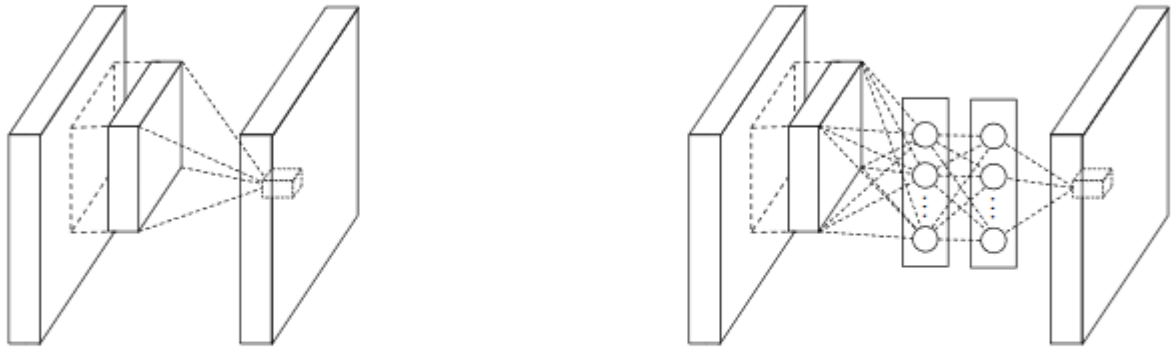
More Convolutions

Network in network

- https://arxiv.org/abs/1312.4400

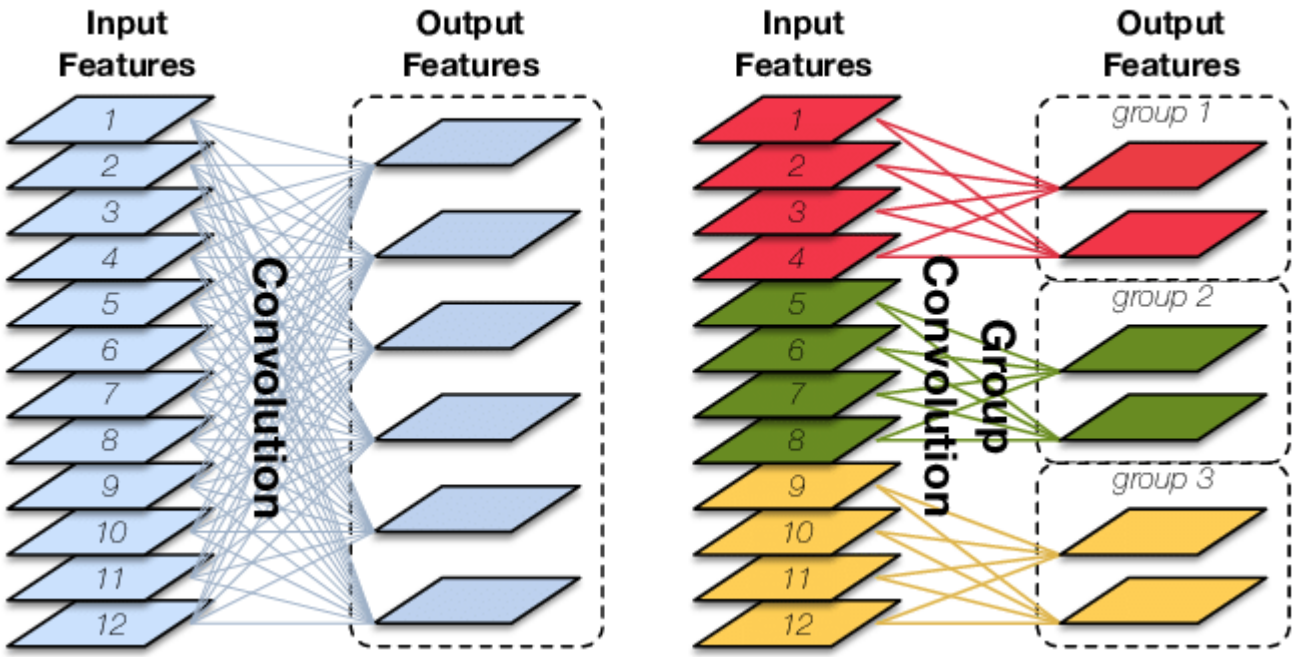


(a) Linear convolution layer

(b) Mlpconv layer

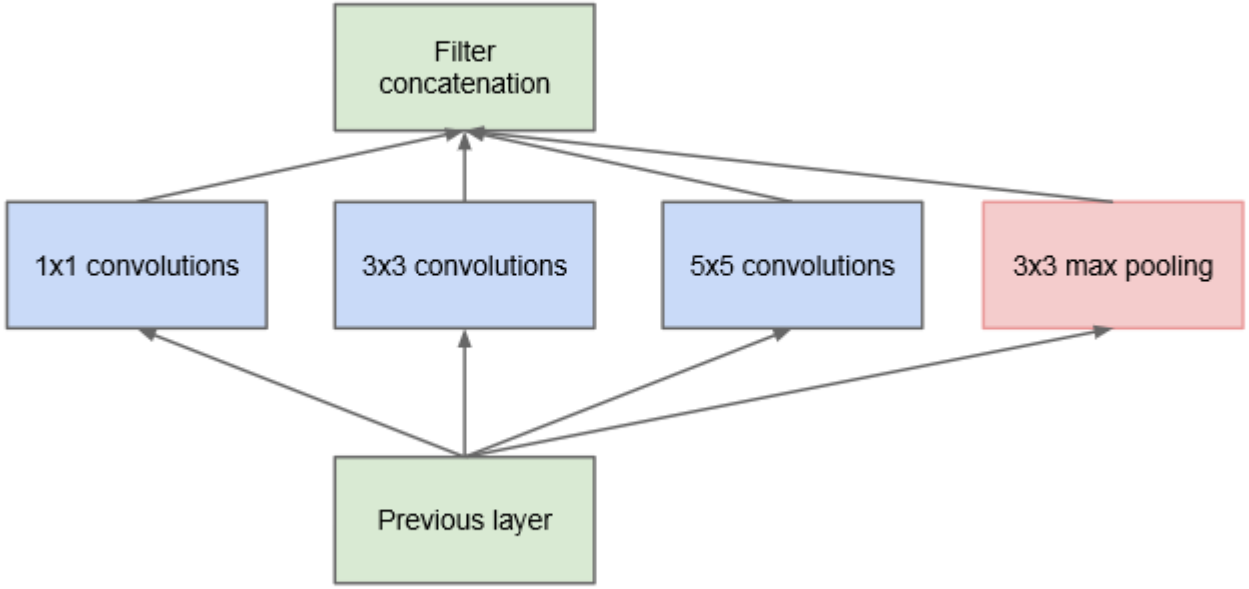
Figure 1: Comparison of linear convolution layer and mlpconv layer. The linear convolution layer includes a linear filter while the mlpconv layer includes a micro network (we choose the multilayer perceptron in this paper). Both layers map the local receptive field to a confidence value of the latent concept.

