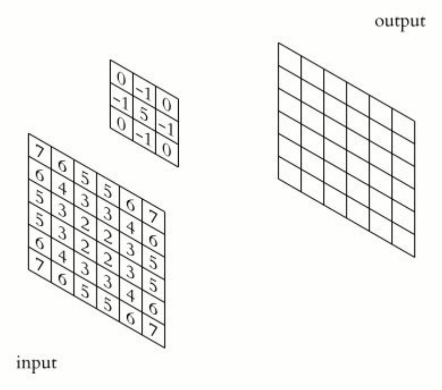
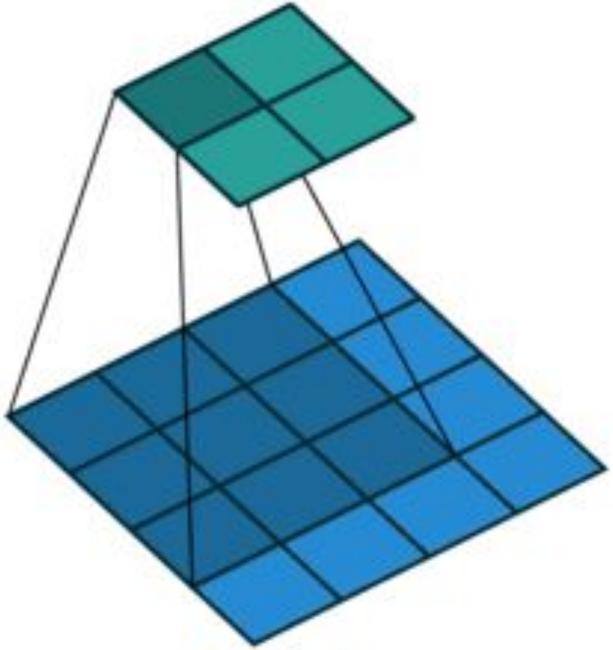
ANN: …trainable layers…

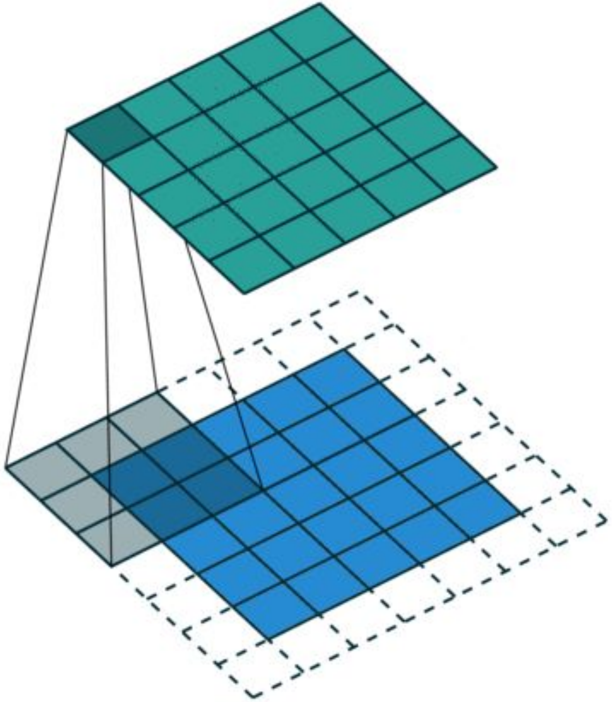


convolution
- every input is connected to the output by a fixed kernel
- kernel is (a)symmetrical (e.g. 3×3, 3×1)
- some input can be skipped (strides > 1 and/or dilation > 1)
- input can be 1D, 2D, 3D, … × channels
- output channel numbers determined by number of filters
- only one set of kernel weights (computationally efficient)
- convolutional 1D layer with 1 kernel == dense layer
- inputs•kernel•activation = output
- inputs•kernel•activation + bias = output
- output size >|=|< input size e.g. [*128*, 16, 16, 32] => [*128*, 8, 8, 32]

(https://commons.m.wikimedia.org/wiki/File:2D_Convolution_Animation.gif)

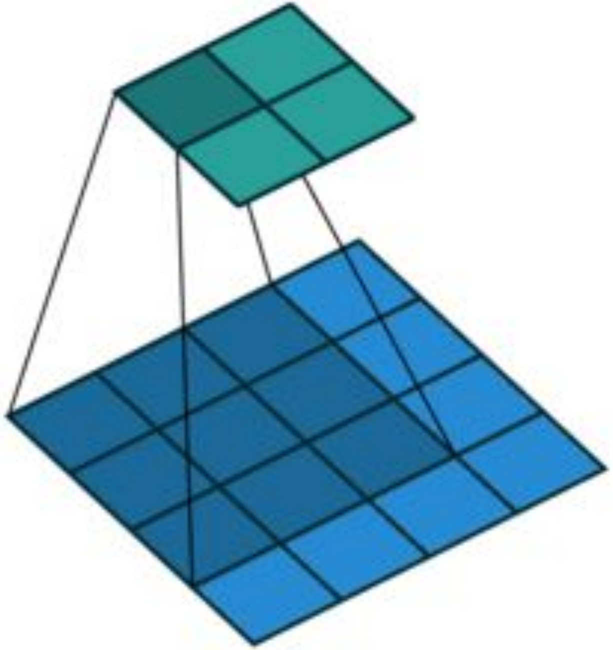


convolution
('valid' padding, strides 1)

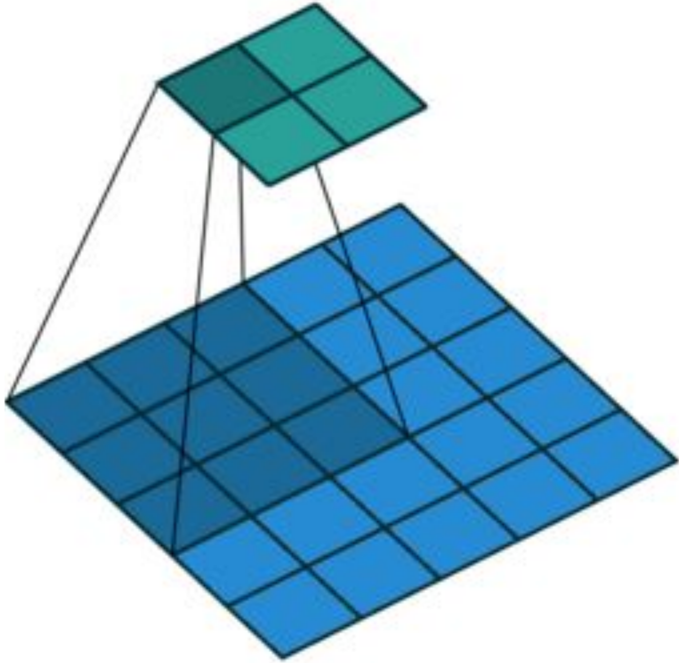


convolution
('same' padding, strides 1)

(Dumoulin and Visin 2016; arXiv:1603.07285; https://github.com/vdumoulin/conv_arithmetic)

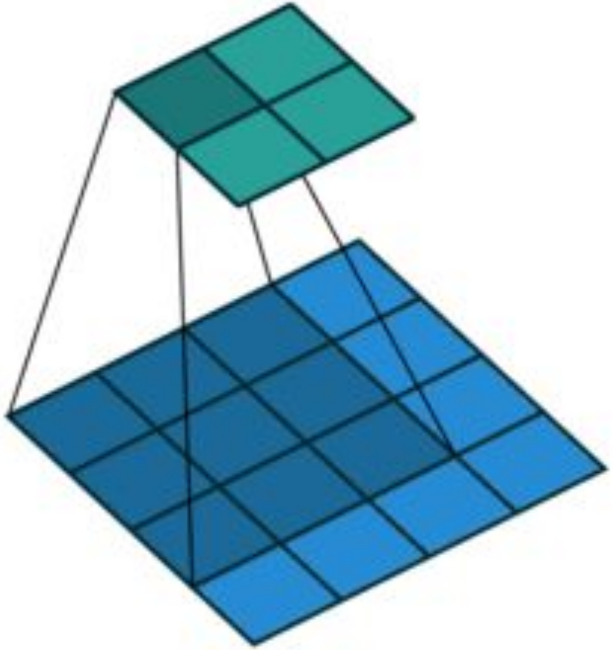


convolution
(‘valid’ padding, strides 1)

