Nizhny Novgorod State University

Institute of Information Technologies, Mathematics and Mechanics

Department of Computer software and supercomputer technologies

Educational course
«Modern methods and technologies
of deep learning in computer vision»

Image classification with a large number of categories using deep learning

Supported by Intel

Getmanskaya Alexandra, Kustikova Valentina

Content

- Goals
- Image classification problem statement
- ImageNet Large Scale Visual Recognition Challenge and the ImageNet dataset
- Deep models for image classification on the ImageNet dataset
- Comparison of classification accuracy and complexity of deep models on the ImageNet dataset
- Conclusion

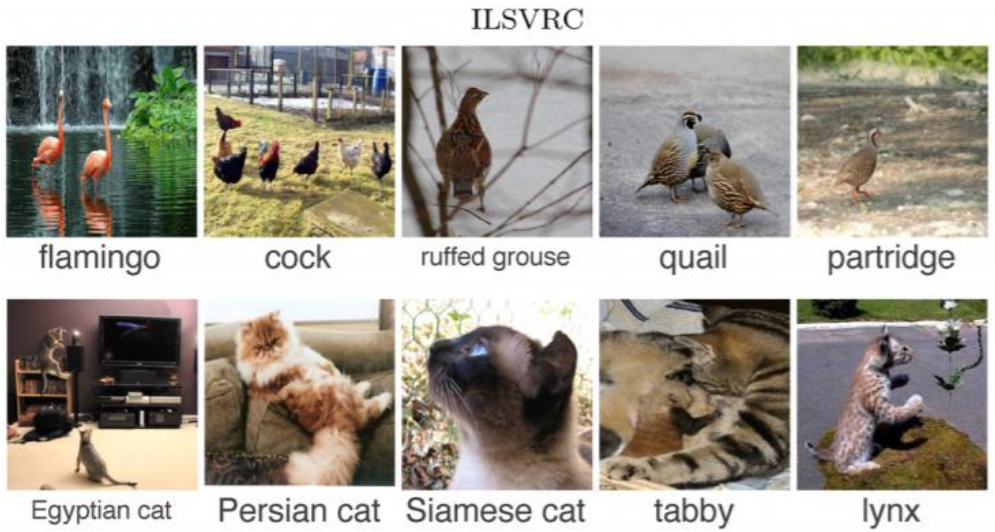
Goals

- ***The goal*** is to study deep neural networks for solving image classification problem

IMAGE CLASSIFICATION PROBLEM STATEMENT

Problem statement (1)

- The problem of image classification is to match the image with the class of objects represented in the image
- Examples of images and corresponding classes:



* Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A.C., Fei-Fei L. ImageNet Large Scale Visual Recognition Challenge // International Journal of Computer Vision, 2015.

Problem statement (2)

- The original image is represented by a set of pixel intensities $I = \left(I_{ij}^{k}\right)_{\substack{0 \le i < w \\ 0 \le j < h \\ 0 \le k < 3}}$, where w and h are image width and height, k is a number of color channels of the image
- $C = \{0, 1, \ldots, N-1\}$ is a set of object classes in the image, the set of class identifiers uniquely corresponds to the set of class names
- The problem of image classification is to match each image with the class to which the image belongs to

$$\varphi : I \to C$$

IMAGENET LARGE SCALE VISUAL RECOGNITION CHALLENGE AND THE IMAGENET DATASET

ImageNet Large Scale Visual Recognition Challenge

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a competition for image classification with a large number of categories and object detection
- From 2010 to 2017, it is based on [http://www.image-net.org], since 2017 it moved to the Kaggle platform
- ImageNet is a public dataset provided as a part of the ILSVRC; it contains 14 197 122 images

* Russakovsky O., et al. ImageNet Large Scale Visual Recognition Challenge. – 2015. – [https://arxiv.org/pdf/1409.0575.pdf].

ImageNet

- The dataset consists of 14 197 122 images belonging to 21 841 categories from the WordNet hierarchy*
- The hierarchy contains 27 high-level categories
- 1 034 908 images contain groundtruth for the problem of object detection (bounding boxes), this groundtruth is also used for the image classification problem
- Image resolution varies, average resolution is 400x350 pixels
- Images are collected from various sources, the owners of the dataset do not have copyrights on the images

* Jia D., Dong W., Socher R., Li L.-J., Li K., Li F.-F. ImageNet: A large-scale hierarchical image database // In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2009. – P. 248-255. – [https://ieeexplore.ieee.org/document/5206848].

ImageNet dataset for image classification of ILSVRC 2012

- 1 000 image categories
- The minimum image resolution is 75x56 pixels
- The maximum image resolution is 4288x2848 pixels
- The size of the training dataset is 1 200 000 images
- The size of the validation dataset is 50 000 images
- The size of test dataset is 150 000 images

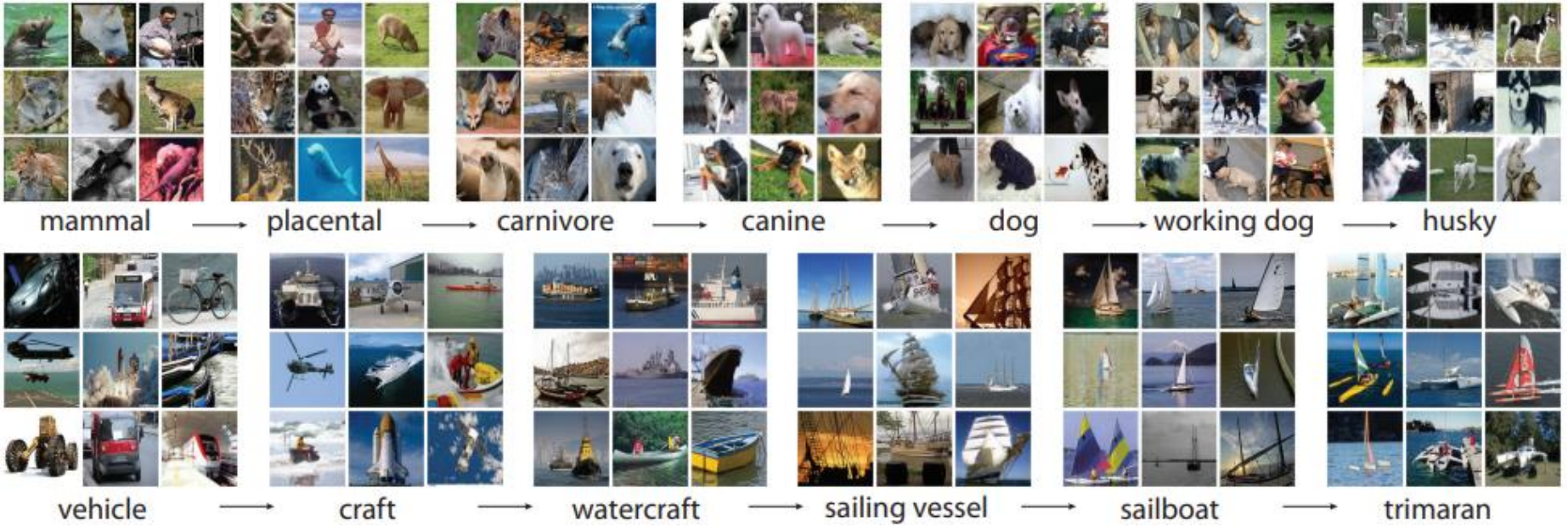
WordNet class hierarchy (1)

- WordNet* is a large lexical database of English words
- The basic relationship between words in WordNet is ***synonymy***
- ***Synonyms*** are words that are close in meaning, interchangeable in many contexts
- Synonyms are grouped into ***unordered sets*** (synset)
- Synonym groups are related by the following relationships:
  - ***Hyperonymy (hyponymy)*** is a relationship of the general and the private (for example, a bed is a furniture)
  - ***Meronymy (partonymy)*** is the relationship of objects and their parts (for example, “engine” is a meronym in relation to the term “car”)

* WordNet. A Lexical Database for English [https://wordnet.princeton.edu].

WordNet class hierarchy (2)

- WordNet* contains ~80 000 nouns
- The goal of developing the ImageNet dataset is to collect 500-1000 images for each set of synonyms



* WordNet. A Lexical Database for English [https://wordnet.princeton.edu].

** Ye T. Visual Object Detection from Lifelogs using Visual Non-lifelog Data. – 2018. – [https://www.researchgate.net/publication/324797660_Visual_Object_Detection_from_Lifelogs_using_Visual_Non-lifelog_Data].

DEEP MODELS FOR IMAGE CLASSIFICATION ON THE IMAGENET DATASET

Deep models (1)

Increasing network depth

- ***AlexNet (2012)***
  - Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in neural information processing systems. – 2012. – [http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf].
- ***OverFeat (2013)***
  - Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. – 2013. – [https://arxiv.org/pdf/1312.6229.pdf].
- ***VGG-16, VGG-19, GoogLeNet (2014)***
  - Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. – 2014. – [https://arxiv.org/pdf/1409.1556.pdf].
  - Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going Deeper with Convolutions. – 2014. – [https://arxiv.org/pdf/1409.4842.pdf].

