Increases # computations
  - Increases memory requirements
  - Limits the patch size/resolution of image processing
  - Little inductive bias -> Models are difficult to train
- Solutions:
  - Local self-attention within windows of patches (e.g. Swin Transformer)
  - Hierarchical/pyramidal network architecture: Later layers have smaller feature maps (e.g. Swin Transformer, NestT)
  - Adding CNNs to pipeline to reduce the feature map size (e.g. MobileViT)
  - Replace classical self-attention with alternatives (e.g. Perceiver, AFT, V-MLP)

Problems with ViT: Difficult to train

- Solutions:
  - Better data-augmentation (DeiT, LV-ViT)
  - Replace early transformer blocks with CNNs (ViTc)
  - Pick correct optimizer, hyper-parameters, training set, training schedule, depth, etc.

Problems with ViT: Low accuracy on “small” datasets



ViT Evolution and Comparison to CNNs

	Initial ViT	Recent ViTs	CNNs
#Parameters	>80M	5-300M	1-300M
#FLOPs	>18G	1-300G	1-200G
Accuracy on small datasets	Worse than CNNs	On-par with CNNs	Best in class
Accuracy on large datasets	Better than CNNs	Better than CNNs	Worse than ViT (but ConvNeXT)
Attention Mechanisms	Multi-Head Self-Attention	Multi-Head Self-attention or alternatives	No attention, Channel attention, (self attention)
Attention Field	Global	Global and Local	None or local
Network depth	Shallow	Deeper	Deep
Layer types	Isotropic	Isotropic and Hybrid	Isotropic and Hybrid

ViT Summary

- ViTs started as Isotropic, shallow, low-inductive bias networks
- Over time, ViT's became hybrid, deeper and added inductive biases typically found in CNNs (local processing, hierarchical architecture)
- This evolution made ViTs more efficient, easier to train and more accurate

ViT Conclusions

- ViTs are typically more accurate when pre-trained on large datasets (IM-22K, JFT-300M), but little research here with CNNs and recent ConvNext comes close
- ViTs can have less inductive biases so more applicable to general (multi-modal) data
- ViTs are easy to scale, so easier to adapt to large (multi-modal) datasets
- CNNs are typically more accurate when trained on small datasets
- CNNs are typically more computationally efficient for CV due to their inductive biases
- CNNs often rely on easier dataflows: BN vs. LN, softmax in final layer vs. after SE, ReLU vs. GELU, conv vs. SE, etc.

ViT Conclusions

- With proper modifications ViTs can be used on the edge
- For typical edge CV, with few resources, there's currently no evidence that ViTs are a better alternative to CNNs. <u>CNNs are usually more efficient, easier to implements and more accurate</u>
- BUT progress is fast and some ViTs are already competitive or even better (depending on your available resources; see VOLO)

Are there efficient low-inductive-bias + scalable models that do not rely on transformer blocks?