Group Conv


Input Features
Output Features
1
2
3
4
5
6
7
8
9
10
11
12
Convolution
Input Features
Output Features
group 1
group 2
group 3
Group Convolution

Inception

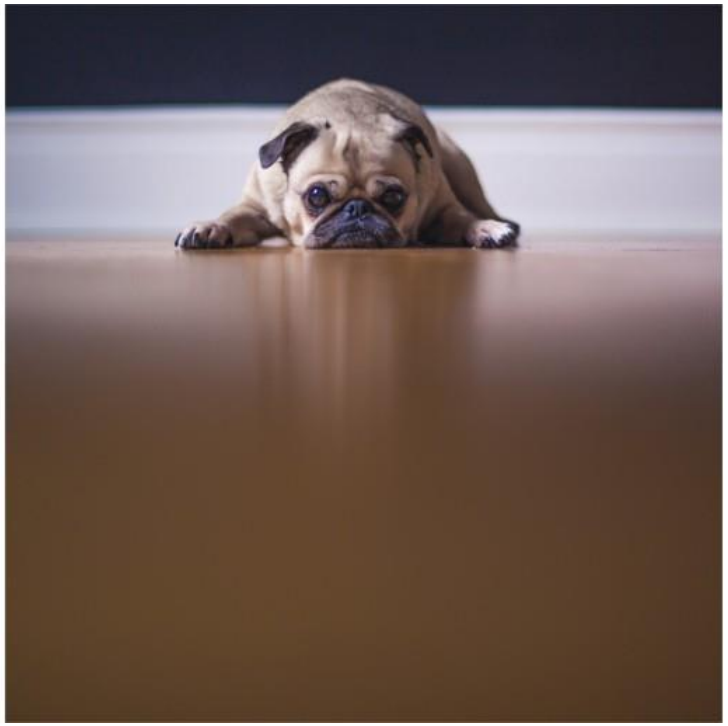
- https://arxiv.org/abs/1409.4842

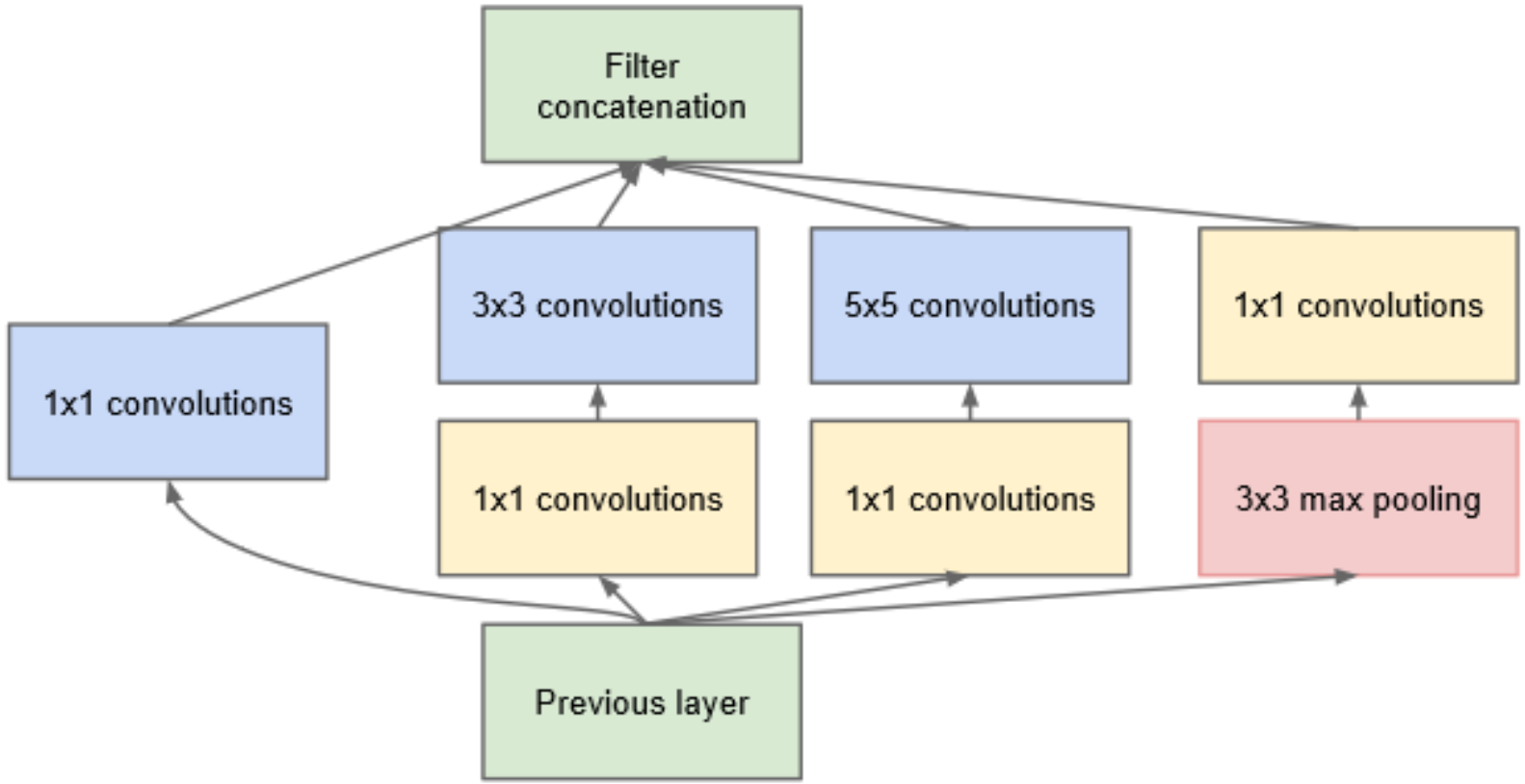


(a) Inception module, naïve version

https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202







(b) Inception module with dimension reductions

Inception v2,v3

- https://arxiv.org/abs/1512.00567

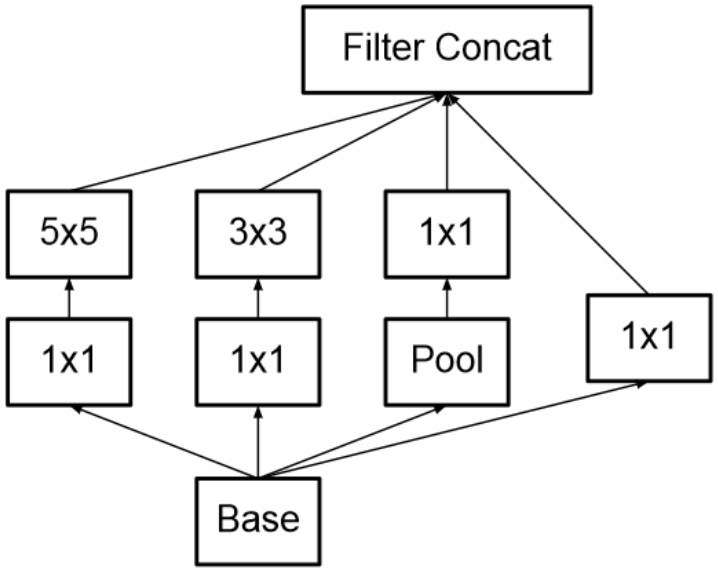


Figure 4. Original Inception module as described in [20].