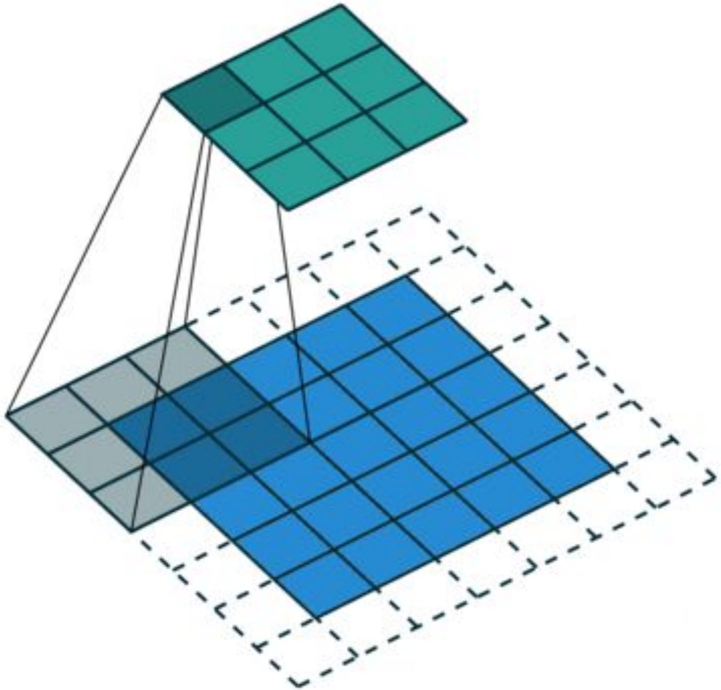


convolution
(‘valid’ padding, strides 2)

(Dumoulin and Visin 2016; arXiv:1603.07285; https://github.com/vdumoulin/conv_arithmetic)

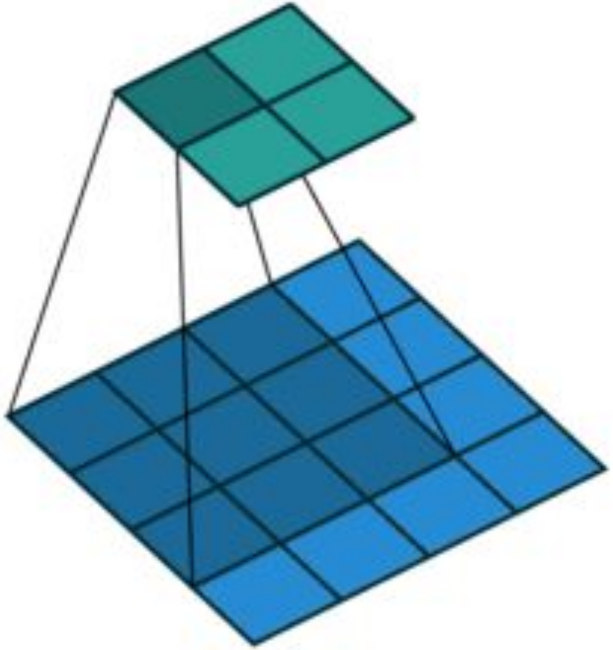


convolution
('valid' padding, strides 1)

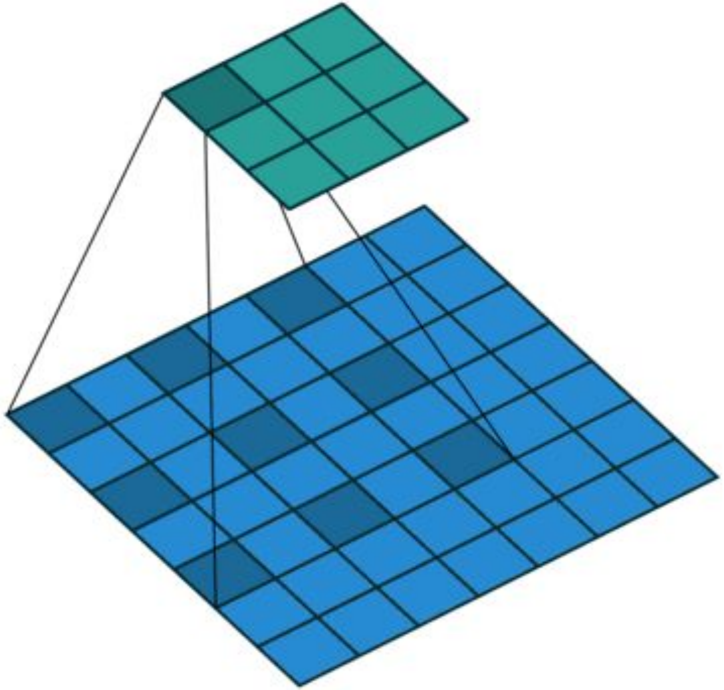


convolution
('same' padding, strides 2)

(Dumoulin and Visin 2016; arXiv:1603.07285; https://github.com/vdumoulin/conv_arithmetic)

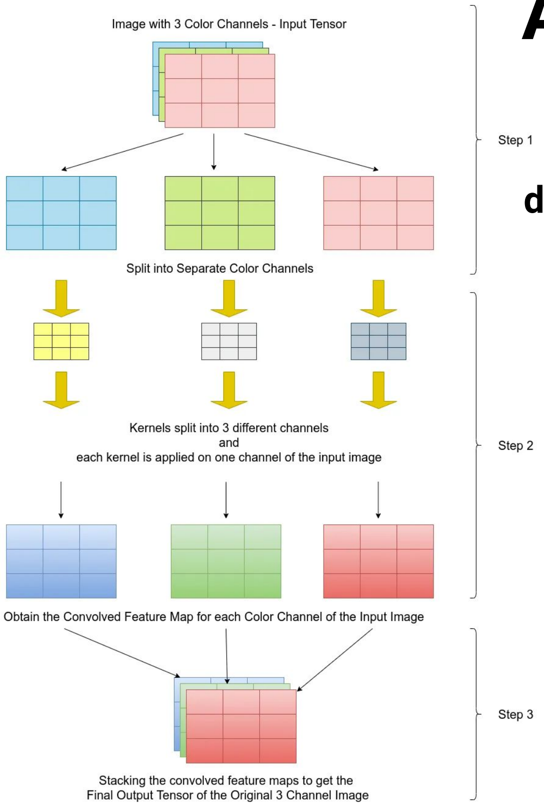


convolution
(‘valid’ padding, strides 1, dilation 1)



convolution
(‘valid’ padding, strides 1, dilation 2)

(Dumoulin and Visin 2016; arXiv:1603.07285; https://github.com/vdumoulin/conv_arithmetic)



ANN: …trainable layers…

depthwise convolution

separate convolution of channels

one kernel per channel

[0] seperate channels

[1] convolute each channel

[2] concatenate channels

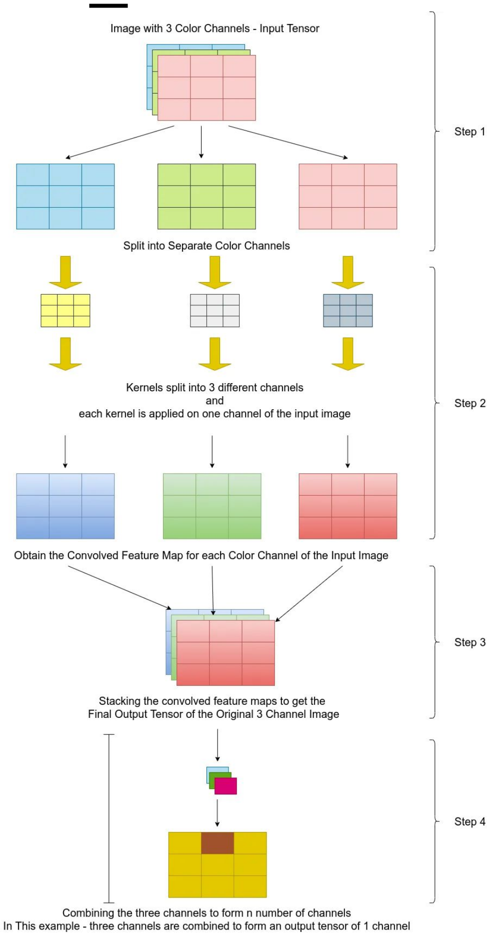
=> many fewer multiplications

faster on CPU, but slower on GPU

output size >|=|< input size

e.g. [*128*, 16, 16, 32] => [*128*, 8, 8, 32]

(https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets-6de3d6b62503)



ANN: …trainable layers…

separable convolution	depthwise + standard convolution
	one kernel per channel
	[0] seperate channels
	[1] convolute each channel
	[2] concatenate channels
	[3] 1×1 convolute
	=> many fewer multiplications
	faster on CPU, but slower on GPU
	output size >|=|< input size
	e.g. [*128*, 16, 16, 32] => [*128*, 8, 8, 32]

(https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets-6de3d6b62503)