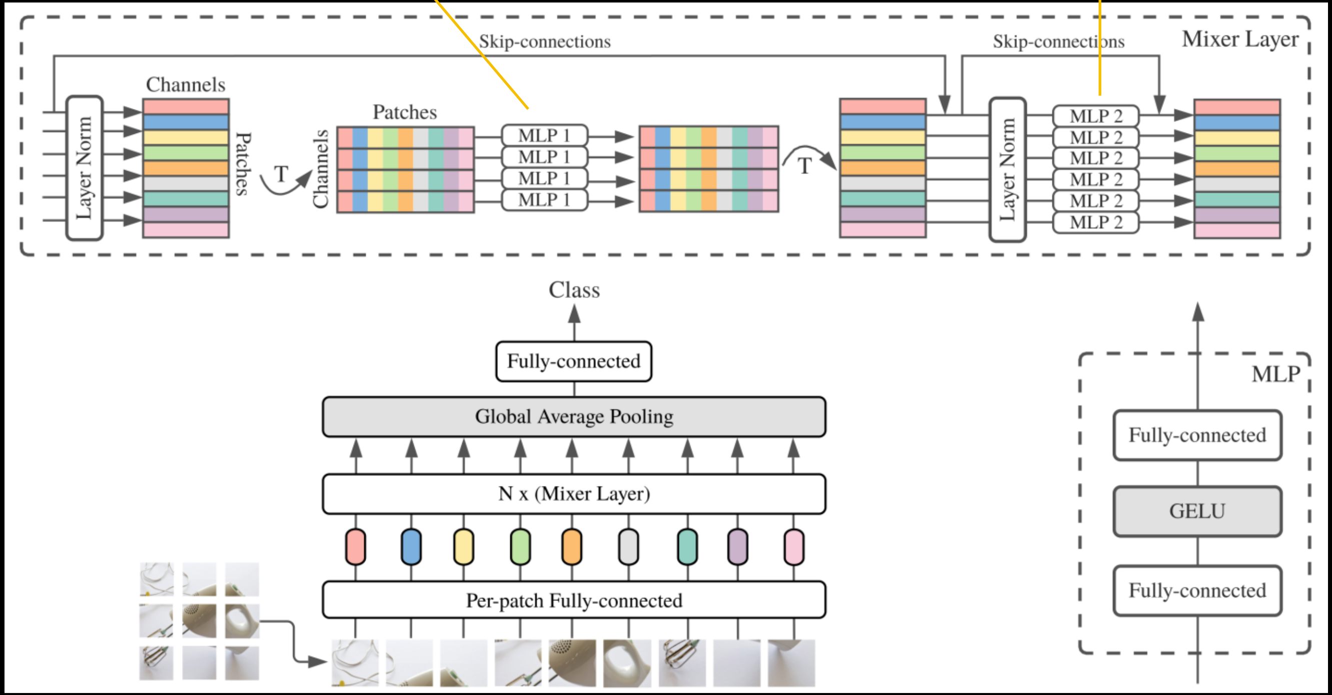
Vision MLPs (V-MLPs)

What are V-MLPs?

Combine spatial information, same per channel

Combine Channel information, same per token

MLP-Mixer:



Token/patch-based
Input like for ViT

What are V-MLPs?

- $Y = spatialMLP\big(LN(X)\big) + X,$
- $Z = channelMLP\big(LN(Y)\big) + Y,$

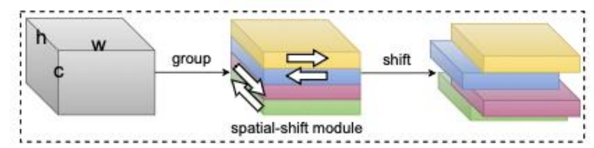
- The spatial MLP captures the global correlations between tokens
- The channel MLP combines information across features

V-MLPs Advantages and Disadvantages

- Advantages:
  - V-MLPs do not rely on self-attention, but attain global processing through fully-connected layers
  - V-MLPs, like some ViTs, have low inductive biases => generally applicable
  - V-MLPs, like some ViTs, are easy to scale
  - V-MLPs do not require positional encoding as used in ViTs
- Disadvantages:
  - V-MLPs, like some ViTs, have low inductive biases => require more parameters
  - Standard V-MLPs require a fixed input resolution: difficult for transfer learning
  - Standard V-MLPs, like initial ViTs, are less accurate compared to CNNs

V-MLPs Overview

Model	New features	#Params (M)	**FLOPs (G)**	IM-1K top-1 (%)
MLP-Mixer [17]	First isotropic V-MLP	59	12.7	76.44
RaftMLP [23]	Token-mixing along columns and rows, multi-scale embedding and bicubic interpolation	36.2	6.5	79.4
ResMLP [18]	Uses an affine transformation instead of layer normalization	116	23	81.0
PoolFormer [24]	Uses simple average pooling as token-mixer, with hierarchical architecture	73	11.9	82.5
S^2-MLPV2 [21]	Uses a special variant of depthwise-separable convolution for local spatial processing, pyramidal network, and split attention	55	16.3	83.6
Wave-MLP [25]	Uses constructive and desctructive interference to dynamically aggregate wave-like tokens	63	10.2	83.6
Hire-MLP [20]	Local and global spatial processing with hierarchical architecture	96	13.4	83.8
MS-MLP [30]	Introduces regional mixing	88	16.1	83.8
DynaMixer [26]	Uses dynamic mixing of tokens based on a trainable matrix	97	27.4	84.3



For each V-MLP, we present the accuracy of the largest reported model that is trained on IM-1K only.