Deep models (2)

Solving model degradation problem

- ***ResNet-*(50, 101, 152), Inception-v*(2,3) (2015)***
  - He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. – 2015. – [https://arxiv.org/pdf/1512.03385.pdf].
  - Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. – 2015. – [https://arxiv.org/pdf/1502.03167.pdf].
  - Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf], [https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf].
- ***DenseNet-* (121, 169, 201, 264), Xception (2016)***
  - Huang G., Liu Z., Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. – 2016. – [https://arxiv.org/pdf/1608.06993.pdf].
  - Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Decreasing number of model parameters

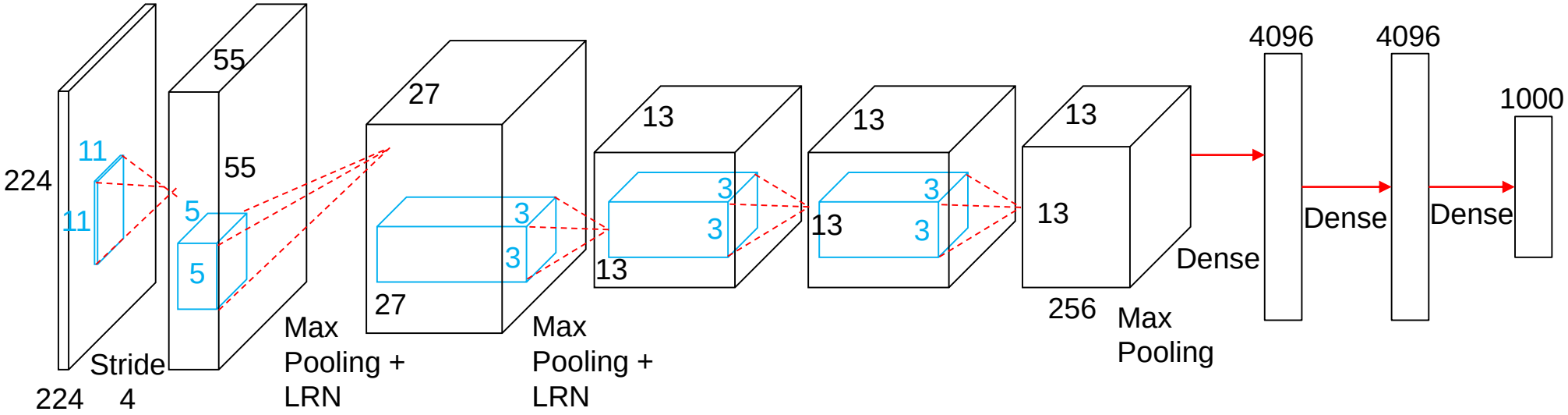
Deep models (3)

Decreasing model complexity

- ***MobileNet, ResNeXT-* (2017)***
  - Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – 2017. – [https://arxiv.org/pdf/1704.04861.pdf].
  - Xie S., Girshick R., Dollar P., Tu Z., He K. Aggregated Residual Transformations for Deep Neural Networks. – 2017. – [https://arxiv.org/pdf/1611.05431v2.pdf], [https://ieeexplore.ieee.org/document/8100117].
- ***MobileNetV2 (2018)***
  - Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. – 2018. – [https://arxiv.org/pdf/1801.04381.pdf], [https://ieeexplore.ieee.org/document/8578572].
- ***EfficientNet-* (B0,…,B7) (2019)***
  - Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. – 2019. – [https://arxiv.org/pdf/1905.11946.pdf].

AlexNet (1)

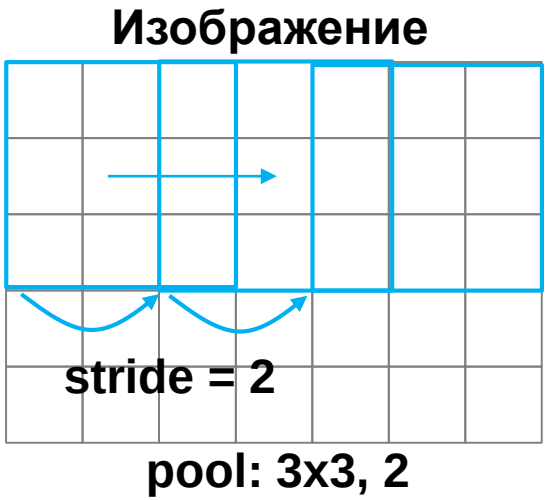
- AlexNet is the first deep convolutional neural network
- Network developers won the LSVRC 2012 image classification contest on the ImageNet dataset
- The classification error is 15.3% compared with the error of 25.7% obtained a year earlier



* Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in neural information processing systems. – 2012. – [http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf].

AlexNet (2)

- Model features:
  - The network input is a three-channel image of 224x224 pixels
  - A rectified linear unit (ReLU) is used as an activation function
  - Using dropout layers (nulling the outputs of neurons with the probability of 0.5)
  - Using overlapping pooling layers
  - Local Response Normalization (LRN) is normalization of output values by dimension corresponding to the depth of the output feature map



AlexNet (3)

- The model complexity:
  - The network contains 62.3 million parameters
  - Forward pass requires ~1 billion operations
  - Convolutional layers, which account for 6% of all parameters, perform 95% of the calculations

- The training features:
  - High training speed through the use of the ReLU activation function
  - Data augmentation by shifting and mirroring images
  - Training on two GPUs

OverFeat (1)

- OverFeat is a deep model based on the AlexNet model
- Overfeat simultaneously solves the problems of image classification, object localization and detection within the ILSVRC contest
- Localization involves determining the position of one object (i.e. constructing a bounding box)
- Differences between object detection and localization:
  - Any number of objects
  - Images contain small objects
  - In the absence of objects, the prediction of the class to which the background belongs is assumed
  - Different quality metrics for detection and localization

* Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. – 2013. – [https://arxiv.org/pdf/1312.6229.pdf].

OverFeat (2)

- The authors propose two models that are very similar to the AlexNet model:
  - Fast model

Layer	1	2	3	4	5	6	7	Output (8)
Stage	conv+max	conv+max	conv	conv	conv+max	full	full	full
K channels	96	256	512	1024	1024	3072	4096	1000
Filter size	11x11	5x5	3x3	3x3	3x3	–	–	–
Convolution stride	4x4	1x1	1x1	1x1	1x1	–	–	–
Pooling size	2x2	2x2	–	–	2x2	–	–	–
Pooling stride	2x2	2x2	–	–	2x2	–	–	–
Zero-Padding size	–	–	1x1x1x1	1x1x1x1	1x1x1x1	–	–	–
Spatial input size	231x231	24x24	12x12	12x12	12x12	6x6	1x1	1x1

AlexNet without overlapping pooling and local response normalization

OverFeat (3)

- The authors propose two models that are very similar to the AlexNet model:
  - Accurate model

Layer	1	2	3	4	5	6	7	8	Output (9)
Stage	conv+ max	conv+ max	conv	conv	conv	conv+max	full	full	full
K channels	96	256	512	512	1024	1024	4096	4096	1000
Filter size	7x7	7x7	3x3	3x3	3x3	3x3	–	–	–
Convolution stride	2x2	1x1	1x1	1x1	1x1	1x1	–	–	–
Pooling size	3x3	2x2	–	–	–	3x3	–	–	–
Pooling stride	3x3	2x2	–	–	–	3x3	–	–	–
Zero-Padding size	–	–	1x1x1x1	1x1x1x1	1x1x1x1	1x1x1x1	–	–	–
Spatial input size	221x221	36x36	15x15	15x15	15x15	15x15	5x5	1x1	1x1

The color indicates differences from the fast model

OverFeat (4)