ANN: …trainable layers…

deconvolution

a.k.a. transposed convolution

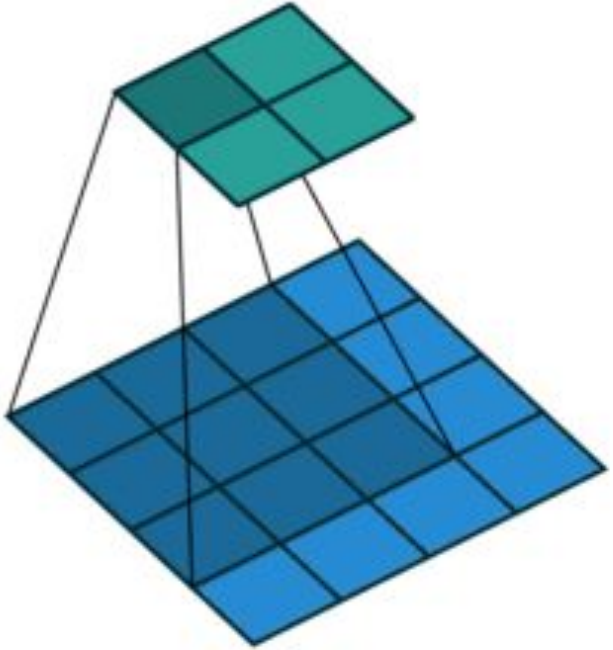
increases tensor time/height/width

a learnable upsampling

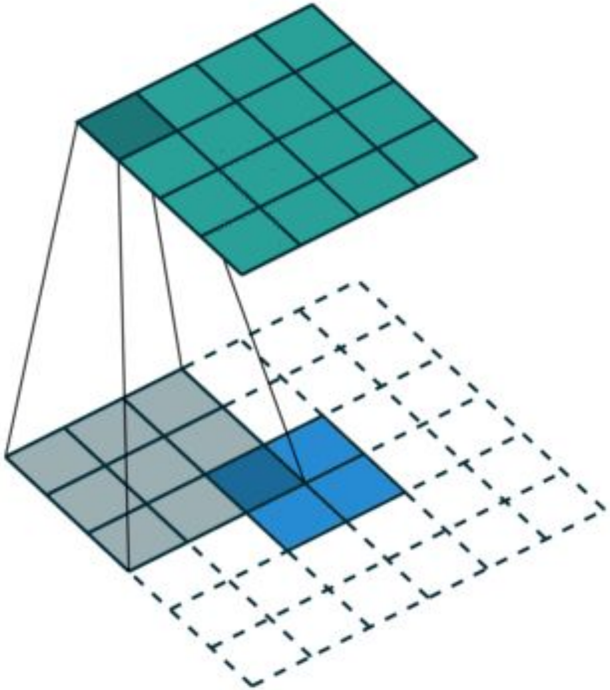
output size >|=|< input size

e.g. [*128*, 16, 16, 32]

=> [*128*, 32, 32, 32]



convolution
('valid' padding, strides 1)



deconvolution
('valid' padding, strides 1)

(Dumoulin and Visin 2016; arXiv:1603.07285; https://github.com/vdumoulin/conv_arithmetic)

ANN: …trainable layers…

LSTM Long Short–Term Memory

stores data from the last *n* inputs

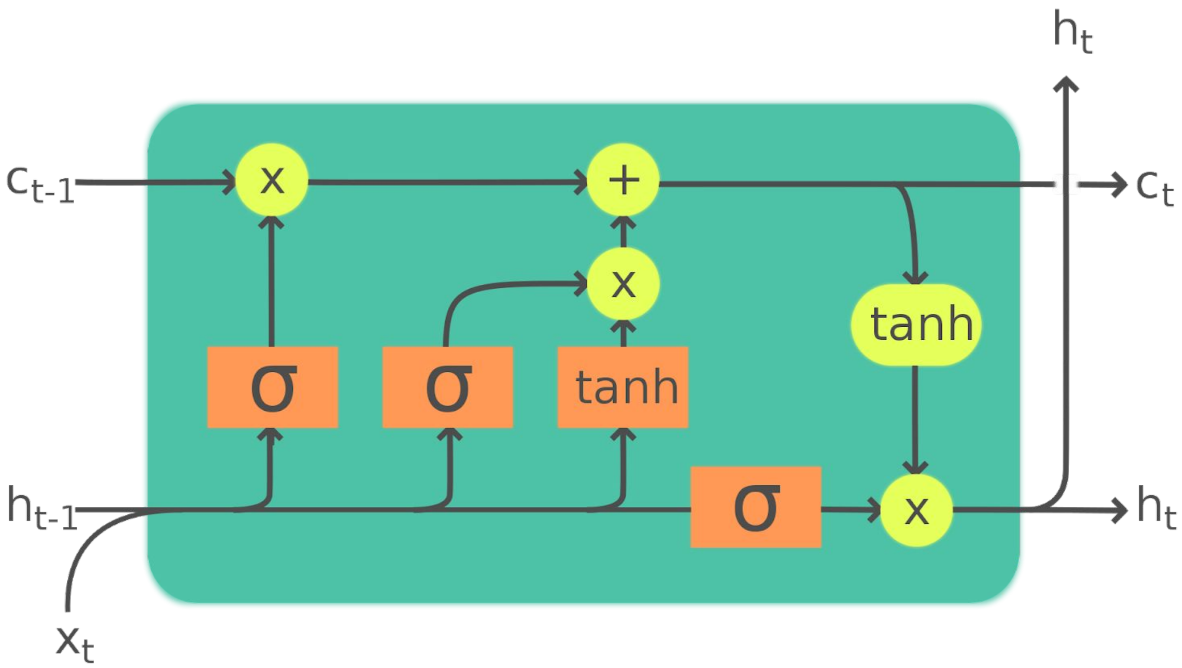
hidden state and cell state act as ‘memory’

input gate, output gate, and forget gate

used singly or in a bidirectional configuration

upweight ‘important’ elements

downweight ‘unimportant’ elements



(https://commons.wikimedia.org/wiki/File:LSTM_Cell.svg)

ANN: …trainable layers…

GRU Gated Recurrent Unit

a simplified LSTM

stores data from the last *n* inputs

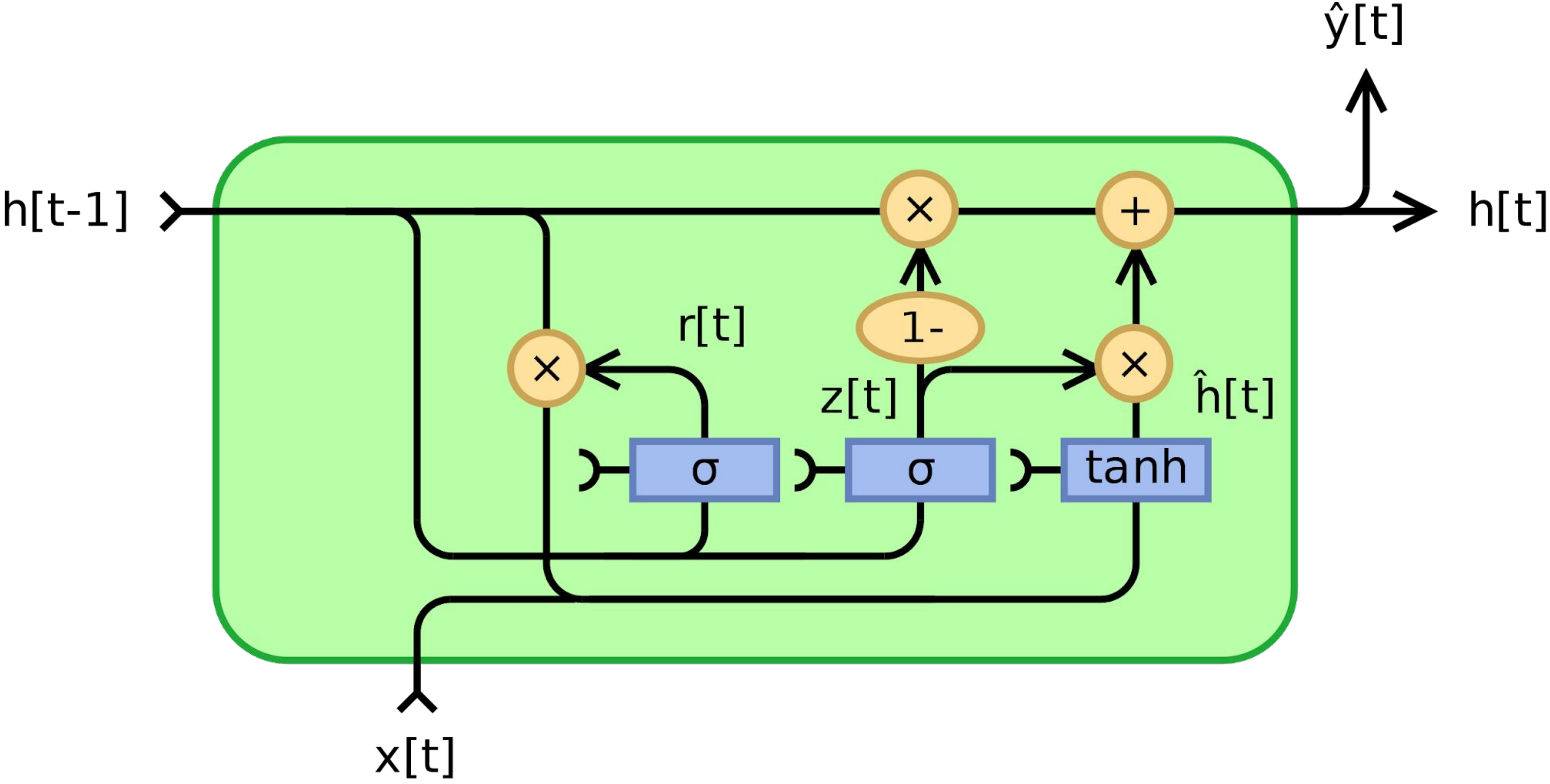
hidden state acts as 'memory'

update gate and reset gate

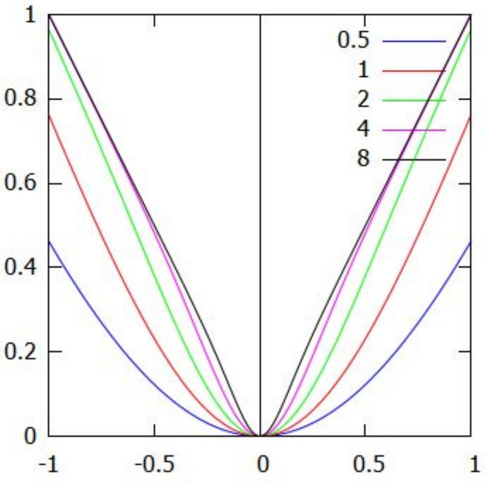
used singly or in a bidirectional configuration

upweight 'important' elements

downweight 'unimportant' elements



(https://en.wikipedia.org/wiki/File:Gated_Recurrent_Unit,_type_1.svg) softmax



ANN: ...trainable layers

output typically a dense layer with activation

regression = tanh [-1, +1] or sigmoid [0, 1]

classification = softmax [0, 1] (sum to 1)

multiclass classification = sigmoid [0, 1]