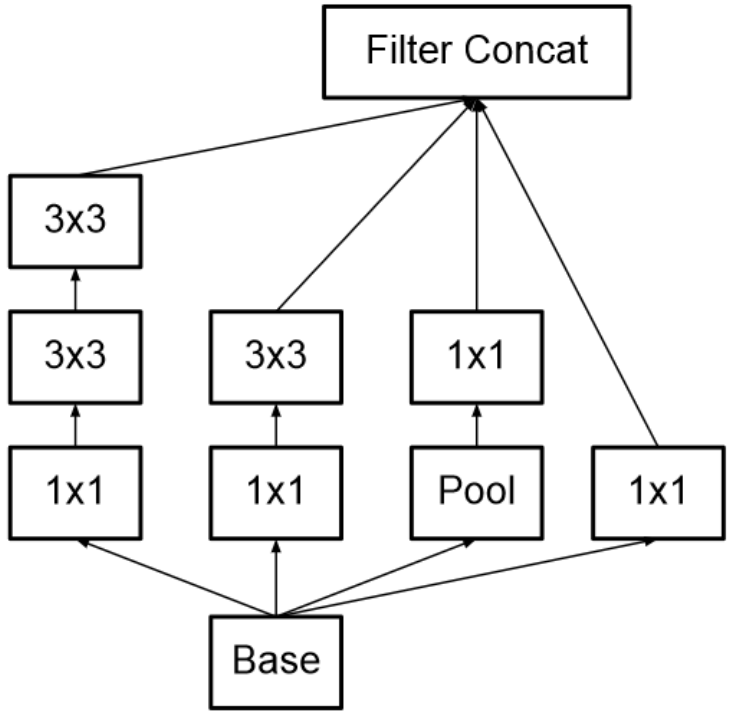


Figure 5. Inception modules where each 5×5 convolution is replaced by two 3×3 convolution, as suggested by principle 3 of Section 2.

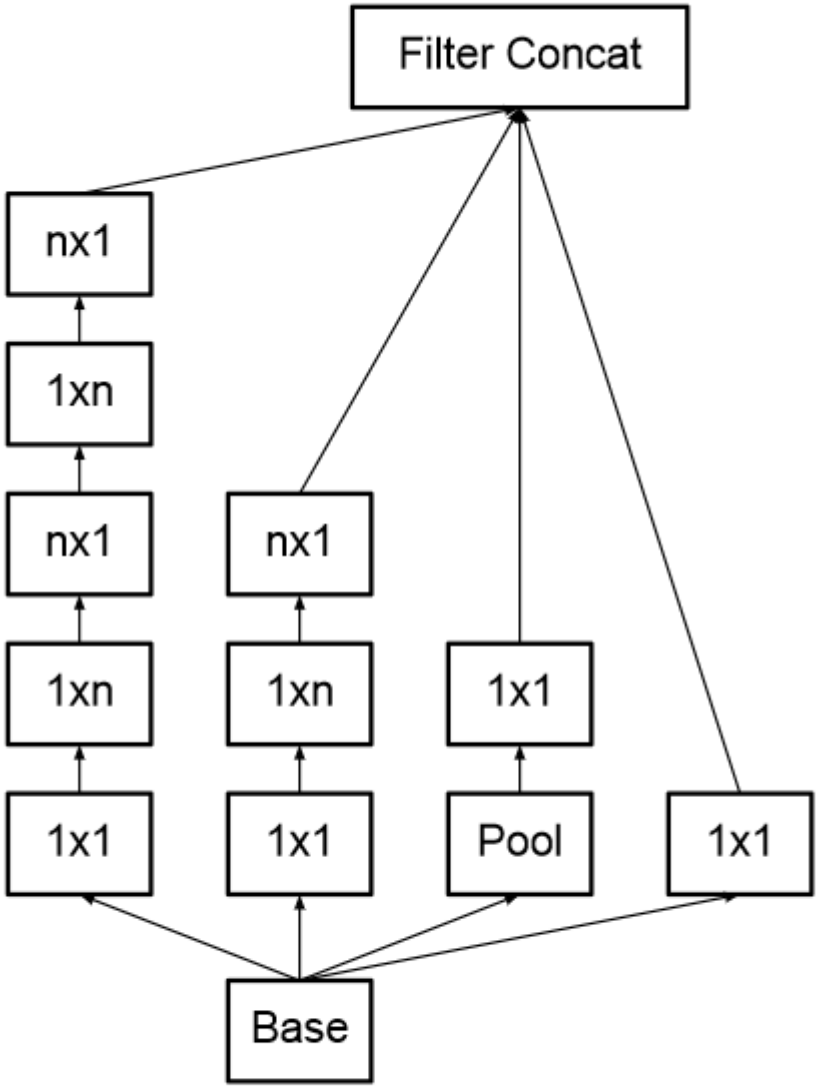


Figure 6. Inception modules after the factorization of the $n \times n$ convolutions. In our proposed architecture, we chose $n = 7$ for the 17×17 grid. (The filter sizes are picked using principle 3)

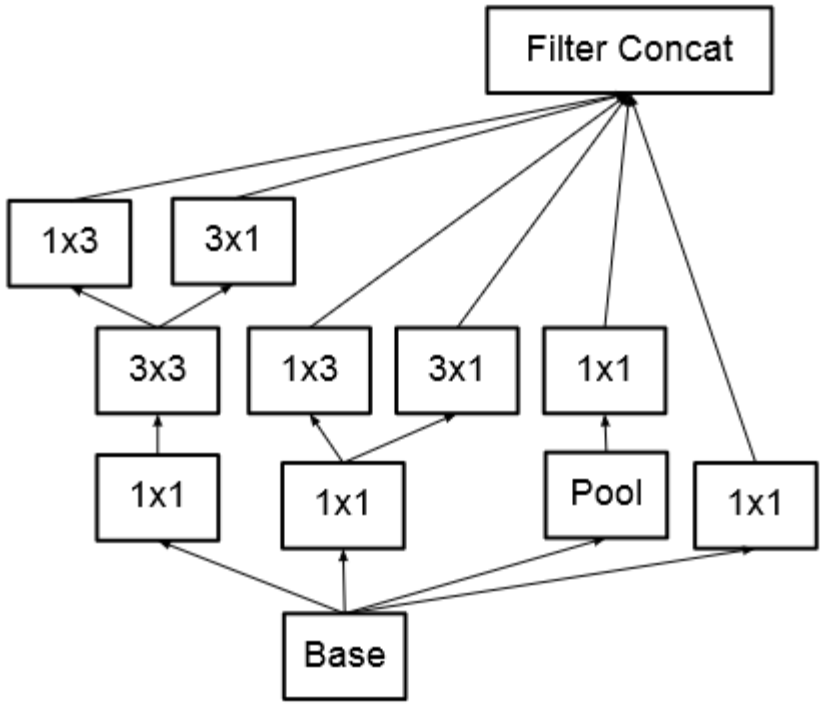


Figure 7. Inception modules with expanded the filter bank outputs. This architecture is used on the coarsest (8×8) grids to promote high dimensional representations, as suggested by principle 2 of Section 2. We are using this solution only on the coarsest grid, since that is the place where producing 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.

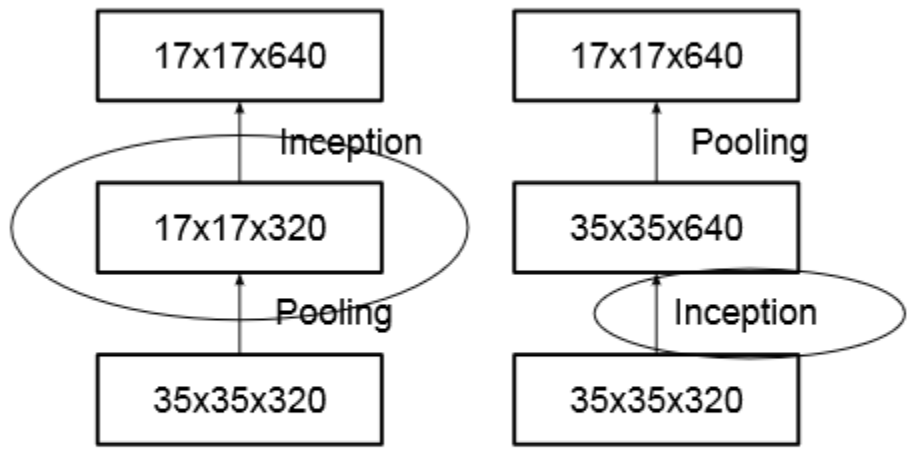


Figure 9. Two alternative ways of reducing the grid size. The solution on the left violates the principle 1 of not introducing an representational bottleneck from Section 2. The version on the right is 3 times more expensive computationally.