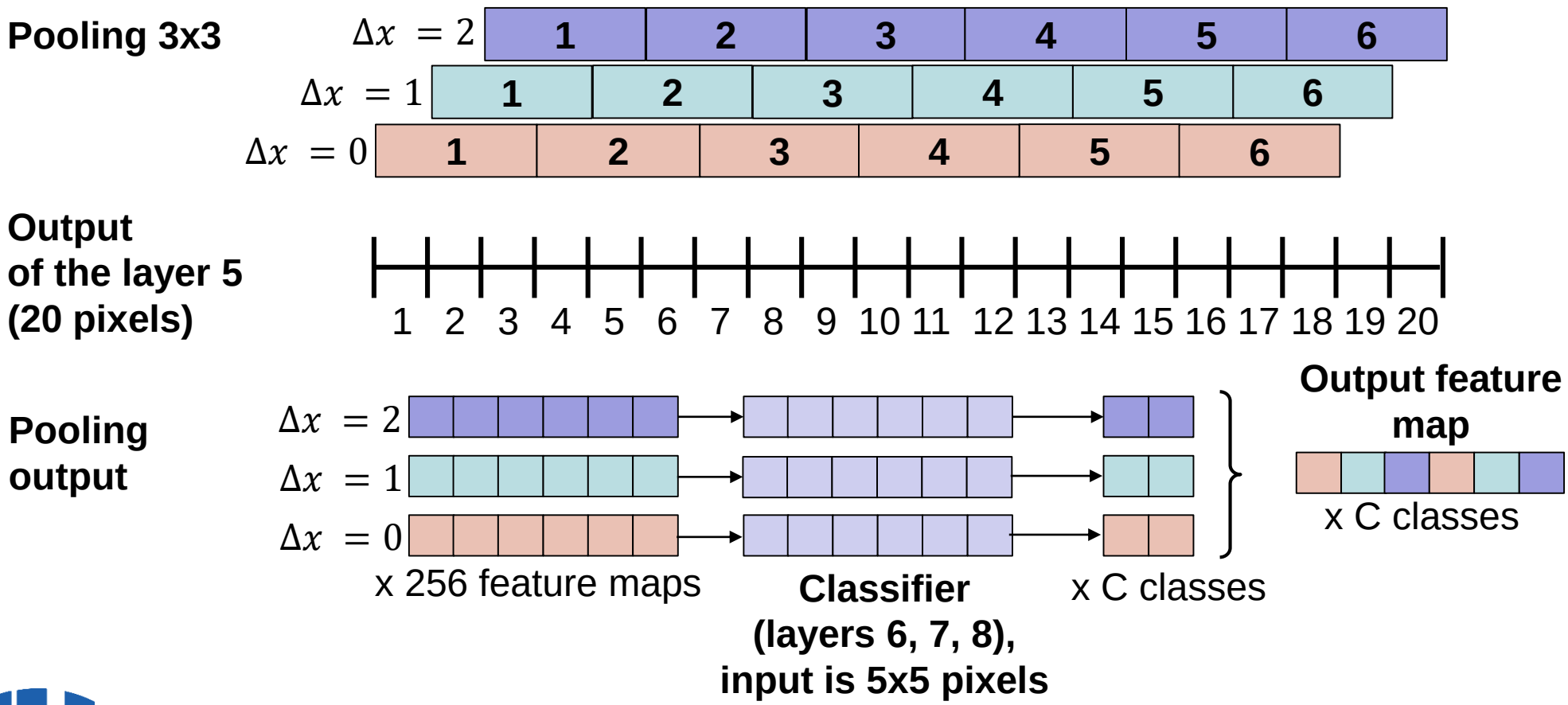
- Differences from the AlexNet model:
  - Absence of overlapping pooling (replacing the kernel size from 3x3 to 2x2)
  - Absence of local response normalization on the first and third layers
  - Application of multiscale classification
    - The problem is different scales of objects
    - The idea is to classify different scales of the image and make an integral decision
    - 6 different scales of the input image are used, for which feature maps are constructed (layer 5 output). The final feature maps are combined and transmitted to the input of the classifier, which makes the final decision on whether the image belongs to the class

OverFeat (5)

- Combining feature maps obtained at different image scales:
  - Extract the output feature maps from the fifth layer of the model
  - Apply max pooling without overlapping with the kernel size 3x3 for each obtained feature map. The operation is performed 9 times (3x3) for all kinds of horizontal and vertical displacements of the kernel $\Delta x, \Delta y \in \{0,1,2\}$ resulting in 9 feature maps
  - Each feature map is an input of the classifier (layers 6, 7, 8). Sizes of feature maps are different, and the classifier input size is fixed, therefore, the classifier is used in the manner of a “sliding” window
  - The shape of the output feature maps is reduced to a three-dimensional tensor (two spatial dimensions and the number of classes)

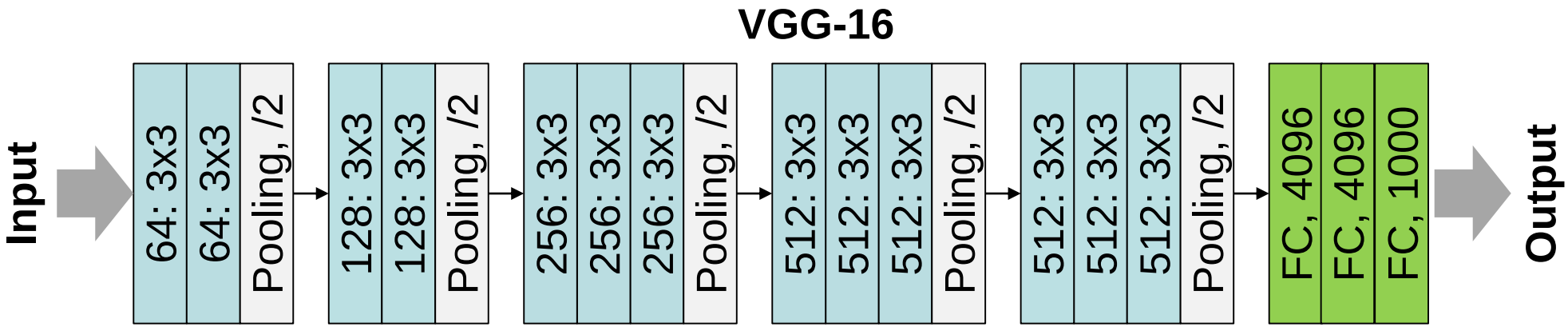
OverFeat (6)

- Combining feature maps obtained at different image scales:



VGG-16, 19 (1)

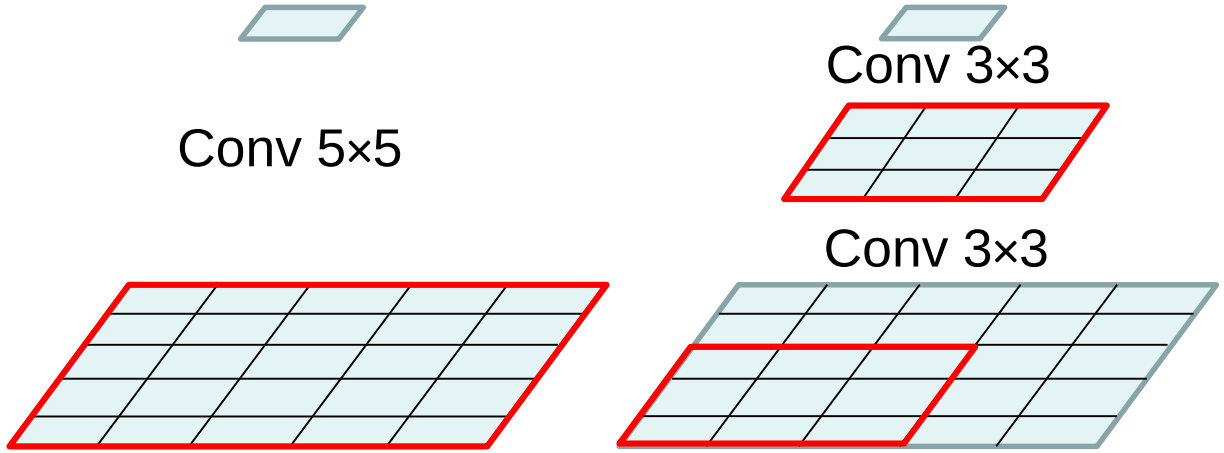
- VGG- * is an improvement of the AlexNet model, the fundamental difference is that the large convolutional kernels (11 and 5) are replaced by 3x3 convolutions



* Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. – 2014. – [https://arxiv.org/pdf/1409.1556.pdf].

VGG-16, 19 (2)

- A convolution with a 5x5 kernel can be replaced by two consecutive convolutions with 3x3 kernels
- In this case, a network contains a smaller number of parameters (25 vs. 18), but with the same size of the input and receptive field



- VGG-19 (16 convolutional layers) contains more convolutional layers than VGG-16. The number of blocks containing the sequence of convolutions and pooling are the same

ResNet-50, 101, 152 (1)

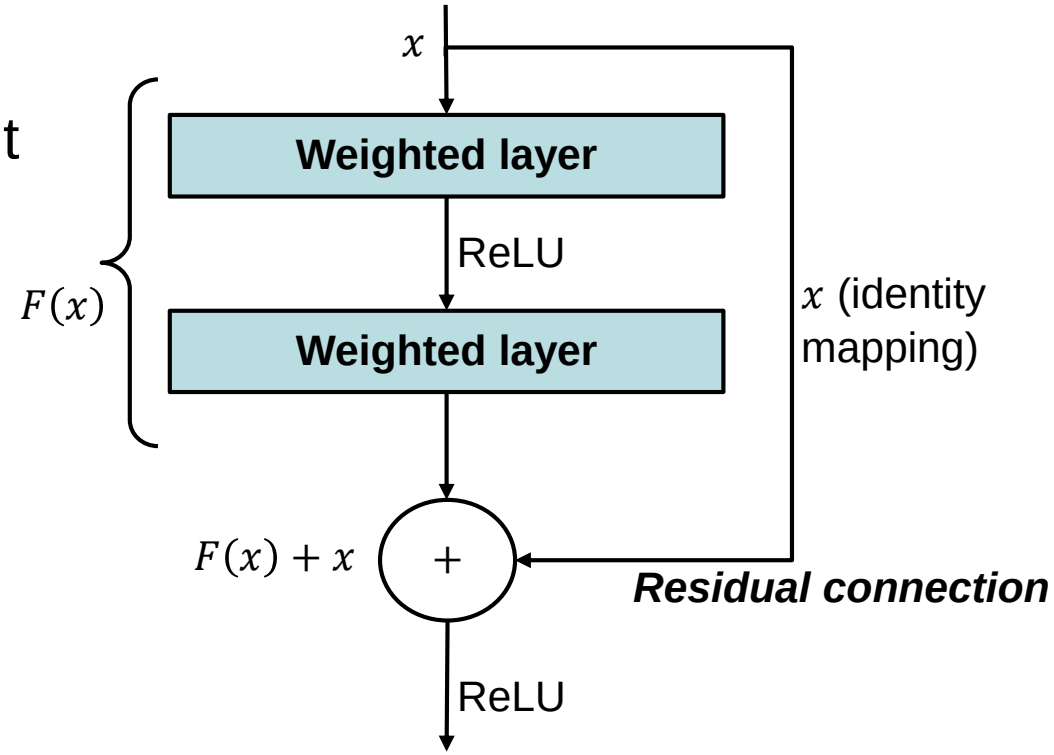
- By the beginning of 2015, the general trend in the development of deep models is to increase the number of convolutional layers
- With increasing network depth, the accuracy saturates and then quickly begins to decrease (degrade)
- ***The problem of deep model degradation*** is not a consequence of overfitting the model, the supplementation of additional layers leads to an even greater value of training error due to vanishing gradients
- ***Residual Networks*** (ResNet) solve the degradation problem
- The idea is to assume that some sequence of network layers approximate not the base mapping, but the residual mapping