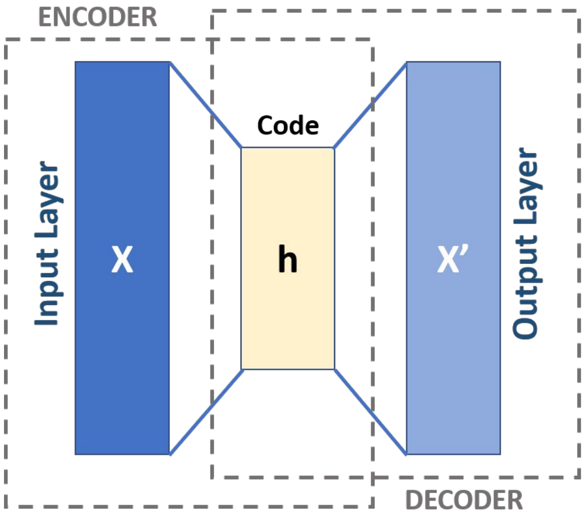
often preceded by a global average pooling layer

(https://en.wikipedia.org/wiki/File:Smoothmax.png)

ANN: block patterns…

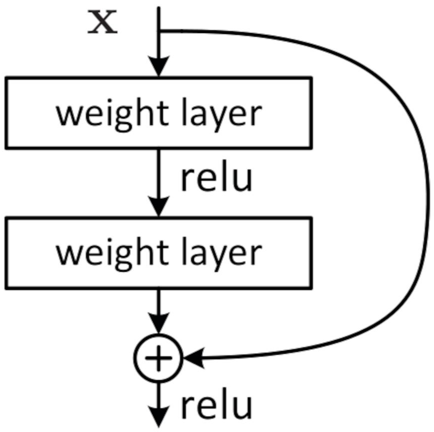
auto encoders/bottlenecks

- minimum of three layers: large => small => large
- size reduction forces compression
    - important information is retained
    - less important information and noise is lost



(https://commons.wikimedia.org/wiki/File:Autoencoder_schema.png)

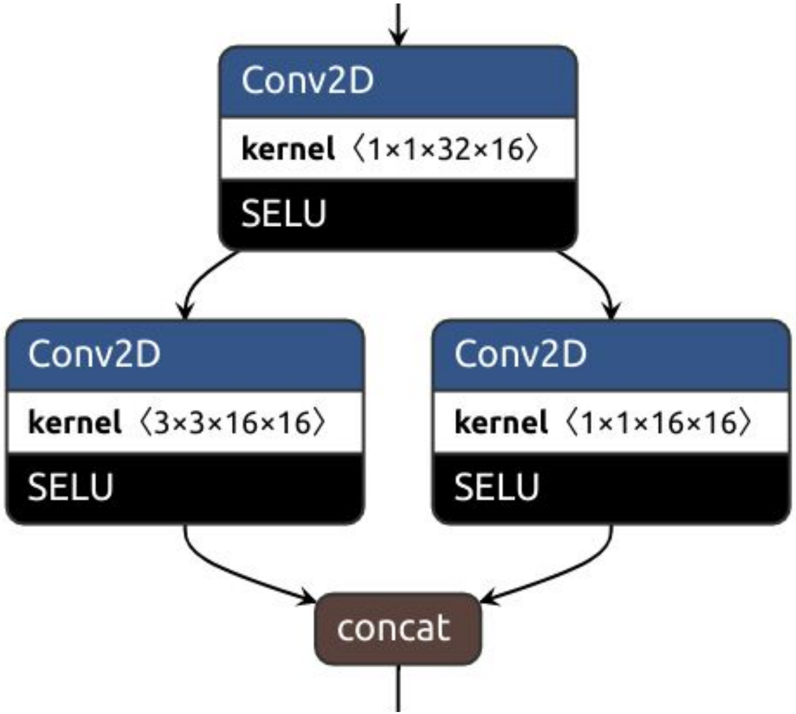
ANN: …block patterns…



residual connection (a.k.a. skip connection)

- allows for deeper networks
  - signal is lost in the Back Propagation (BP) algorithm
  - not an issue with Direct Feedback Alignment (DFA)
- can differentially weight 'important' features
- structure:
  - [0] copy data into two (or more) paths
  - [1] process each path separately (e.g. convolutions)
  - [2] recombine paths (usually addition or concatenation)

(He et al. 2015; https://doi.org/10.48550/arXiv.1512.03385)

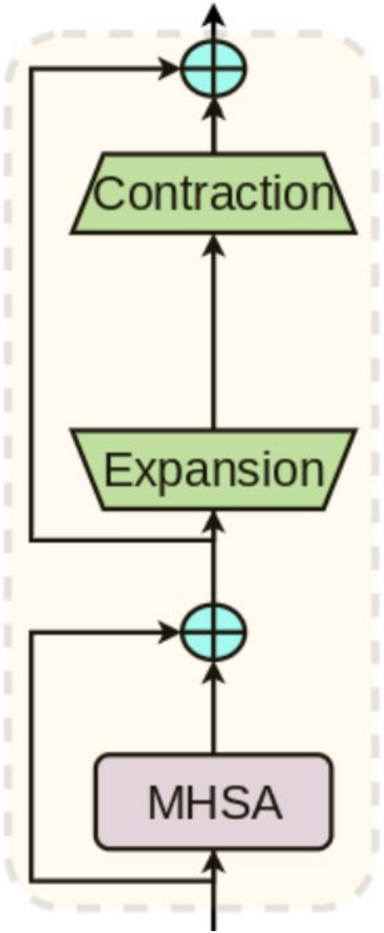


SqueezeNet Fire module (Iandola et al. 2016; https://arxiv.org/abs/1602.07360)

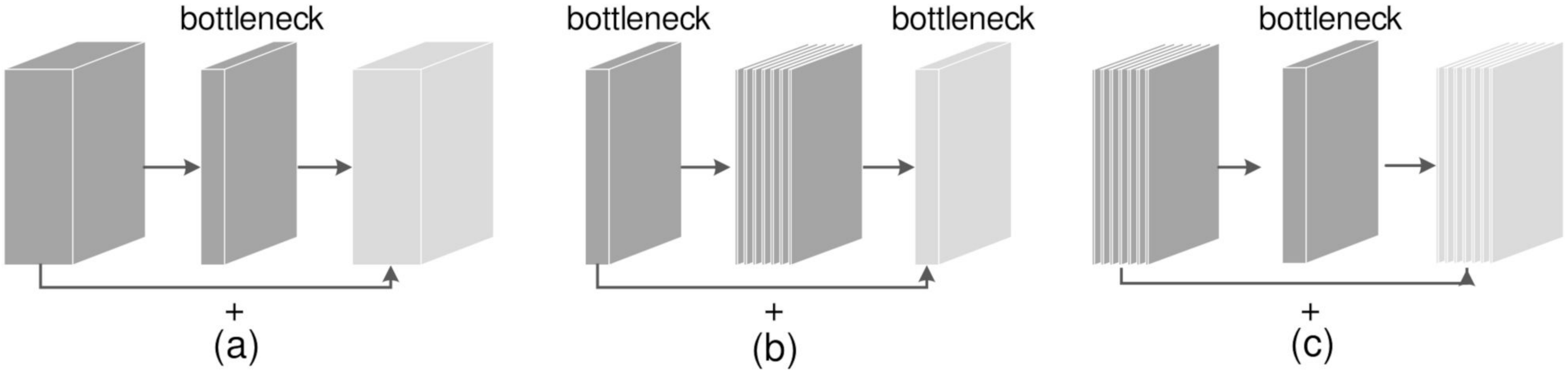
ANN: …block patterns…

inverted residual blocks

- minimum of two layers: expand => contract
- size change forces information recoding and filtering
- structure:
  - [0] split or copy data into two (or more) paths
  - [1] process (or not) each path separately
  - [2] recombine paths (usually addition)



MLP inverted bottleneck (Vaswani et al. 2017 fide Srinivas et al. 2101.11605; https://arxiv.org/abs/2101.11605)



(Zhou et al. 2017; https://arxiv.org/abs/2007.02269)