ResNet

- https://arxiv.org/abs/1512.03385

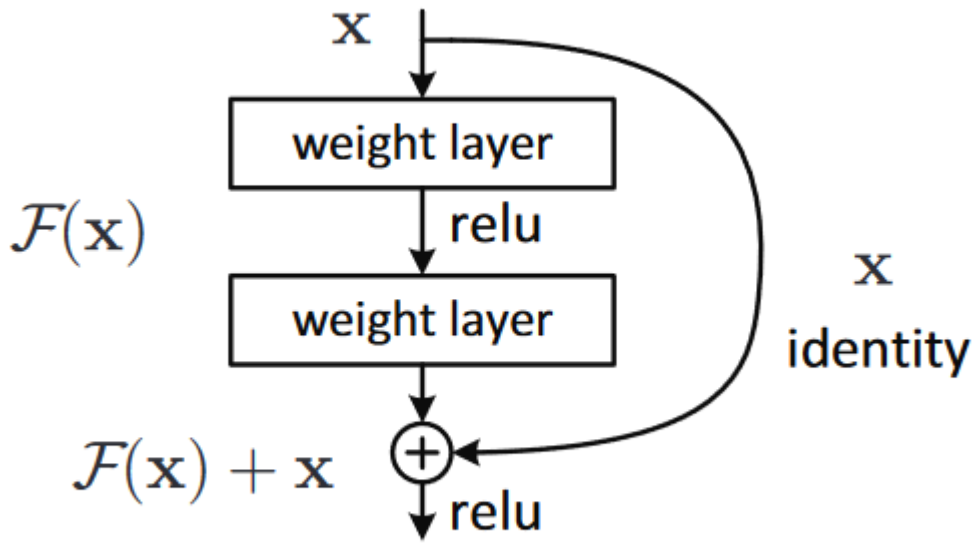


Figure 2. Residual learning: a building block.

Inception-v4

- https://arxiv.org/pdf/1602.07261.pdf

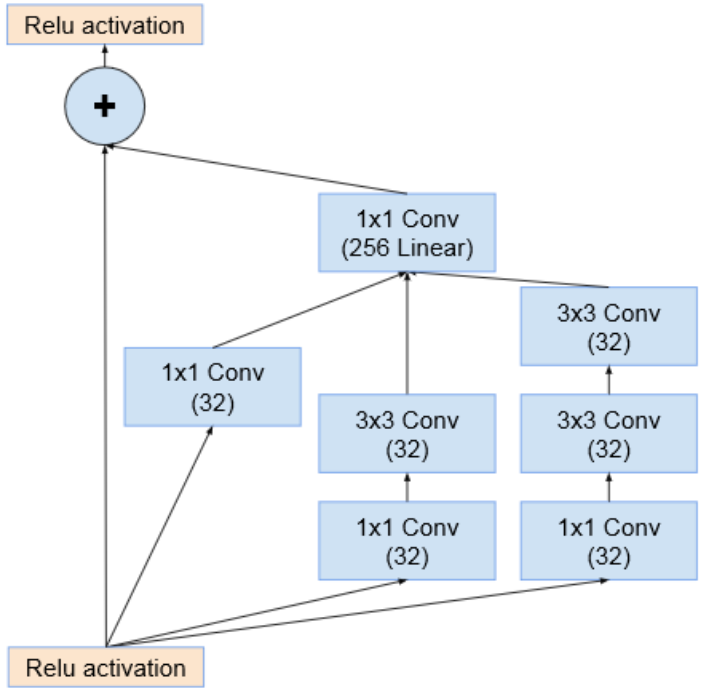


Figure 10. The schema for 35 × 35 grid (Inception-ResNet-A) module of Inception-ResNet-v1 network.

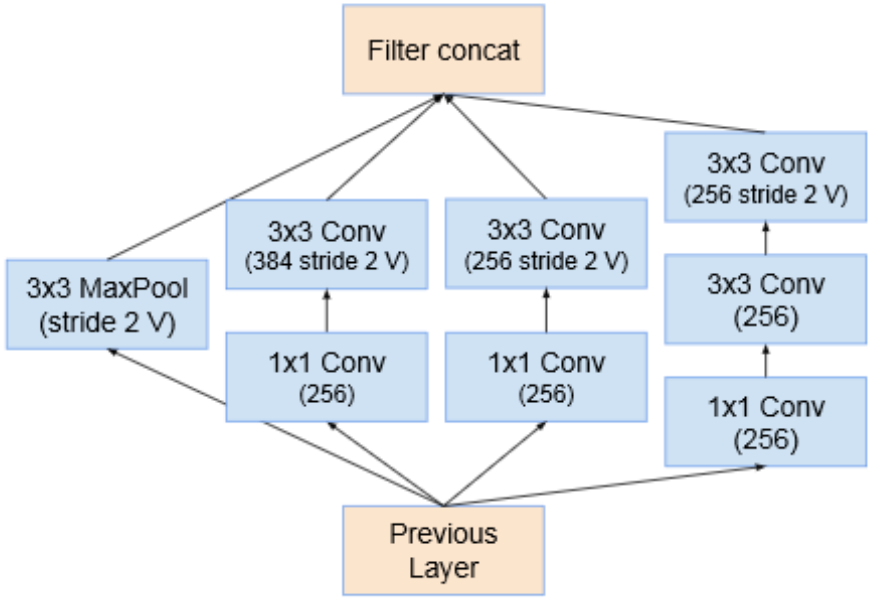
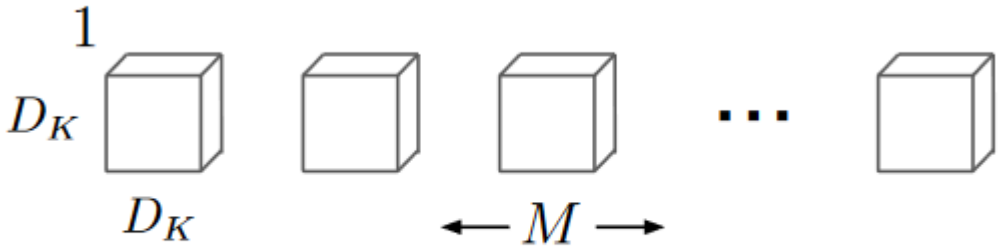


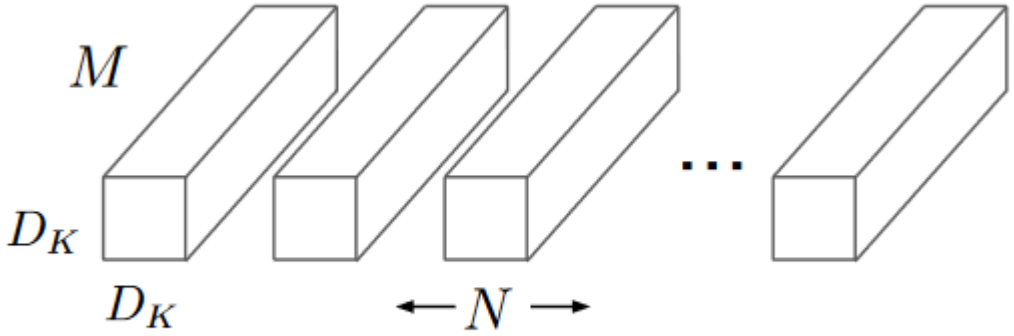
Figure 12. "Reduction-B" 17 × 17 to 8 × 8 grid-reduction module. This module used by the smaller Inception-ResNet-v1 network in Figure 15.