* He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. – 2015. – [https://arxiv.org/pdf/1512.03385.pdf].

ResNet-50, 101, 152 (2)

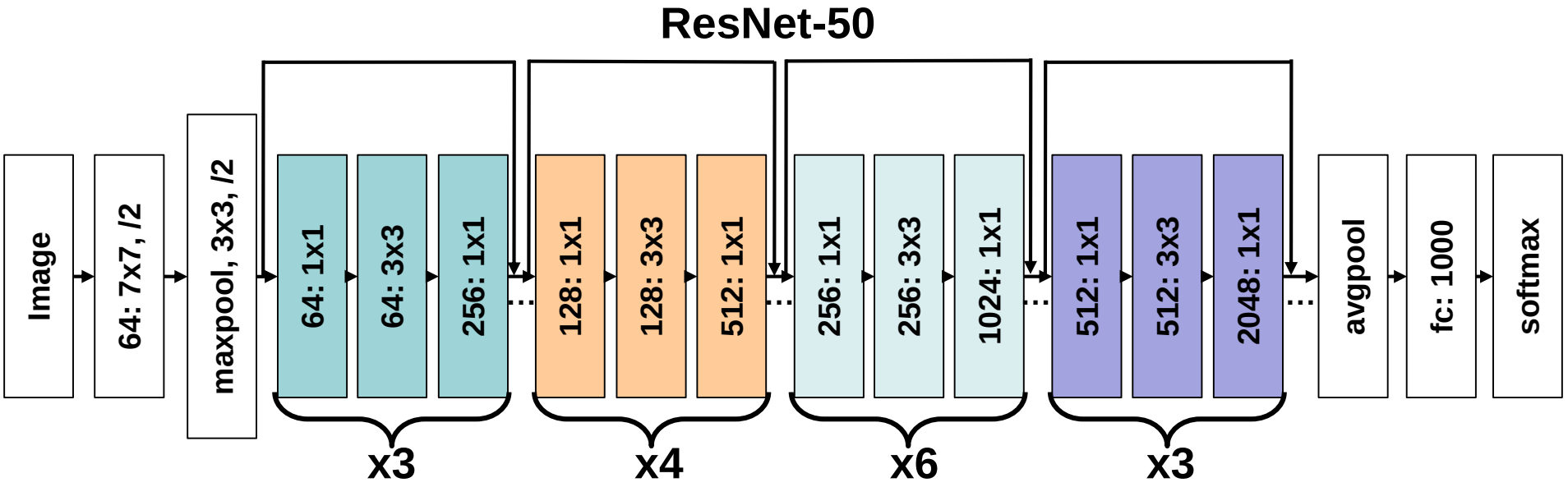
- $H(x)$ is a base mapping
- $F(x) = H(x) - x$ is a residual mapping
- The base mapping can be thought of as element-wise addition of feature maps $F(x) + x$

- $F(x)$ and x may have different dimensions, to correct this dimensions, it is enough to project the input feature vector $y = F(x, W_i) + W_s x$



ResNet-50, 101, 152 (3)

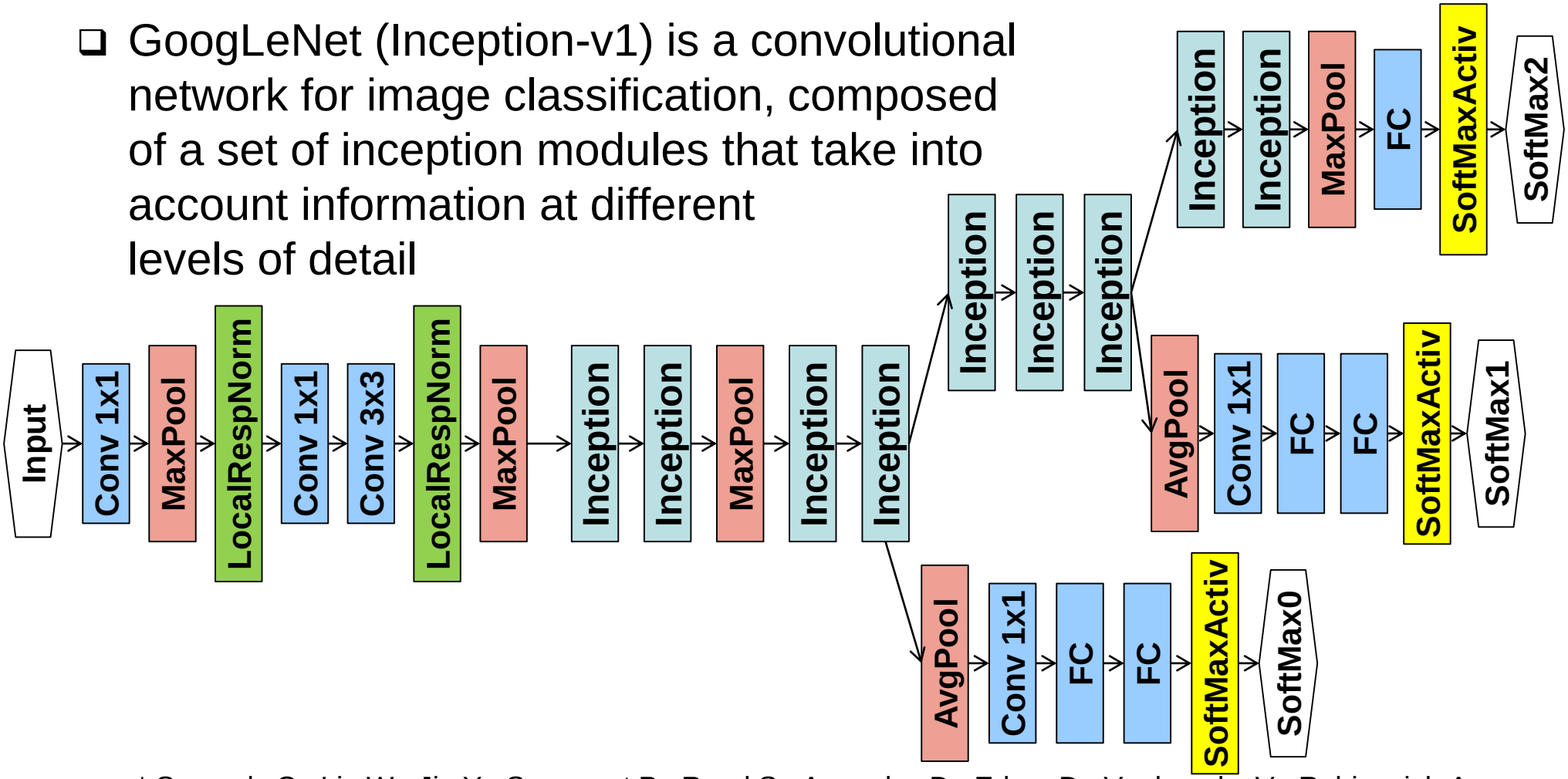
- The ResNet-50, 101, 152 models are constructed based on the principle of increasing convolutional layers, the problem of model degradation is solved by introducing residual connections for each successive three convolutional layers



* He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. – 2015. – [https://arxiv.org/pdf/1512.03385.pdf].