ANN: …block patterns…

distributed convolutions

- designed to increase the receptive field
- more efficient than a single large–kernel convolution
- structure:
  - [0] split or copy data into two (or more) paths
  - [1] convolute each path separately
  - [2] recombine paths (usually concatenation)

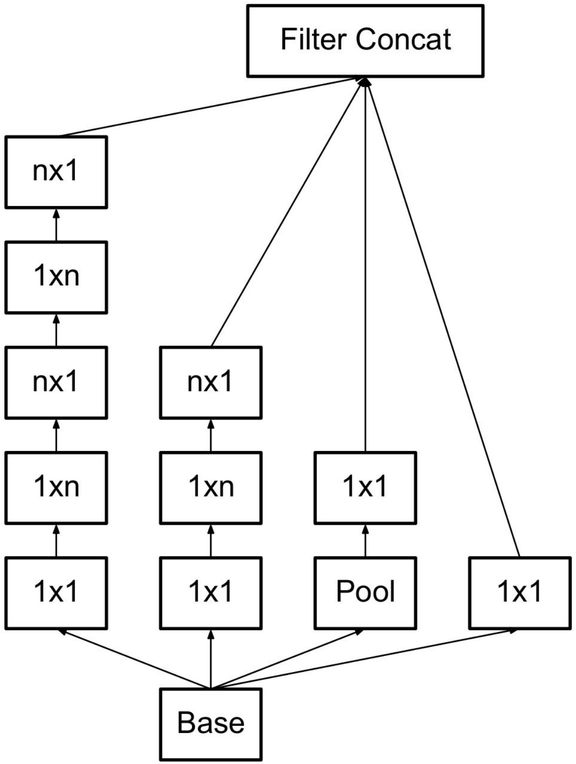


Figure 6. Inception modules after the factorization of the $n \times n$ convolutions. In our proposed architecture, we chose $n = 7$ for the 17×17 grid. (The filter sizes are picked using principle 3)

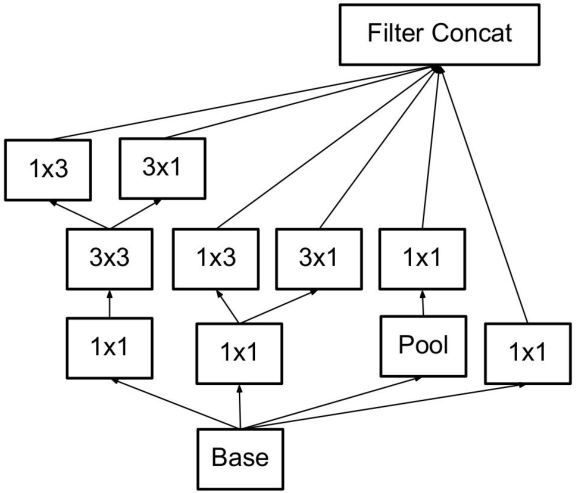
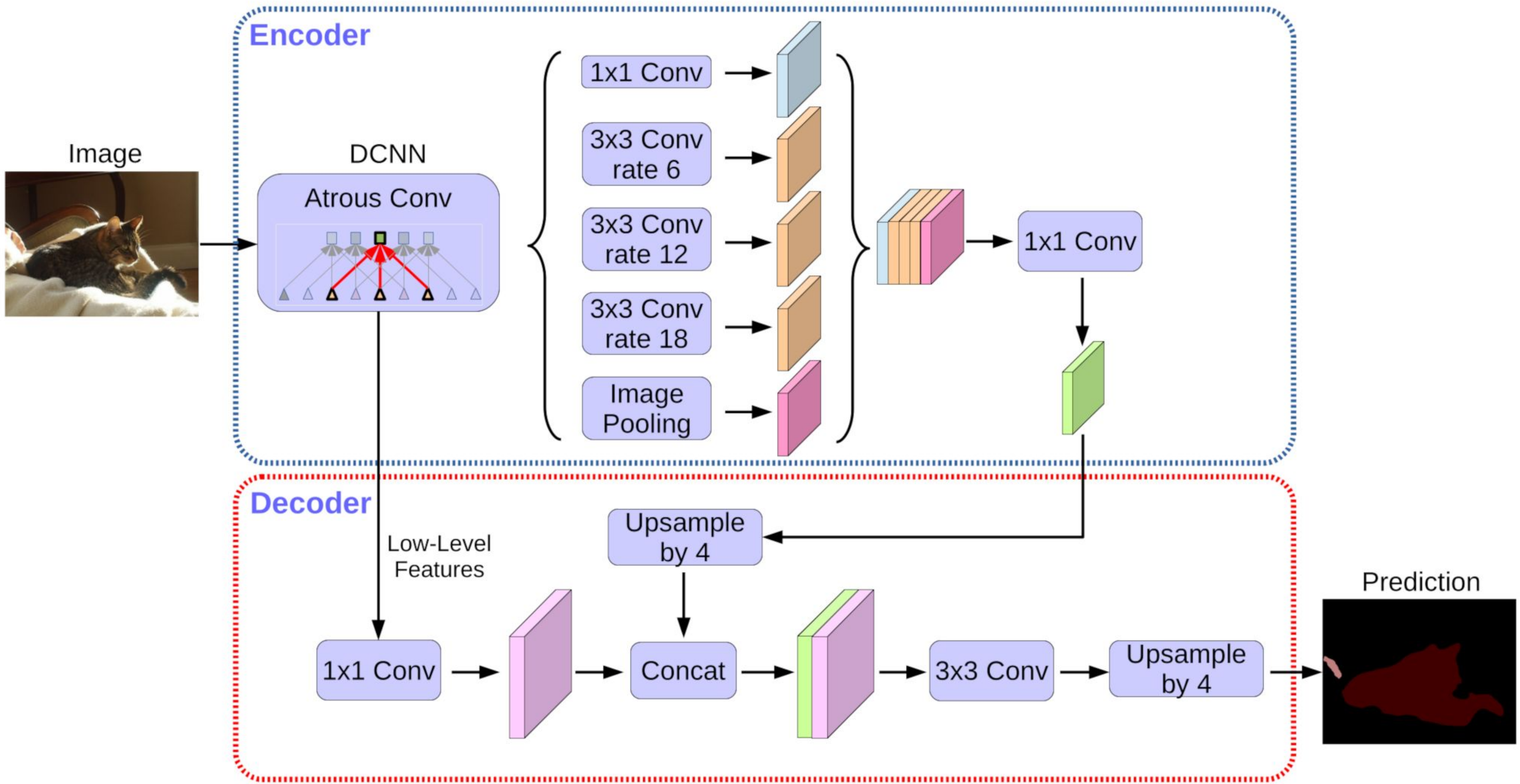


Figure 7. Inception modules with expanded the filter bank outputs. This architecture is used on the coarsest (8×8) grids to promote high dimensional representations, as suggested by principle 2 of Section 2. We are using this solution only on the coarsest grid, since that is the place where producing high dimensional sparse representation is the most critical as the ratio of local processing (by 1×1 convolutions) is increased compared to the spatial aggregation.

Inception v3 (Szegedy et al. 2015; https://arxiv.org/abs/1512.00567)



DeepLabv3+ (Chen et al. 2018; https://arxiv.org/abs/1802.0261)

ANN: …block patterns…

(self) attention

- designed to differientally weight ‘important’ features
- (usually) more efficient than convolutions or residual blocks alone
- structure:
  - [0] split or copy data into two (or more) paths (heads)
  - [1] process each path separately (e.g. convolutions)
  - [2] activate with [0, 1] output (e.g. softmax, sigmoid)
  - [3] recombine paths (usually with multiplication)