MobileNets

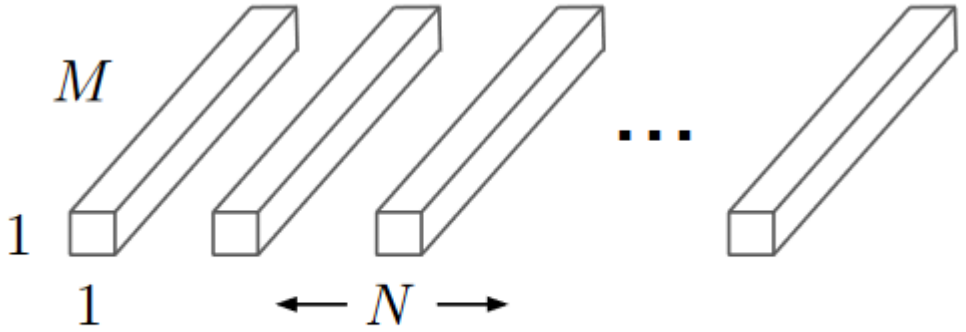
- https://arxiv.org/abs/1704.04861



(b) Depthwise Convolutional Filters



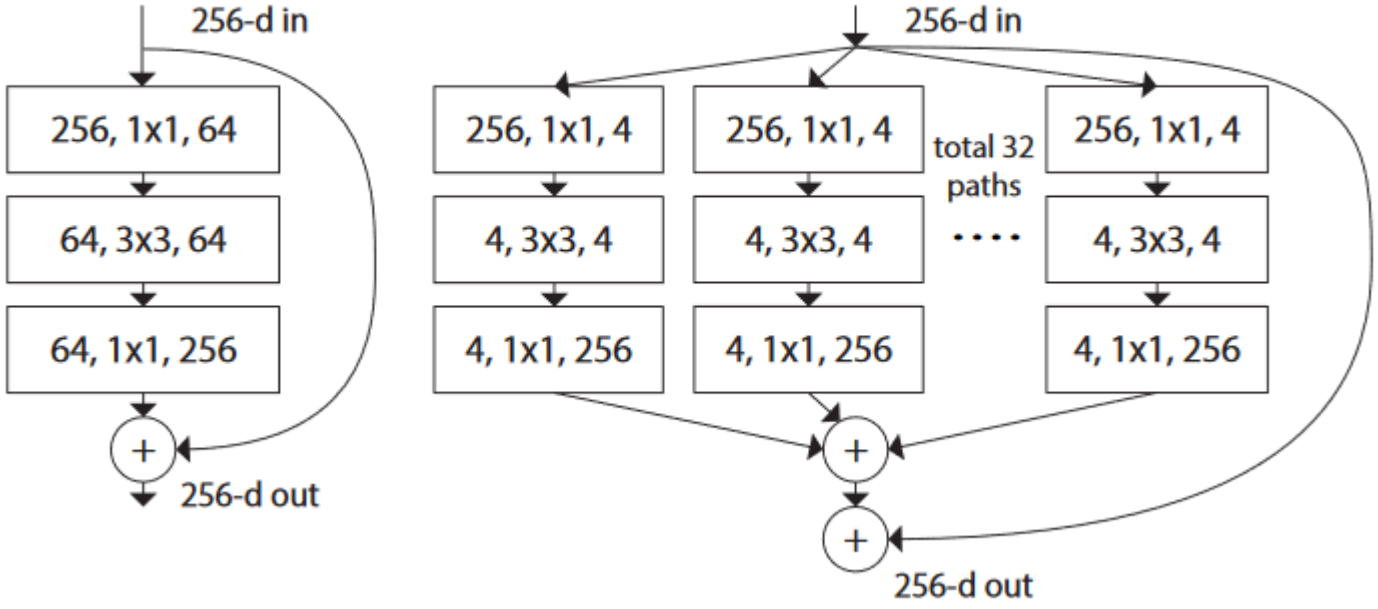
(a) Standard Convolution Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

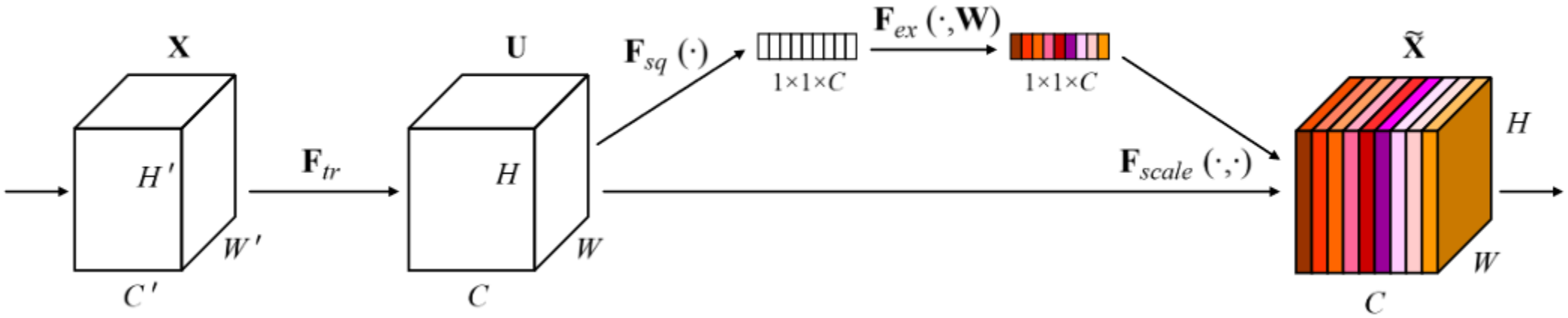
ResNext

- https://arxiv.org/pdf/1611.05431.pdf



Squeeze & Excitation

- https://arxiv.org/abs/1709.01507



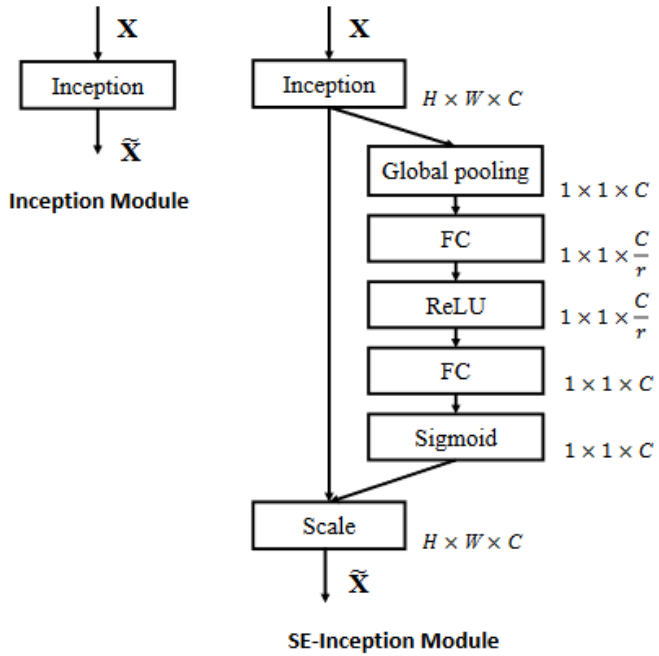


Fig. 2. The schema of the original Inception module (left) and the SE-Inception module (right).