GoogLeNet (1)

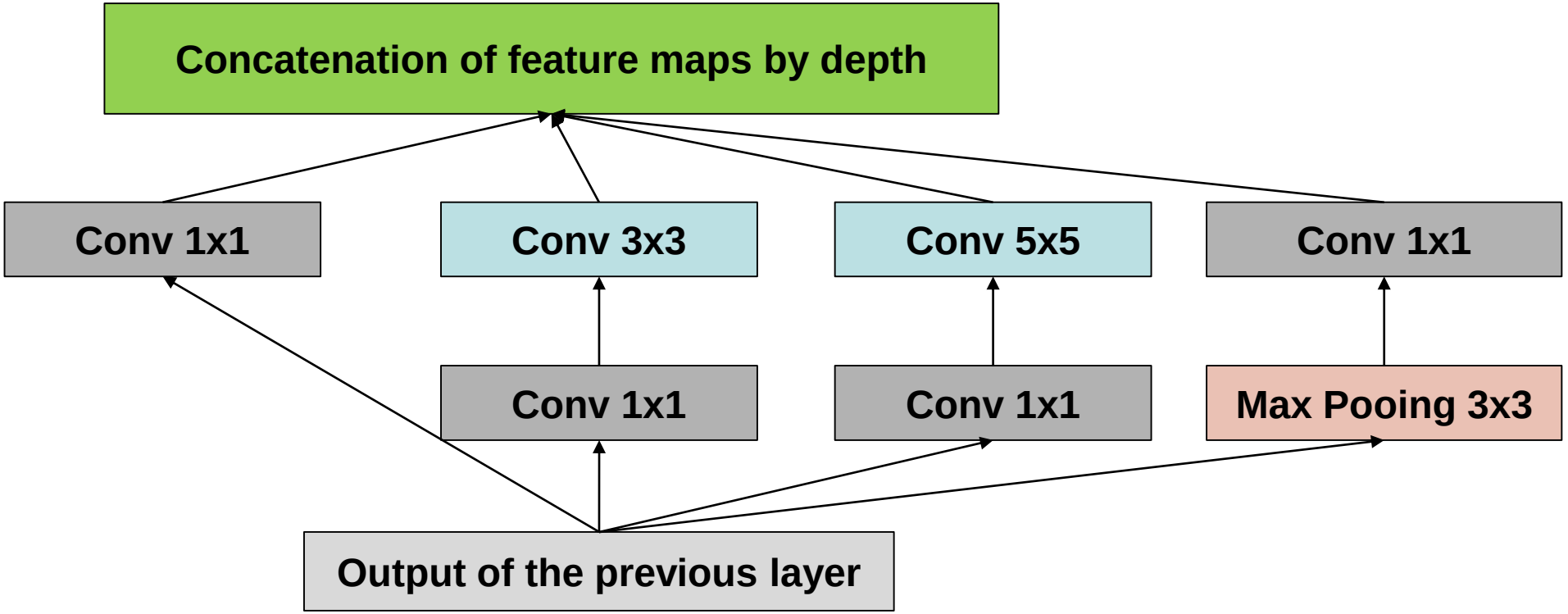
- GoogLeNet (Inception-v1) is a convolutional network for image classification, composed of a set of inception modules that take into account information at different levels of detail



* Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going Deeper with Convolutions. – 2014. – [https://arxiv.org/pdf/1409.4842.pdf].

GoogLeNet (2)

- Inception module:



* Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going Deeper with Convolutions. – 2014. – [https://arxiv.org/pdf/1409.4842.pdf].

GoogLeNet (3)

- GoogLeNet consists of 9 inception modules
- Each module uses convolutions of different kernel size to extract features of different scales
- The convolution kernels are small, so the number of trained model parameters is reduced. GoogLeNet contains about 10 times less parameters than the AlexNet model
- Auxiliary classifiers (softmax0 and softmax1) predict the class based on the features of a lower level. These classifiers allow “pushing” gradients to the early layers and thereby reduce the effect of vanishing gradients
- Branches leading to the auxiliary outputs are removed during model inference

Inception-v2

- Inception-v2 is a modification of the GoogLeNet model
  - 5x5 convolutions replaced by two consecutive 3x3 convolutions
  - As an activation function, a rectified linear unit (ReLU) is used
  - Before applying the activation function, batch normalization is applied
    - For a batch of samples, a set of feature maps is constructed
    - For individual elements in the feature map, the mean and standard deviation are determined
    - The value of each element is reduced by the mean and divided by the standard deviation

* Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. – 2015. – [https://arxiv.org/pdf/1502.03167.pdf].

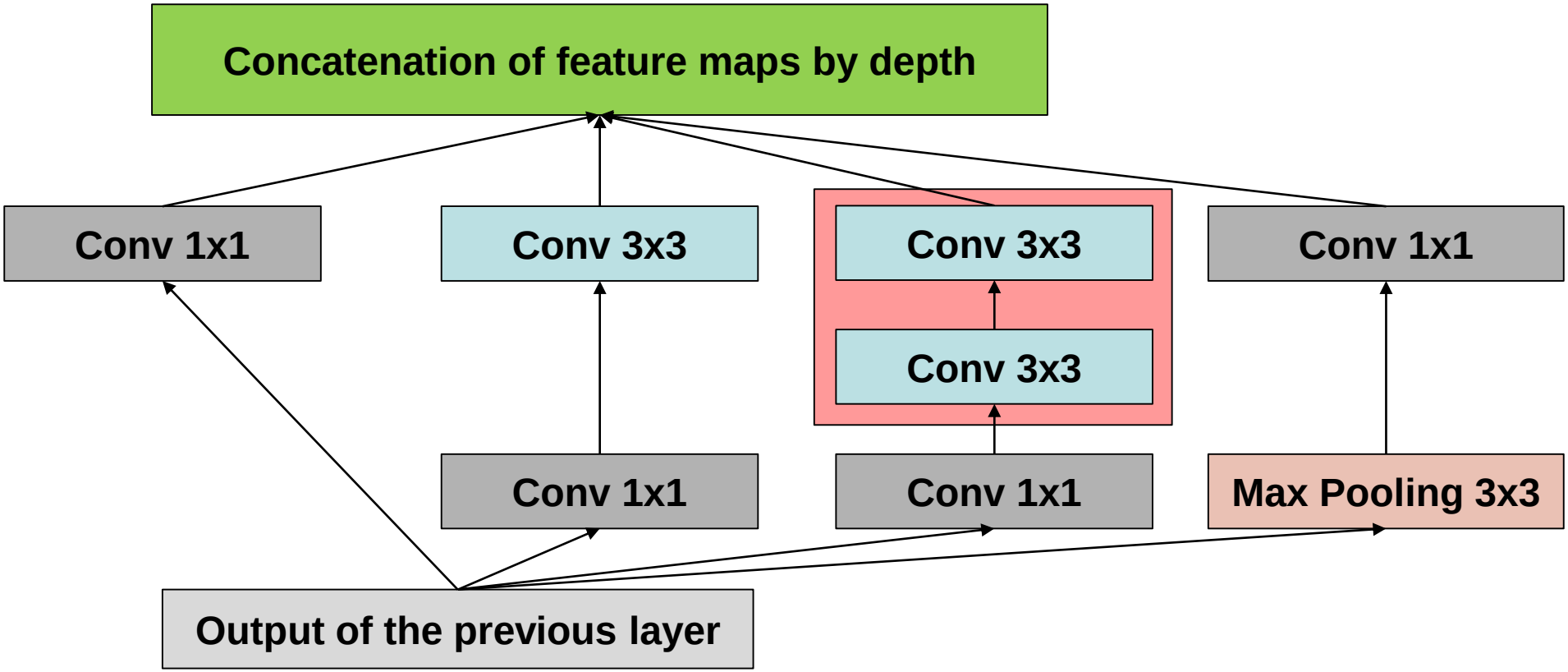
Inception-v3 (1)

- Inception-v3 improves the GoogLeNet model
  - Convolutions with larger kernels replaced by convolutions with smaller kernels
  - The model contains 3 types of modified inception modules
- Principles of constructing an effective model are introduced
  - Avoid bottlenecks in the network, especially on the initial layers
  - Pooling should be applied to the lower-dimensional feature maps to reduce computational complexity (see inception module scheme)
  - Balance network depth and width

* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf].

Inception-v3 (2)