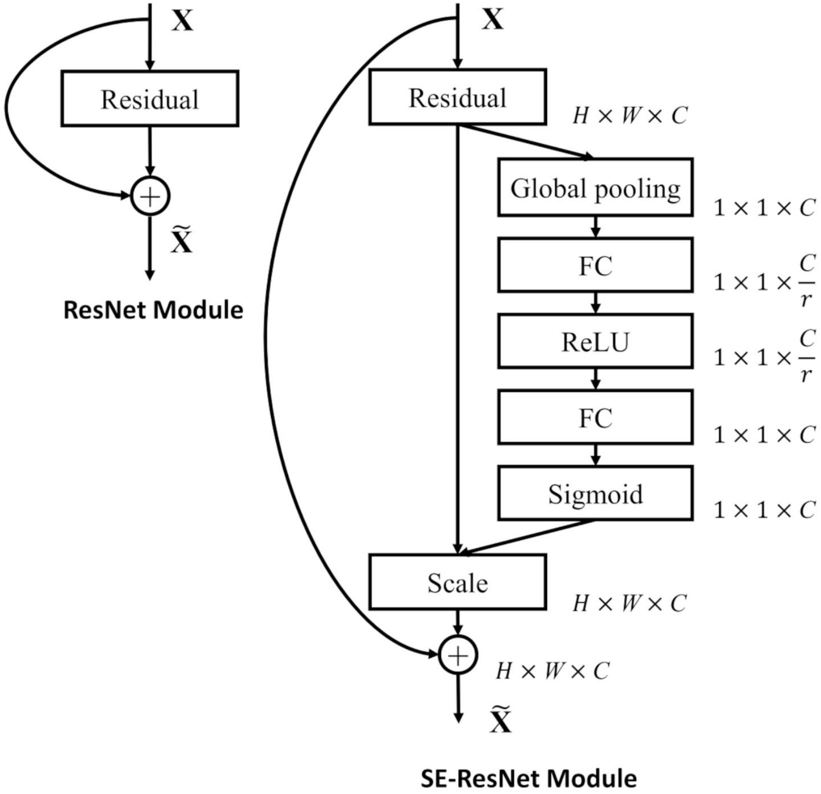
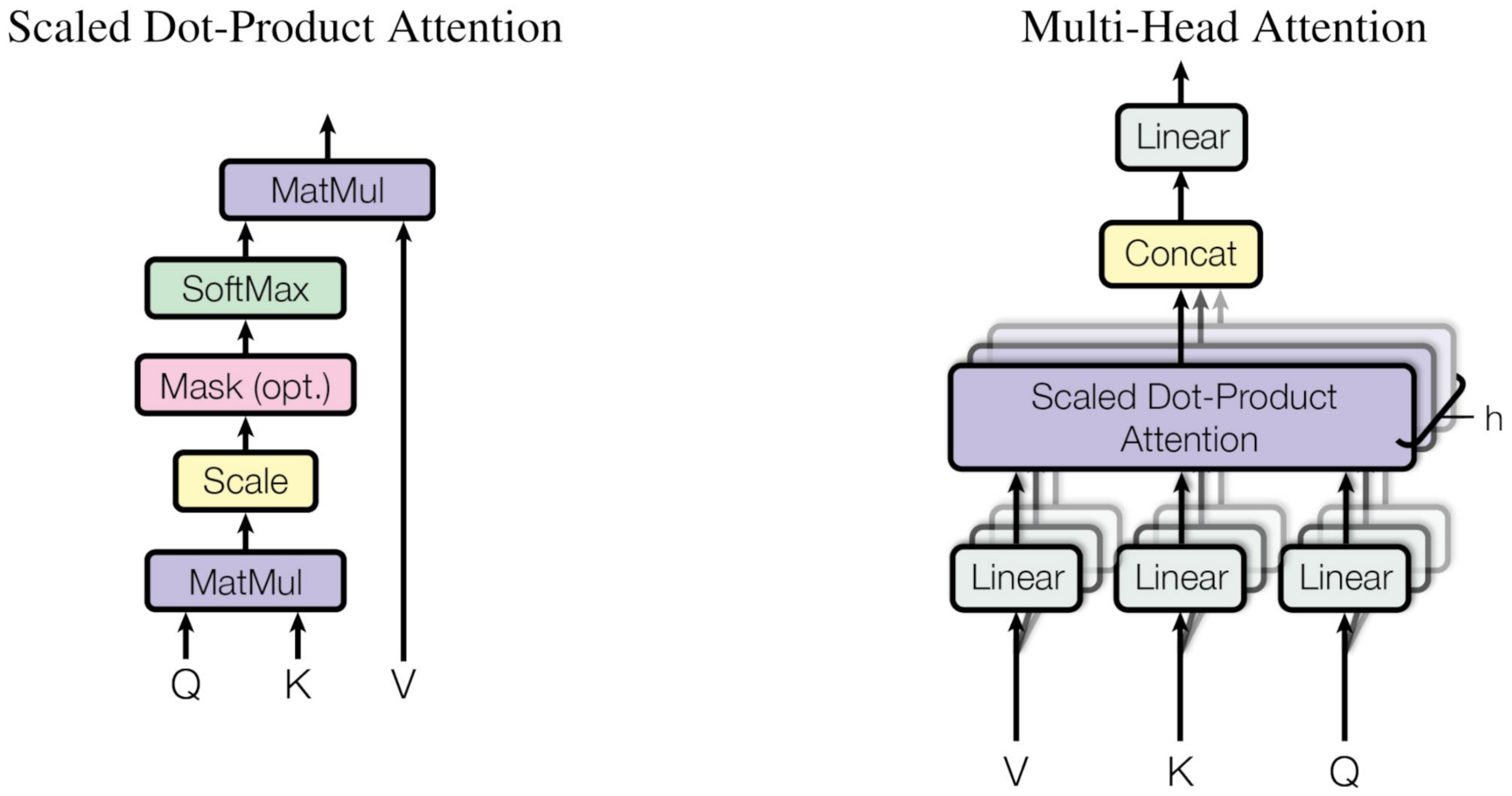
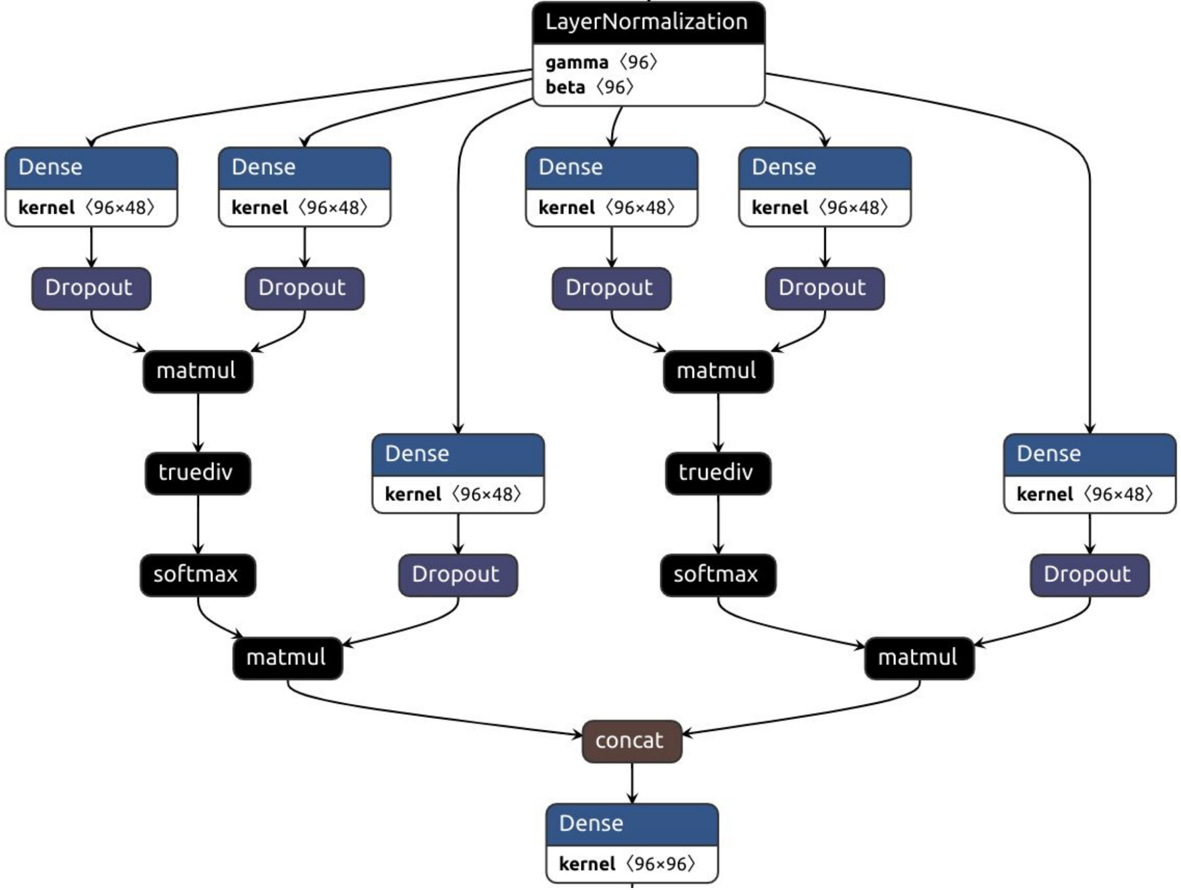


Fig. 3. The schema of the original Residual module (left) and the SE-ResNet module (right).