V-MLPs Overview: Larger models

Model	New features	#Params (M)	**FLOPs (G)**	IM-1K top-1 (%)
MLP-Mixer	First isotropic V-MLP	59	12.7	76.44
RaftMLP	Token-mixing along columns and rows, multi-scale embedding and bicubic interpolation	36.2	6.5	79.4
ResMLP	Uses an affine transformation instead of layer normalization	116	23	81.0
PoolFormer	Uses simple average pooling as token-mixer, with hierarchical architecture	73	11.9	82.5
S^2-MLPV2	Uses a special variant of depthwise-separable convolution for local spatial processing, pyramidal network, and split attention	55	16.3	83.6
Wave-MLP	Uses constructive and desctructive interference to dynamically aggregate wave-like tokens	63	10.2	83.6
Hire-MLP	Local and global spatial processing with hierarchical architecture	96	13.4	83.8
MS-MLP	Introduces regional mixing	88	16.1	83.8
DynaMixer	Uses dynamic mixing of tokens based on a trainable matrix	97	27.4	84.3

Introducing inductive biases, ~CNN, and some efficient attention mechanisms

Accuracy improved by ~8% in just 9 months

Compare to ConvNeXt-B (89M, 45G) 83.5%

For each V-MLP, we present the accuracy of the largest reported model that is trained on IM-1K only.

V-MLPs Overview: Small models

V-MLP

Model	#Params (M)	**FLOPs (G)**	IM-1K top-1 (%)
S^2-MLPV2-small/7	25	6.9	82.0
Wave-MLP-S	30	4.5	82.6
Hire-MLP-S	33	4.2	82.1
MS-MLP	28	4.9	82.1
DynaMixer-S	26	7.3	82.7
ResNet-50	25	4.1	79.8
RegNetY-8GG	39	8	82.1
ConvNeXt-T	29	4.5	82.1

CNN

Small V-MLPs are as good or better compared to similarly-sized CNNs or ViTs
(Besides VOLO, which builds upon special Token-Labeling-based training)

Model	#Params (M)	**FLOPs (G)**	IM-1K top-1 (%)
SwinT-T	29	4.5	81.3
ViTC-4GG	17.8	4	81.4
VOLO-D1	27	6.8	84.2

ViT

For each model, we present the accuracy of models with ~30M params that are trained on IM-1K only.

V-MLPs Summary

- V-MLPs started as Isotropic, shallow, low-inductive bias networks without self-attention
- Over time, V-MLPs became hybrid, deeper and added inductive biases typically found in CNNs (local processing, hierarchical architecture)
- This evolution made V-MLPs more efficient, easier to train and more accurate

V-MLPs Conclusions

- V-MLPs are typically more accurate when pre-trained on large datasets (IM-22K, JFT-300M), but little research here with CNNs and recent ConvNext comes close
- V-MLPs can have less inductive biases so more applicable to general (multi-modal) data
- V-MLPs are easy to scale, so easier to adapt to large (multi-modal) datasets
- CNNs are typically more accurate when trained on small datasets <u>BUT V-MLPs have become very competitive or better in just few months</u>
- CNNs are typically more computationally efficient for CV due to their inductive biases <u>BUT V-MLPs have become very competitive or sometimes better in just few months</u>
- CNNs often rely on easier dataflows: BN vs. LN, softmax in final layer vs. after SE, ReLU vs. GELU, etc., convolution vs. spatial shifting