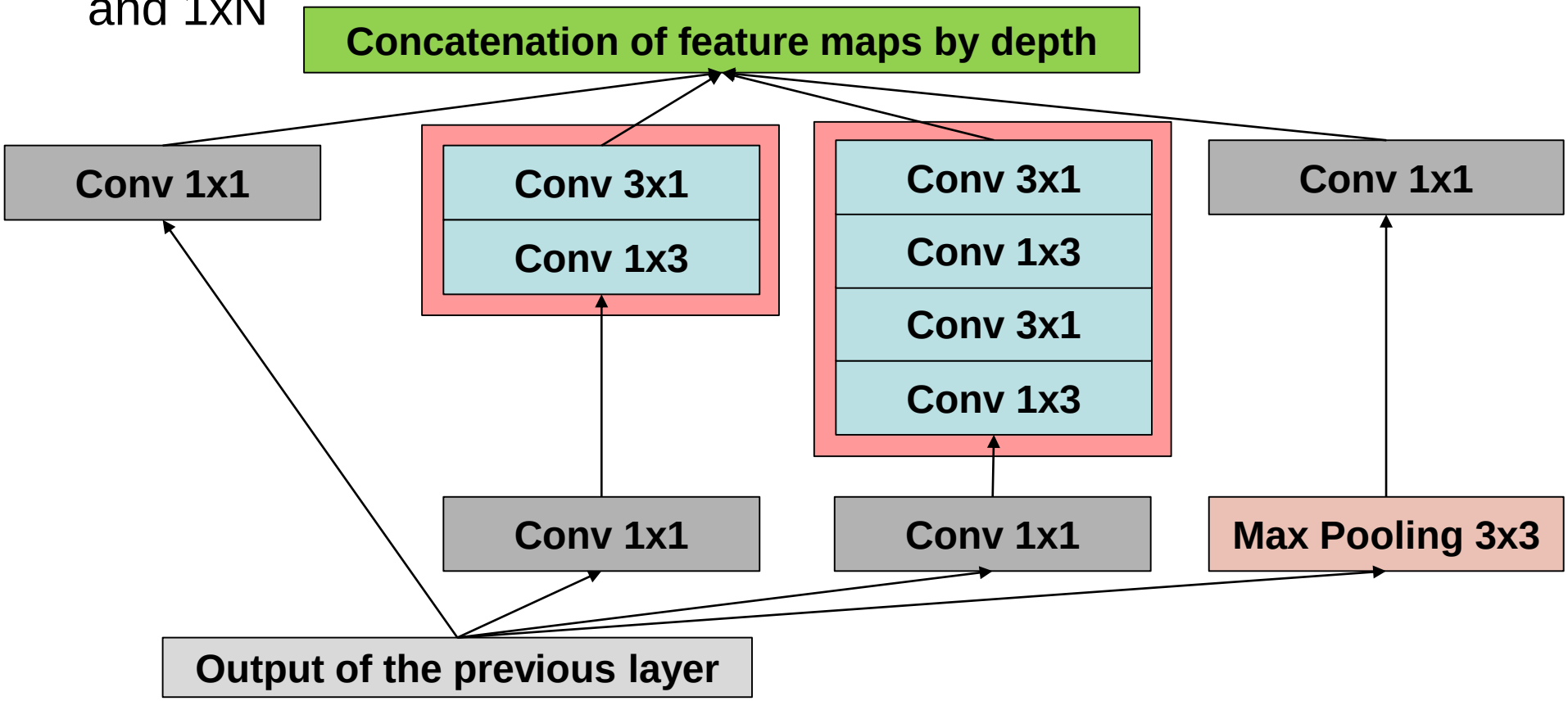
- ***Inception module A:*** replacing 5x5 convolutions with two 3x3 convolutions



* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf].

Inception-v3 (3)

- ***Inception module B:*** factorization of convolutions NxN to Nx1 and 1xN



* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf].

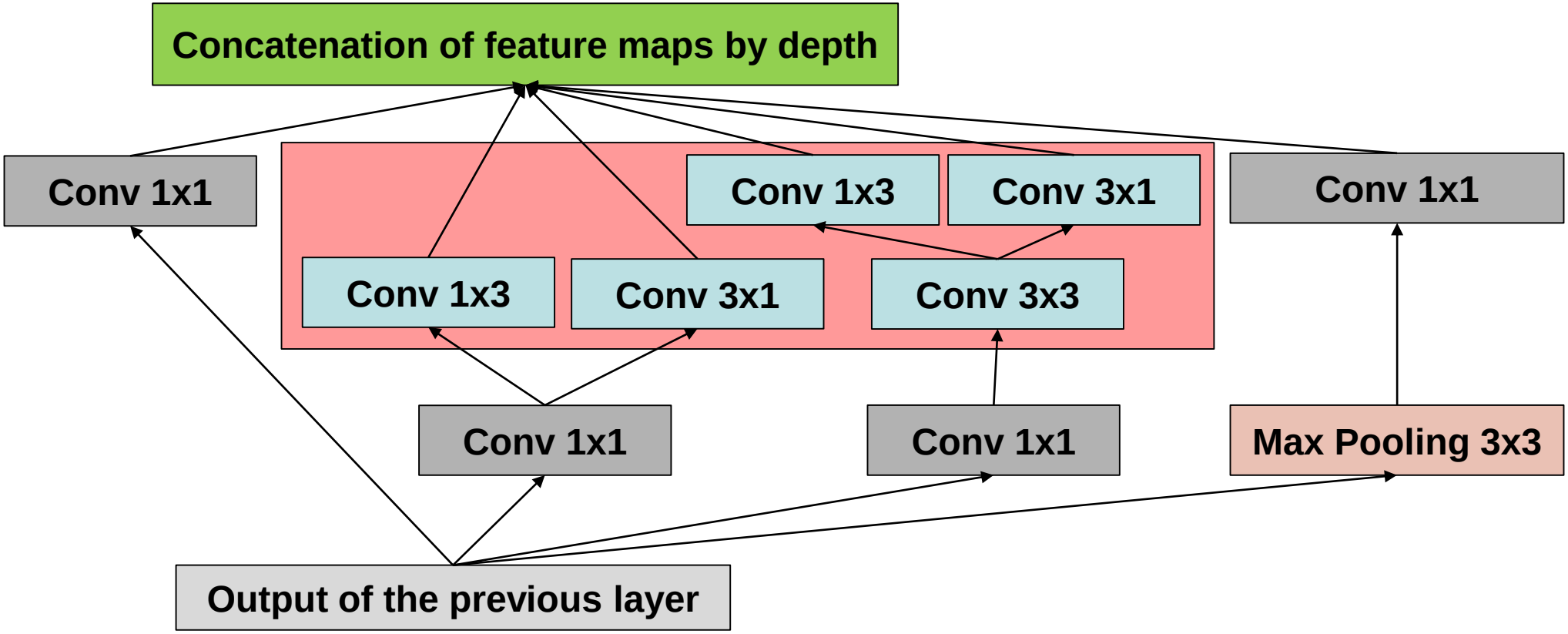
Inception-v3 (4)

- Using ***a convolution of 3x3 kernel size***, the number of parameters is ***3x3=9***
- Using ***convolutions of 3x1 and 1x3 sizes***, the number of parameters is ***3x1+1x3=6***
- ***The number of parameters is reduced by 33%***

- ***The number of parameters is reduced for the entire network, the likelihood that they will be overfitted is less, and the network may be deeper!***

Inception-v3 (5)

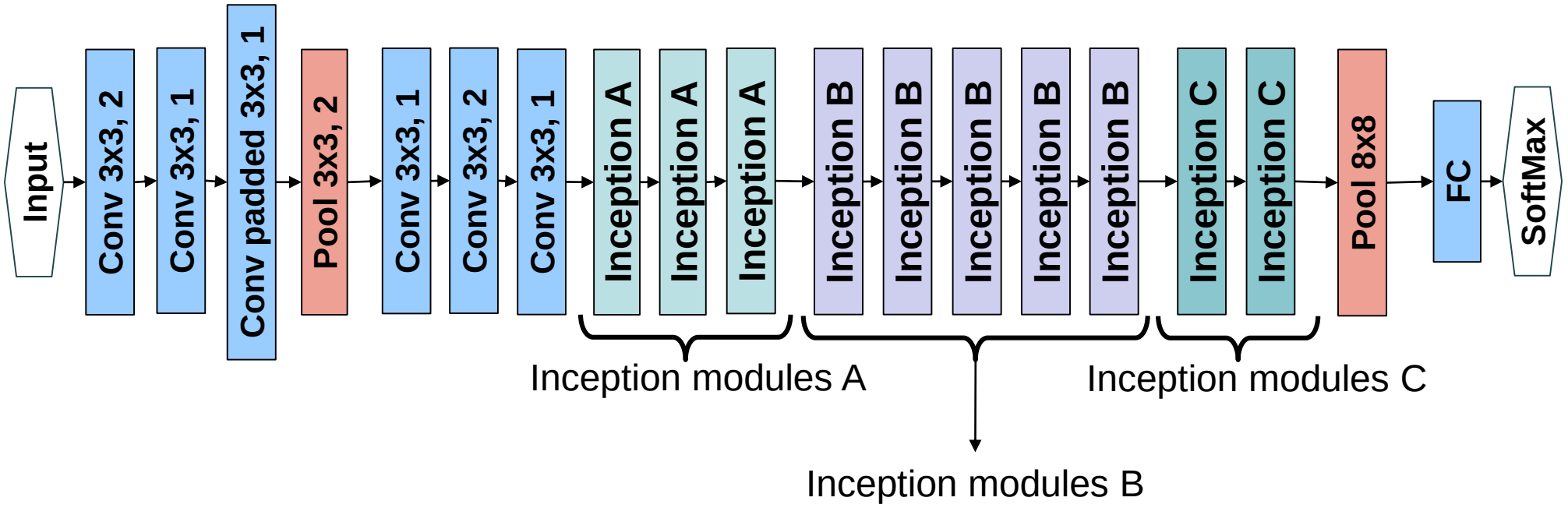
- ***Inception module C:*** increasing number of outputs



* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf].

Inception-v3 (6)

- The Inception-v3 model:



* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf], [https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf].

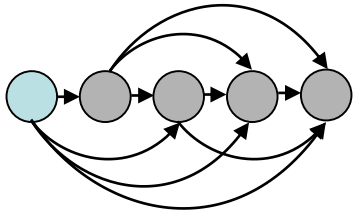
Inception-v3 (7)

- Authors of the Inception-v3 model* propose modifications of the represented model:
  - The number of inception modules of different types changes
  - Batch normalization is applied
  - Auxiliary classifier is introduced
  - Model regularization via label smoothing* is used
  - A scheme of efficient grid size reduction* is being implemented

* Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf], [https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf].