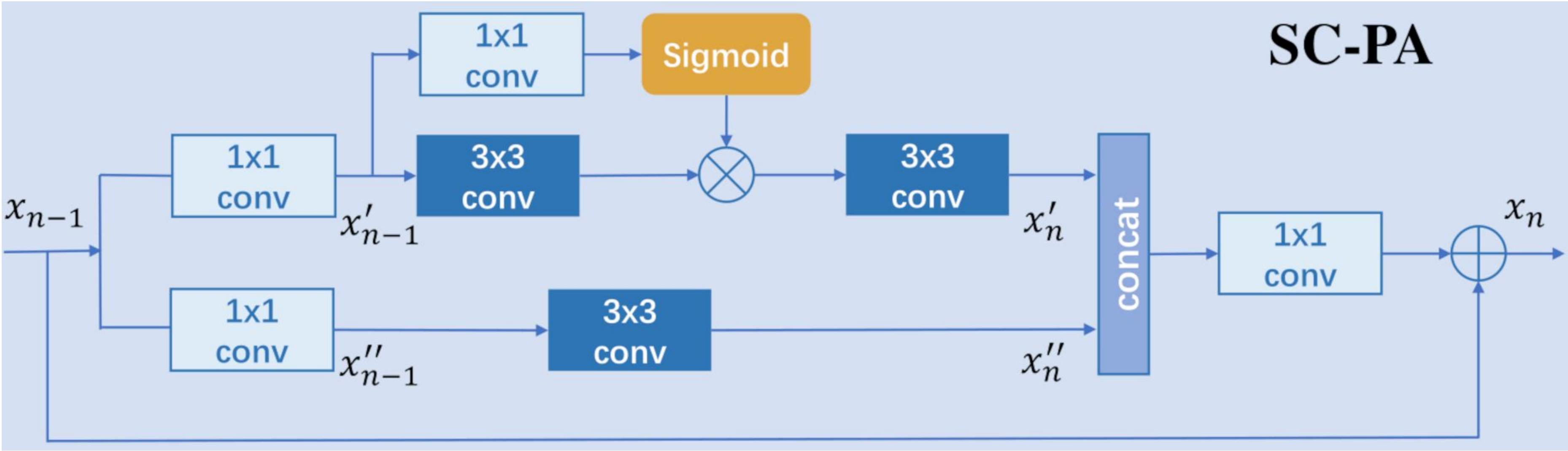
Squeeze–and–Excitation (Hu et al. 2017; https://arxiv.org/abs/1709.01507)



Multi–Head Self–Attention (Vaswani et al. 2017; https://arxiv.org/abs/1706.03762)



Multi–Head Self–Attention (Vaswani et al. 2017; https://arxiv.org/abs/1706.03762)

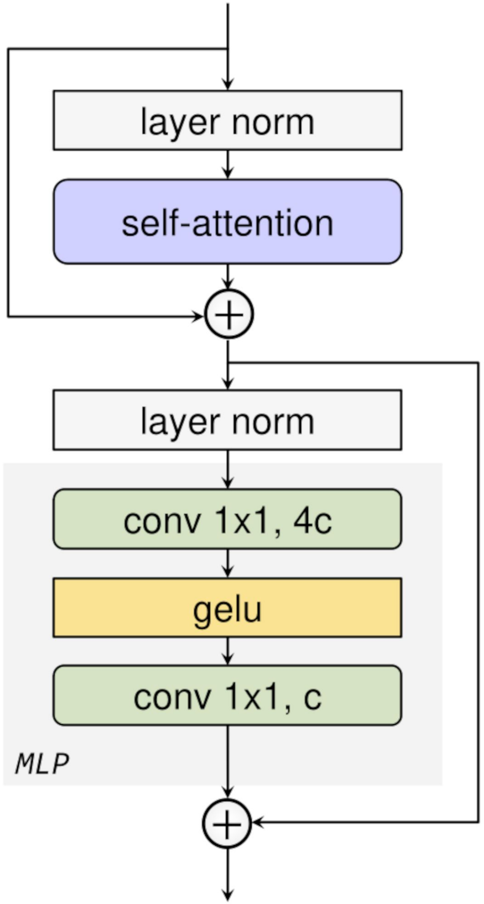


Self−Calibrated Pixel Attention (Zhao et al. 2020; https://arxiv.org/abs/2010.01073)

ANN: …block patterns…

Multi–Layer Perceptron (MLP)

- in general, any multilayer network
- specifically, used as a recoding layer in metaformers
- architecture:
  - a layer–wise normalization layer
  - 2 (or more) dense layers
    - same size or expand then contract channels
    - activate the first and/or the second
  - an additive residual connection to the input

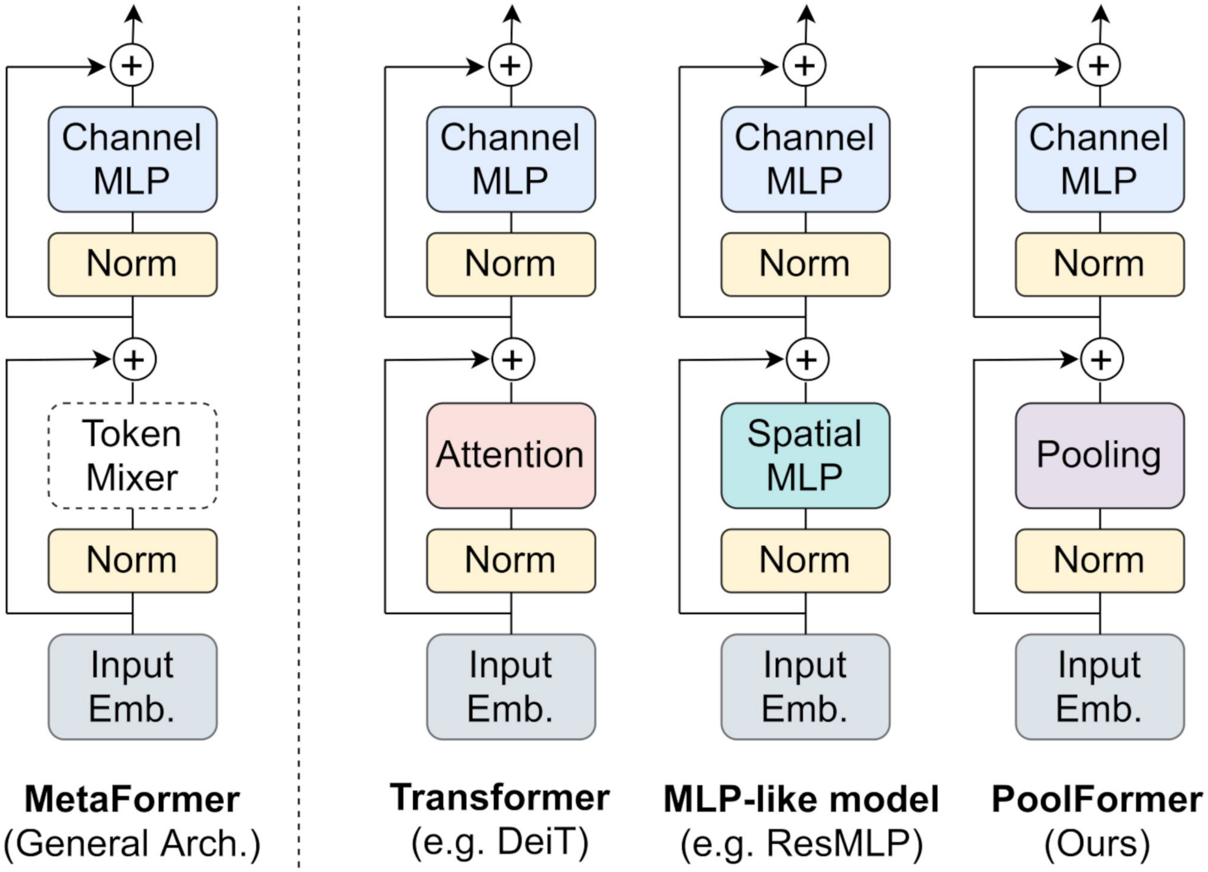


MLP in a transformer (Yang et al. 2023; https://arxiv.org/abs/2210.01820)

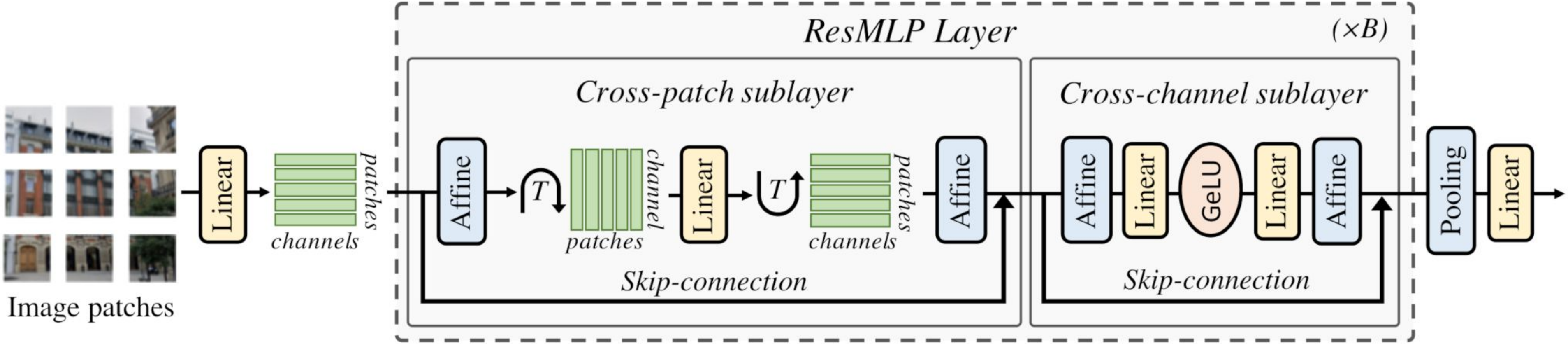
ANN: …block patterns…

metaformers (transformers and their modifications)