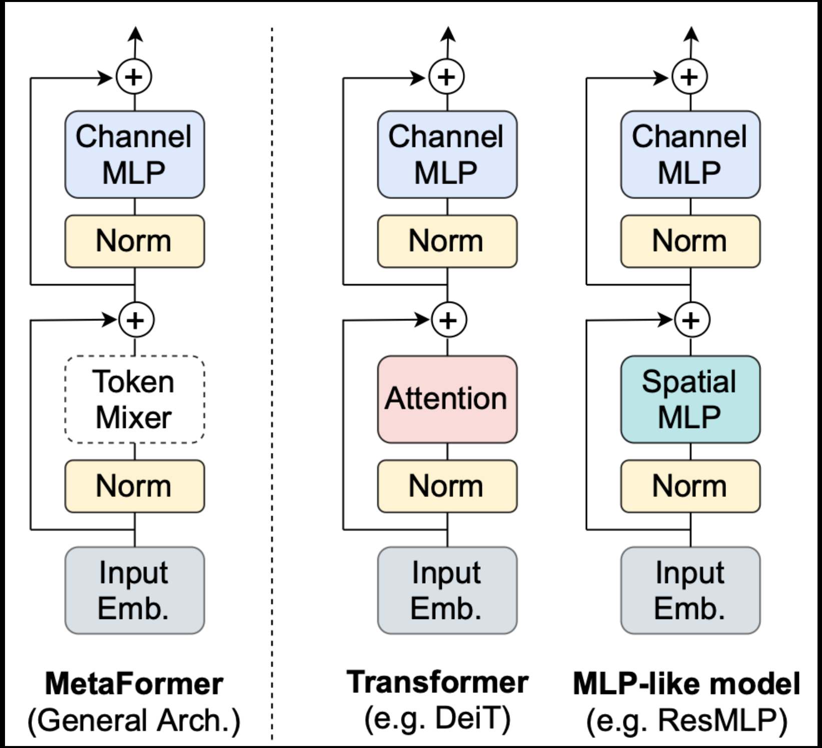
-> <u>For typical edge CV V-MLPs may become a good alternative to CNNs</u>

Conclusions

Conclusions

- Recent ViTs (and V-MLPs) can be used at the edge but seem mostly appropriate when:
  - Datasets are multi-modal
  - Datasets are large
  - Compute is abundant
  - Regular compute patterns are required (isotropic models)
- For typical edge CV tasks, there's currently no evidence that (self)-attention is a necessary ingredient for good accuracy
- Some evidence that attention can slightly improve accuracy
- Several attention mechanisms have been proposed, classical (QKV) self-attention is not necessary
- V-MLPs improve upon ViTs, and are close to, or in some cases already, outperforming CNNs
- Hybrid models can combine the best of several worlds

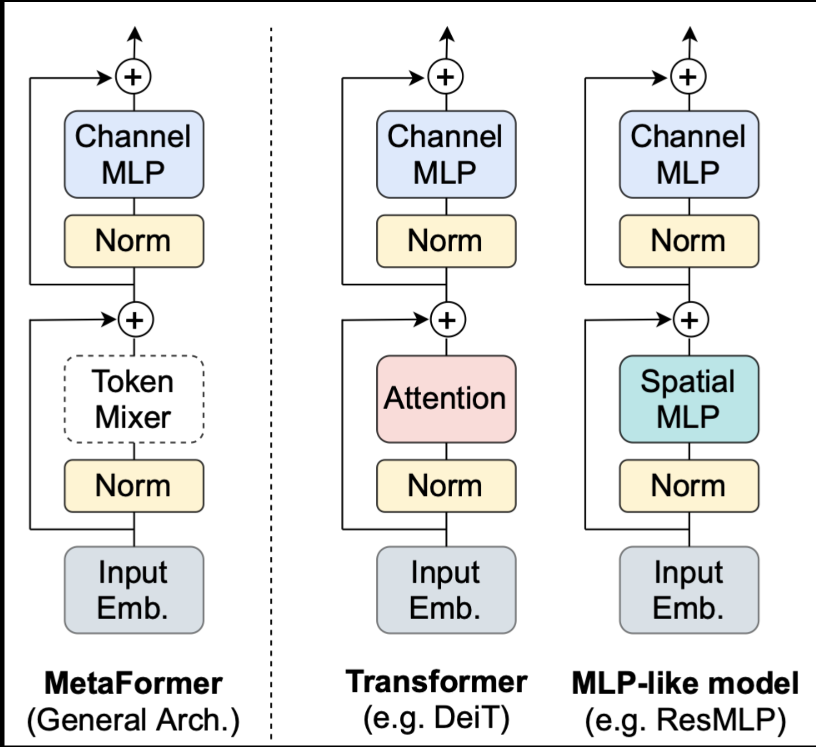
What Matters for efficient and accurate CV?



[Yu et al., “MetaFormer is Actually What You Need for Vision”]

- A single formulation for these models can be devised.
- Two main components: Token mixer and channel mixer.
- Even CNNs (e.g. MobileNet) can se seen as fitting the pattern presented here (“MetaFormer”).

What Matters for efficient and accurate CV?



- Non-overlapping spatial patch-embeddings
- Token and Channel mixing
- Normalization
- Residual connection
- Local processing (large enough)
- Hierarchical processing
- Hybrid models

Thank you!