DenseNet-121, 169, 201, 264 (1)

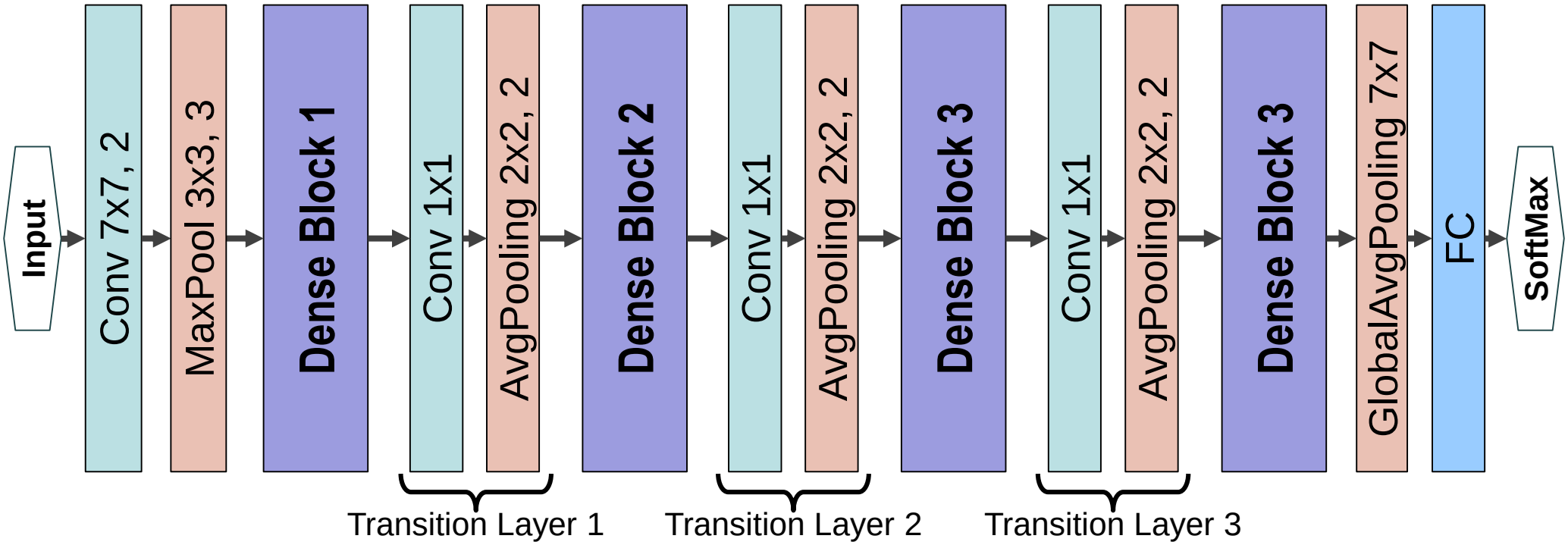
- DenseNet is an improvement of the ResNet models, aimed at reducing the number of model parameters
- Each layer of DenseNet receives information directly from all previous layers
- The model is implemented through constructing a sequence of dense blocks
  - Each block contains a set of convolutional layers
  - The input of each next layer is the concatenation of feature maps constructed on the previous layers



* Huang G., Liu Z., Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. – 2016. – [https://arxiv.org/pdf/1608.06993.pdf].

DenseNet-121, 169, 201, 264 (2)

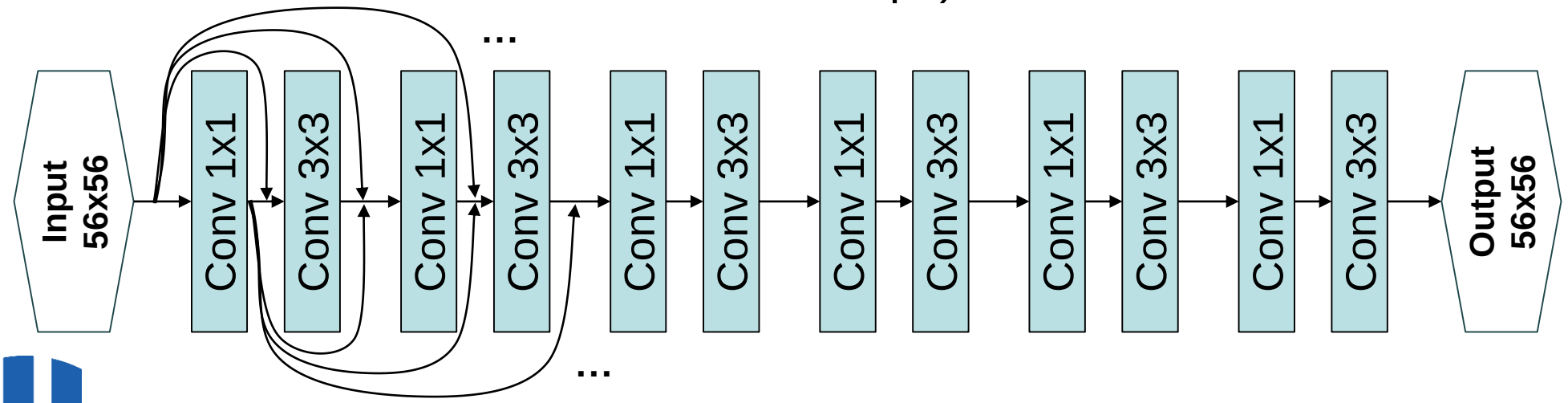
- The structure of the DenseNet models:



- The layers between two adjacent dense blocks are called ***transition layers***; they change dimension of a feature map

DenseNet-121, 169, 201, 264 (3)

- The dense block structure of DenseNet-121:
  - Dense Block 1: 6 x [Conv 1x1, Conv 3x3]
  - Dense Block 2: 12 x [Conv 1x1, Conv 3x3]
  - Dense Block 3: 24 x [Conv 1x1, Conv 3x3]
  - Dense Block 4: 16 x [Conv 1x1, Conv 3x3]
- Dense Block 1 (the arcs in the represented scheme show the transition of concatenated feature maps):



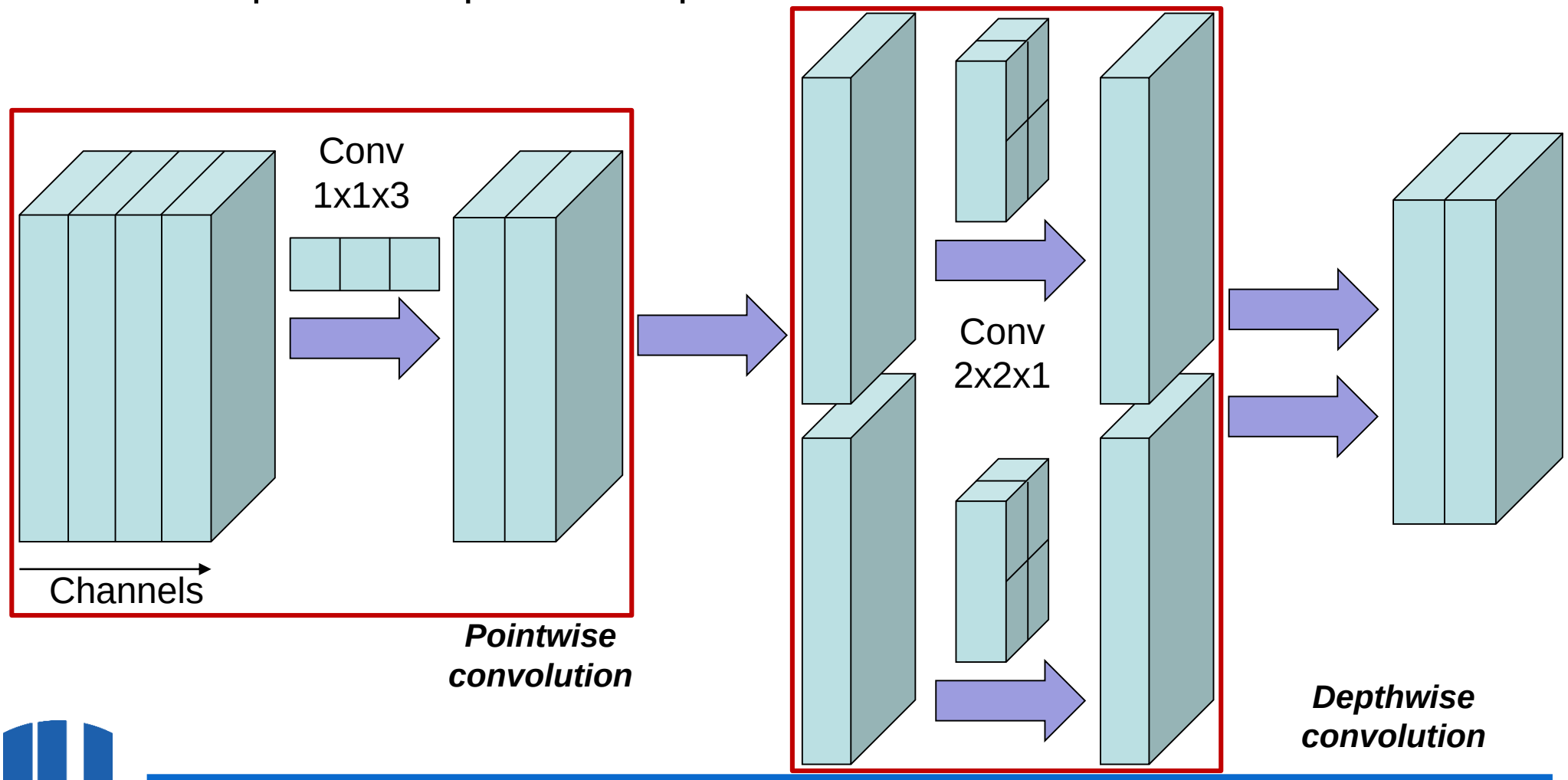
Xception (1)