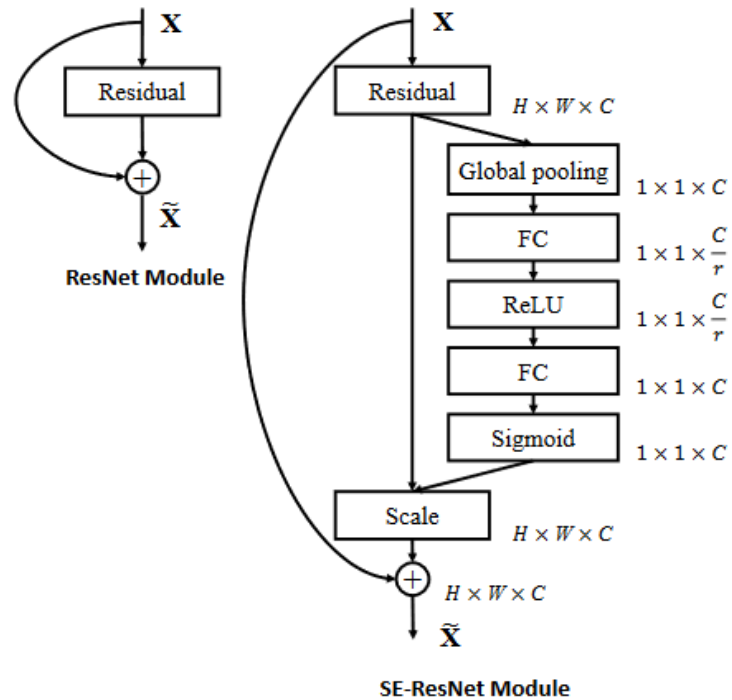


Fig. 3. The schema of the original Residual module (left) and the SE-ResNet module (right).

DenseNet

https://arxiv.org/abs/1608.06993

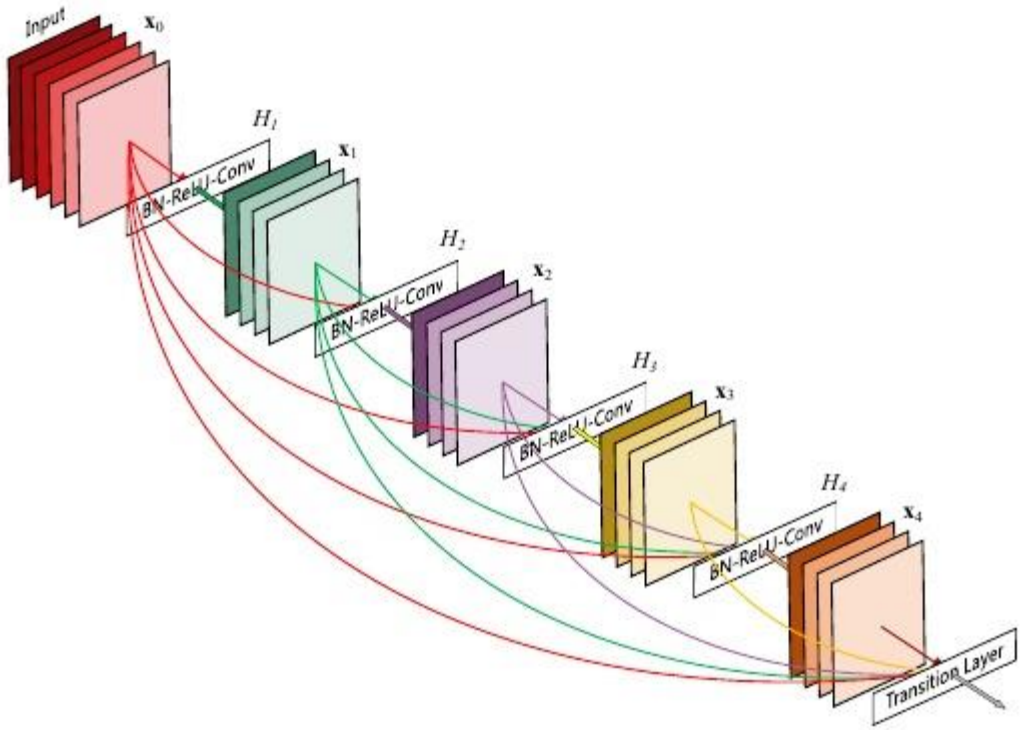


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature maps as input.