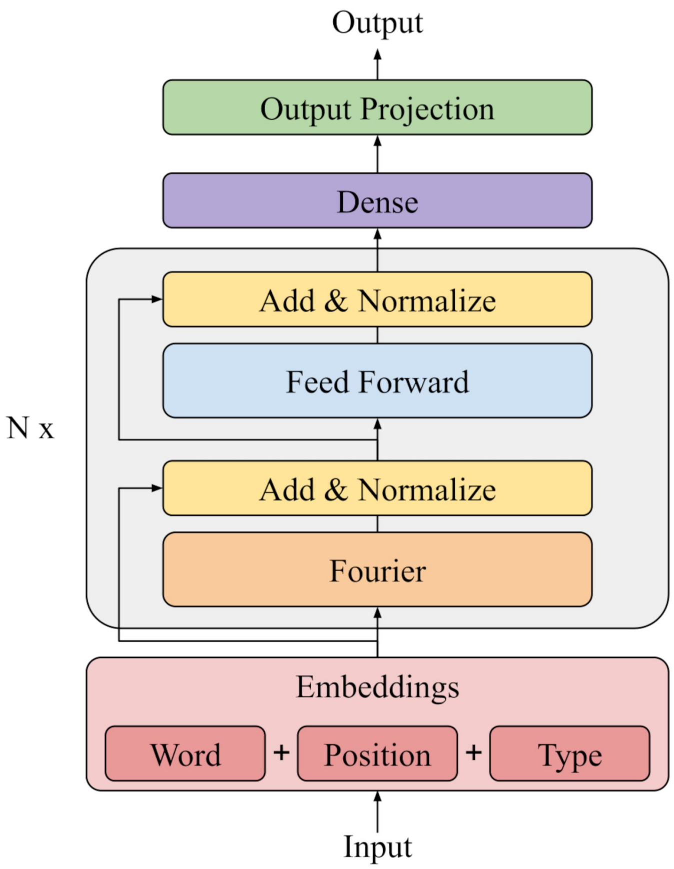
- initially designed as a better LSTM for NLP
  - used with tokenized text
  - used for computer vision with tokenized images
- architecture:
  - self–attention, smoothing, or mixing unit
    - an additive residual connection to the input
  - an ML unit



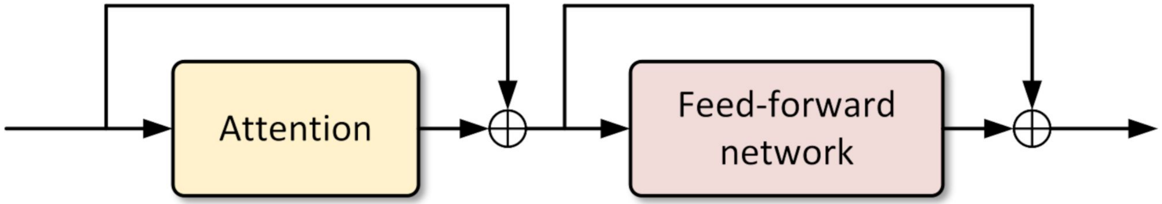
MetaFormer (Yu et al.; https://arxiv.org/abs/2111.11418)



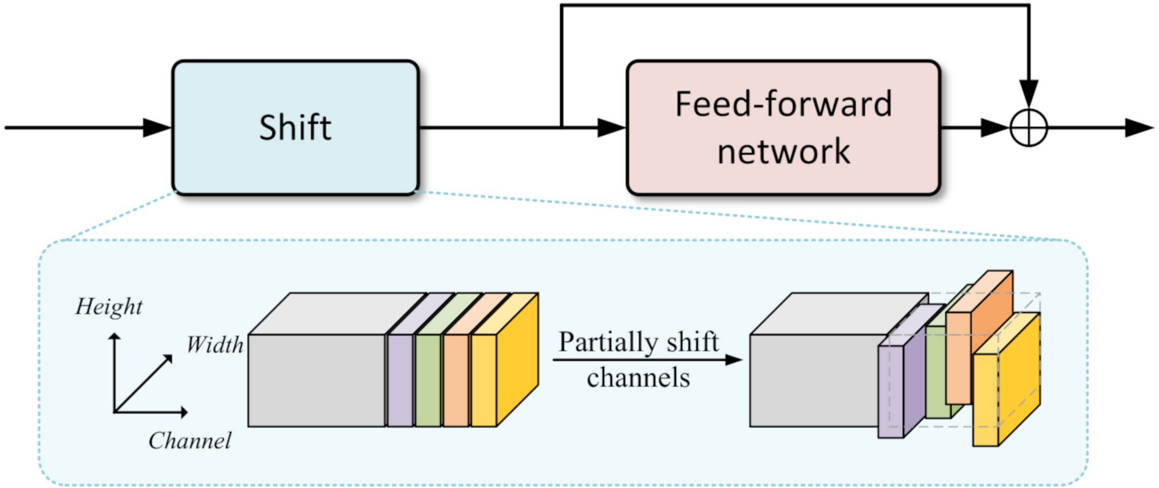
Touvron et al. 2021: Fig. 1 (ResMLP; Residual Multi–Layer Perceptrons)



Fast Fourier Transform Attention (Lee–Thorp et al. 2022; https://arxiv.org/abs/2105.03824)



(a) Standard attention building block.



(b) Our shift building block.

Shift Attention (Wang et al. 2022; https://arxiv.org/abs/2201.10801)

ANN: model designs…

inversion (e.g. SqueezeNet, Inception, ResNet, EfficientNet)

- used for classification or regression
- h $\times$ w $\times$ c => (1/*n*)h $\times$ (1/*n*)w $\times$ *m*c | w $\times$ c => (1/n)w $\times$ mc
  - signal size: large => small
  - channels: few => many
- many repeated blocks with residual connections
  - h $\times$ w reduction between blocks or in first block of a repeat
- usually bottlenecks or inverted bottlenecks within each block
- sometimes with attention mechanisms within or between blocks

ANN: …model designs…

constant size and shape (e.g. transformers)

used for classification, regression, or translation

many repeated blocks with residual connections

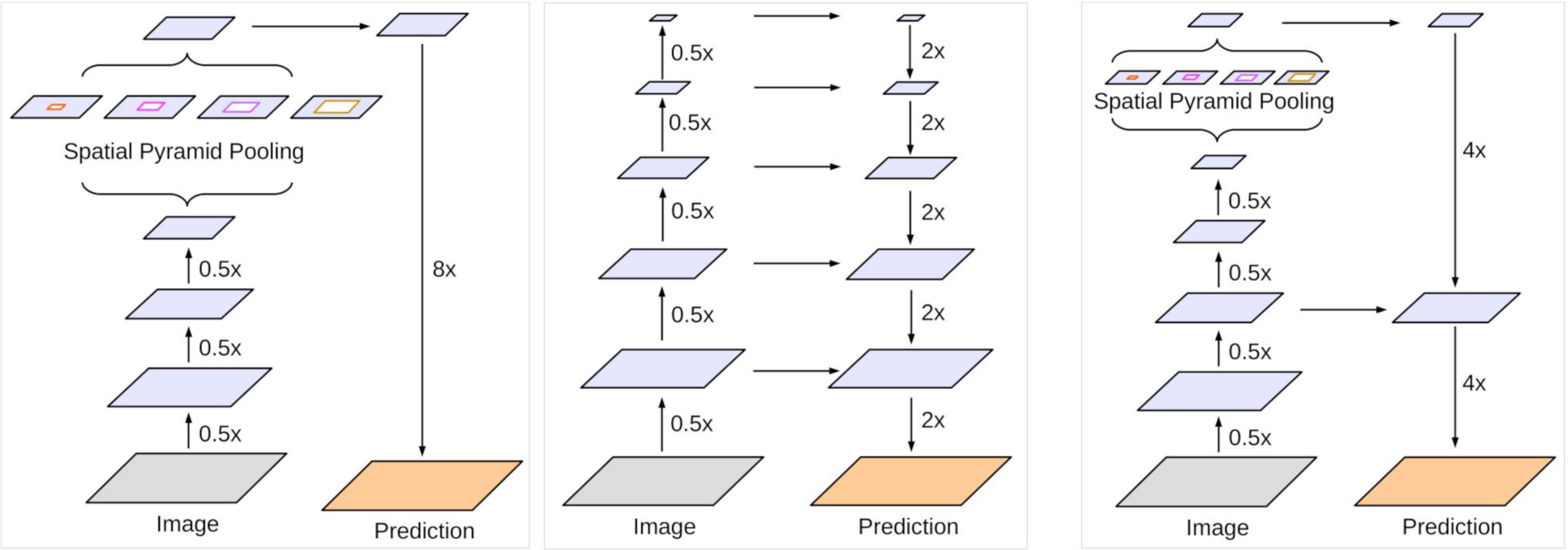
usually with attention, smoothing, or mixing

bottlenecks or inverted bottlenecks within each block

ANN: …model designs

bottleneck and unet (e.g. autoencoders)

- used for translation
- rarely with bottlenecks or inverted bottlenecks in blocks
- many repeated blocks
  - rarely with bottlenecks or inverted bottlenecks in blocks
    - the whole network is one giant bottleneck
  - with no residual connections between blocks
    - or long residual connections between blocks (unet)



(a) Spatial Pyramid Pooling   (b) Encoder-Decoder   (c) Encoder-Decoder with Atrous Conv

DeepLabv3+ (Chen et al. 2018; https://arxiv.org/abs/1802.0261)