Shuffle Net

- https://zpascal.net/cvpr2018/Zhang_ShuffleNet_An_Extremely_CVPR_2018_paper.pdf

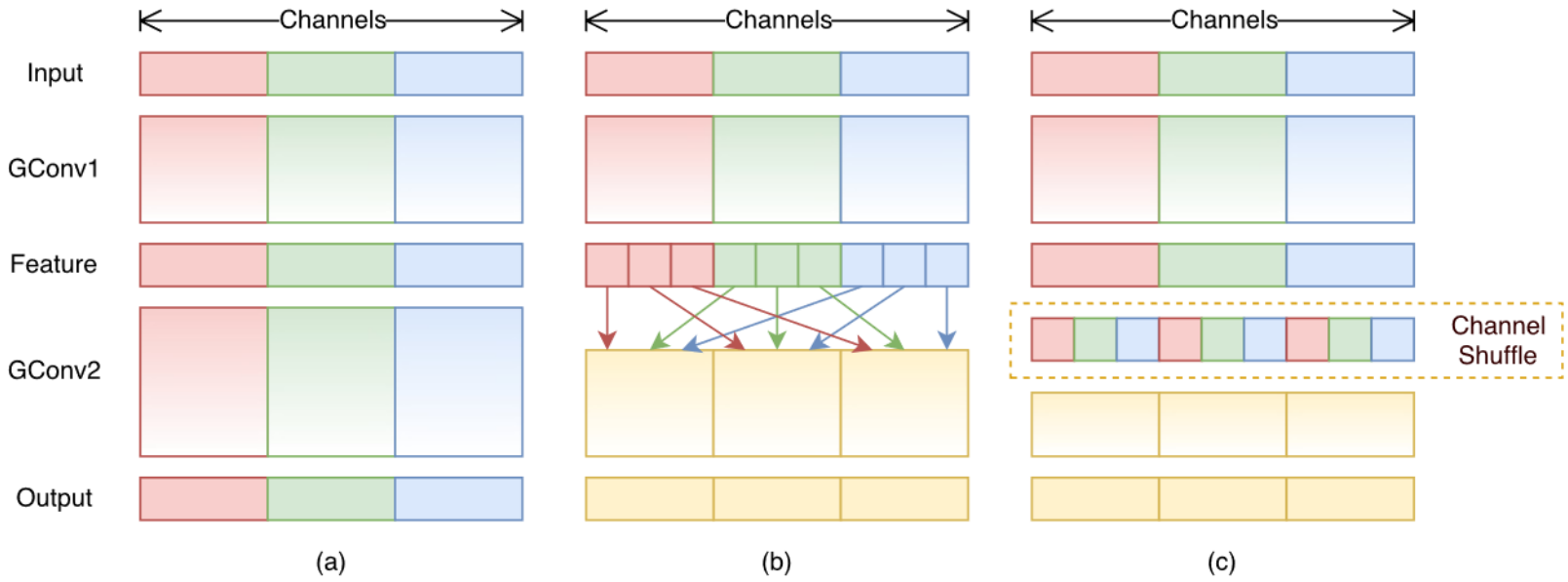


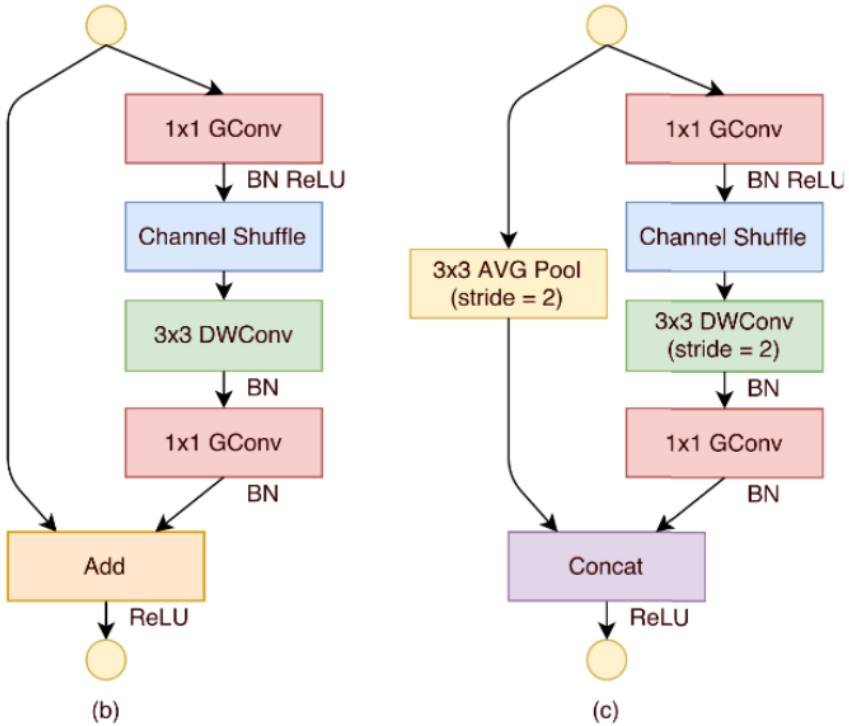
Figure 1. Channel shuffle with two stacked group convolutions. GConv stands for group convolution. a) two stacked convolution layers with the same number of groups. Each output channel only relates to the input channels within the group. No cross talk; b) input and output channels are fully related when GConv2 takes data from different groups after GConv1; c) an equivalent implementation to b) using channel shuffle.

MobileNets V2

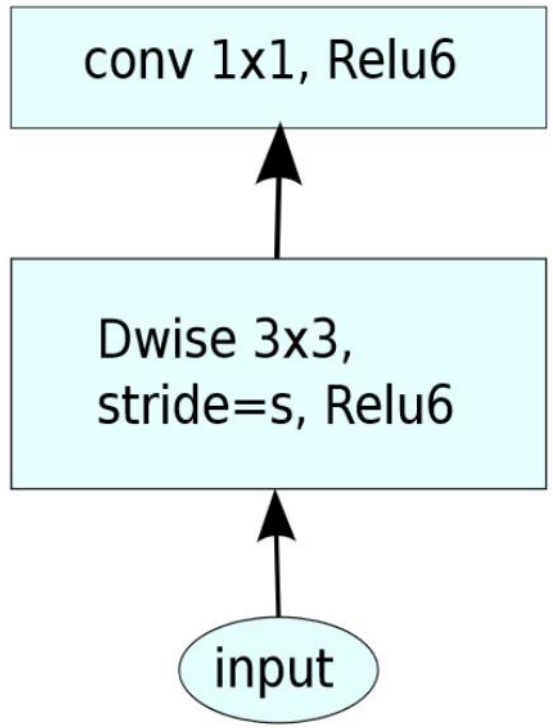
- https://arxiv.org/pdf/1801.04381.pdf
-

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

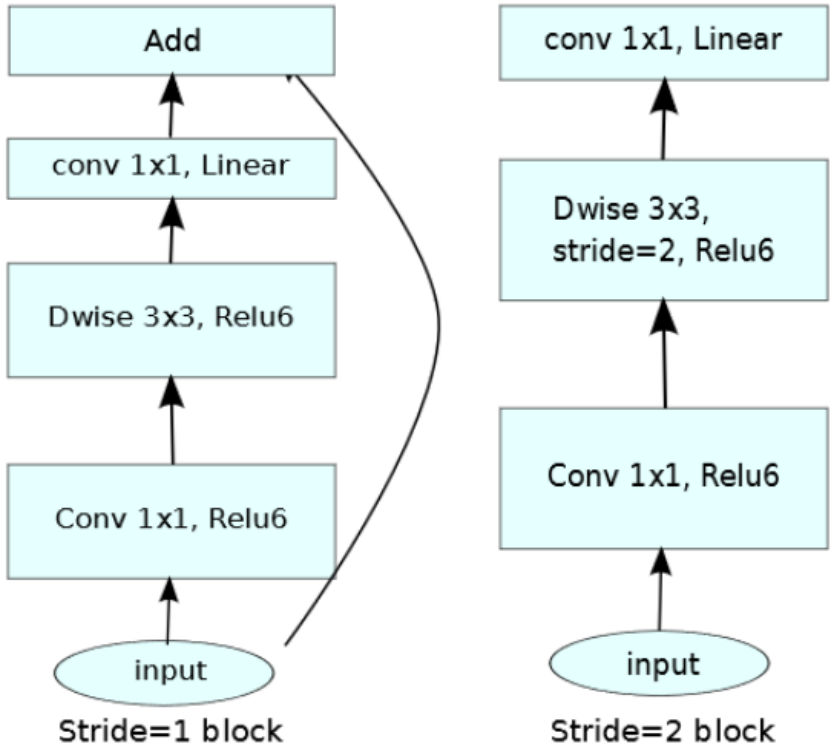
Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s, and expansion factor t.



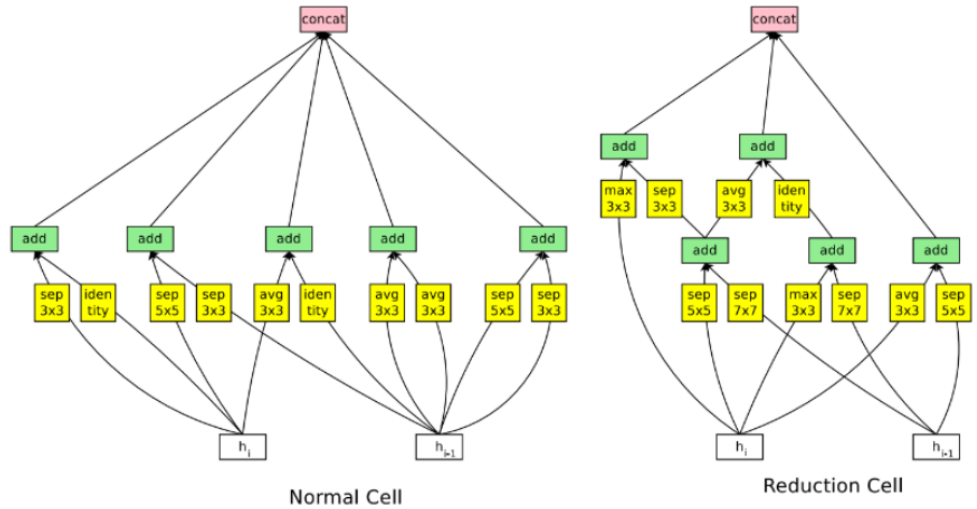
(c) ShuffleNet [20]



(b) MobileNet[27]



(d) Mobilenet V2



(a) NasNet[23]

Dilated Conv

- https://arxiv.org/abs/1511.07122

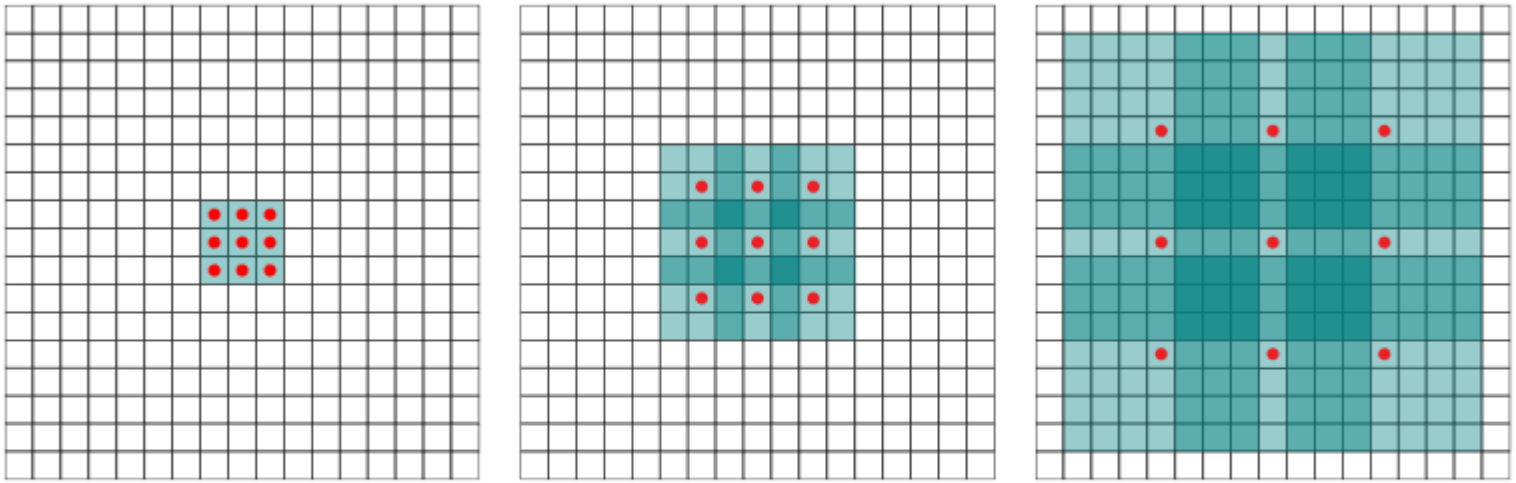


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}).$$

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}).$$

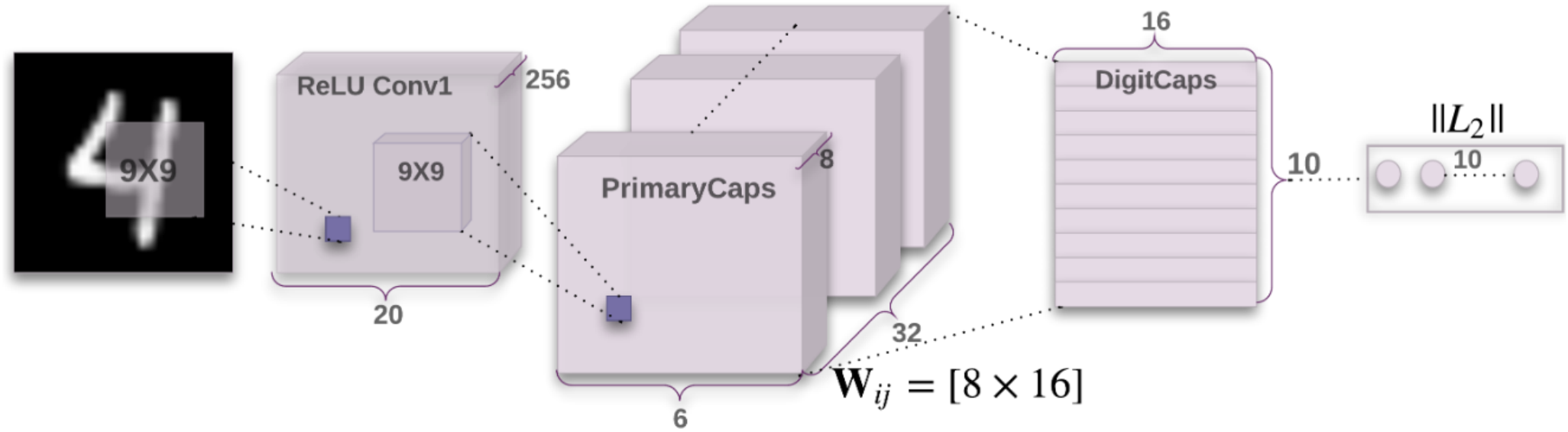
CapsNets

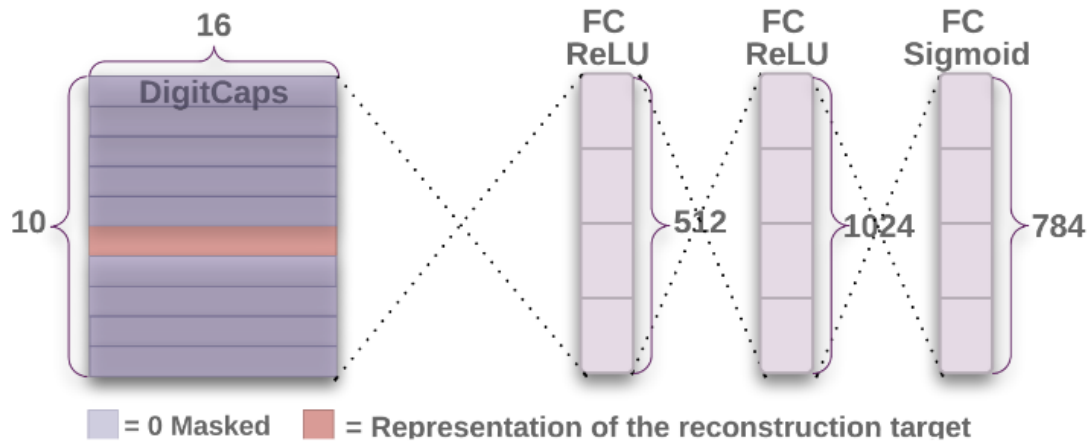
- https://arxiv.org/abs/1710.09829
- What's wrong with cnn? https://www.youtube.com/watch?v=rTawFwUvnLE
- https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks
-

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

Procedure 1 Routing algorithm.

procedure ROUTING($\hat{\boldsymbol{u}}_{j|i}$, r, l)
  for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  for r iterations **do**
    for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ `softmax` computes Eq. 3
    for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$
    for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ `squash` computes Eq. 1
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
  return $\mathbf{v}_j$


9X9
ReLU Conv1
256
9X9
20
PrimaryCaps
8
32
6
16
DigitCaps
10
$\|L_2\|$
10
$\mathbf{W}_{ij} = [8 \times 16]$


16
DigitCaps
10
FC
ReLU
512
FC
ReLU
1024
FC
Sigmoid
784
= 0 Masked
= Representation of the reconstruction target