- Xception (Extreme version of Inception) is a modification of the Inception-v3 model, which uses modified depthwise separable convolutions
- Modified depthwise separable convolution consist of two transformations:
  - ***Pointwise convolution*** is a 1x1 convolution of the third dimension corresponding to the channels of a feature map
  - ***Depthwise convolution*** is a NxN convolution applied to individual channels of a feature map
- In the classical form, the reverse order of transformations is applied

* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

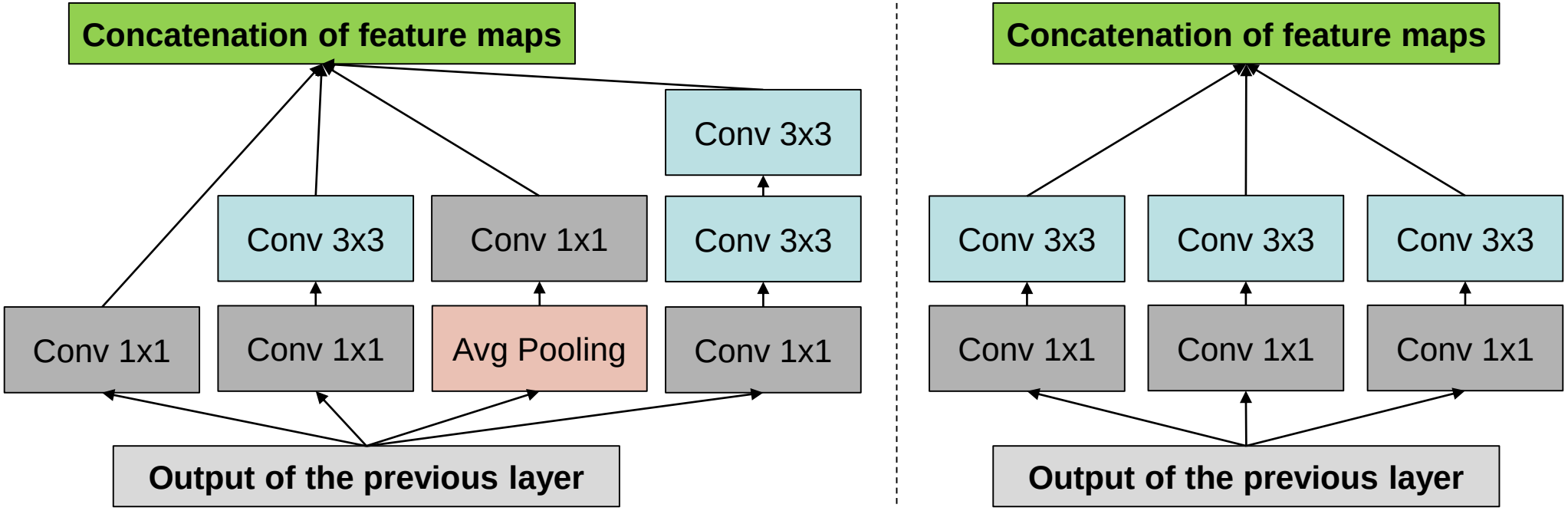
Xception (2)

- Example of a depthwise separable convolution:



Xception (3)

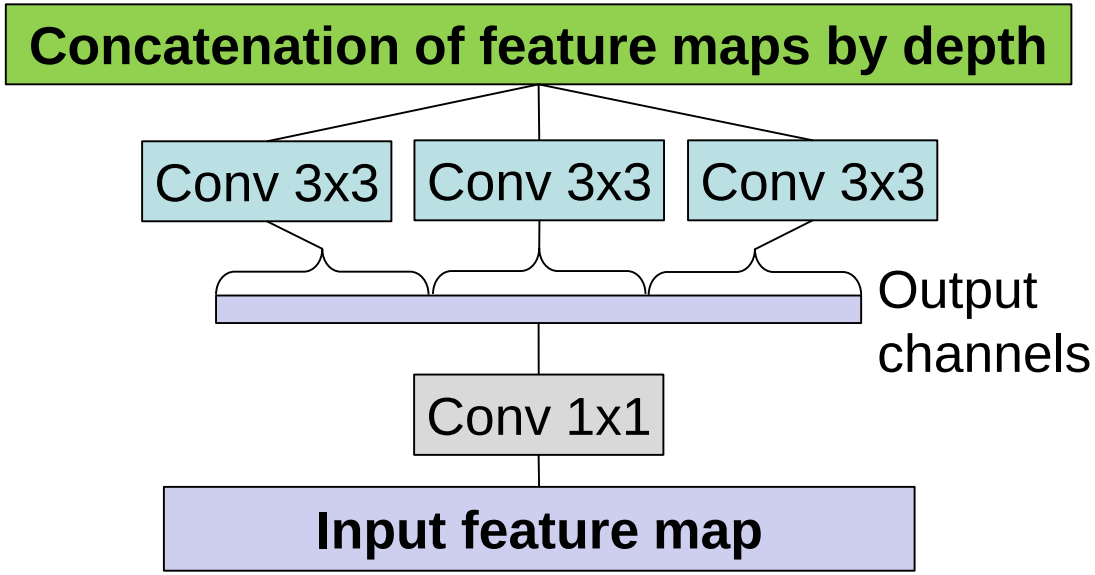
- ***How to use?*** Simplified inception module:
  - Only one kind of convolution kernel size 3x3 is used
  - The sequence of transformations containing pooling is removed



* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Xception (4)

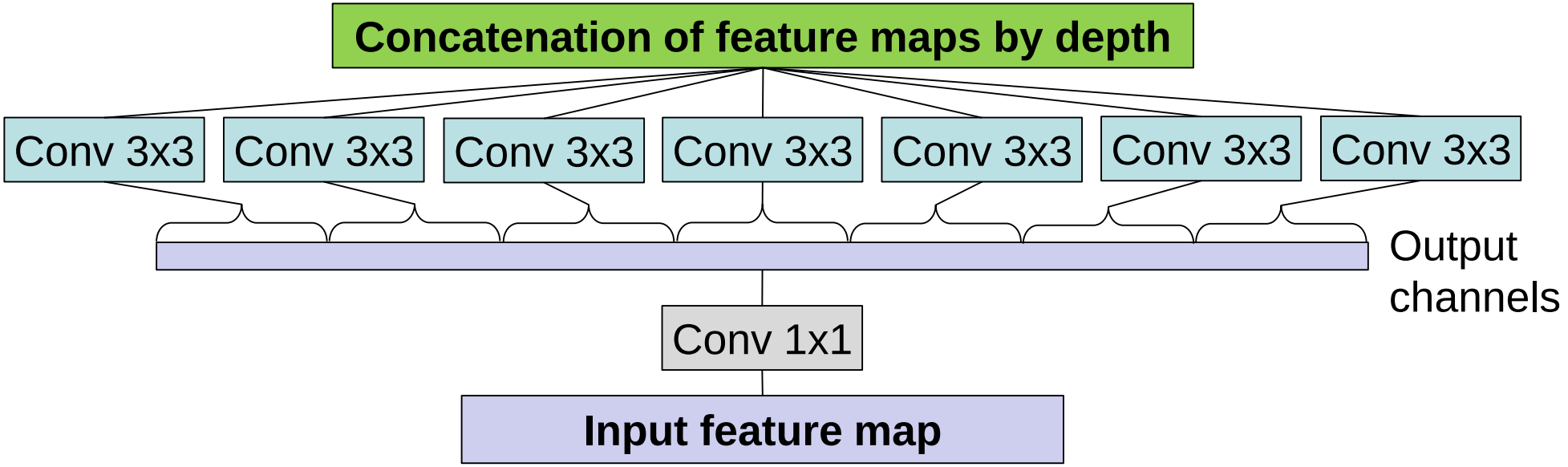
- A stronger hypothesis is that inter-channel and spatial correlations can be represented completely separately
- A simplified module can be represented as a 1x1 convolution followed by 3x3 spatial convolutions applied to disjoint output channels



* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Xception (5)

- The “extreme” version of the inception module is based on the given hypothesis
- First, an 1x1 convolution is used to represent the inter-channel correlations, further, the spatial correlations of each output channel are separately represented



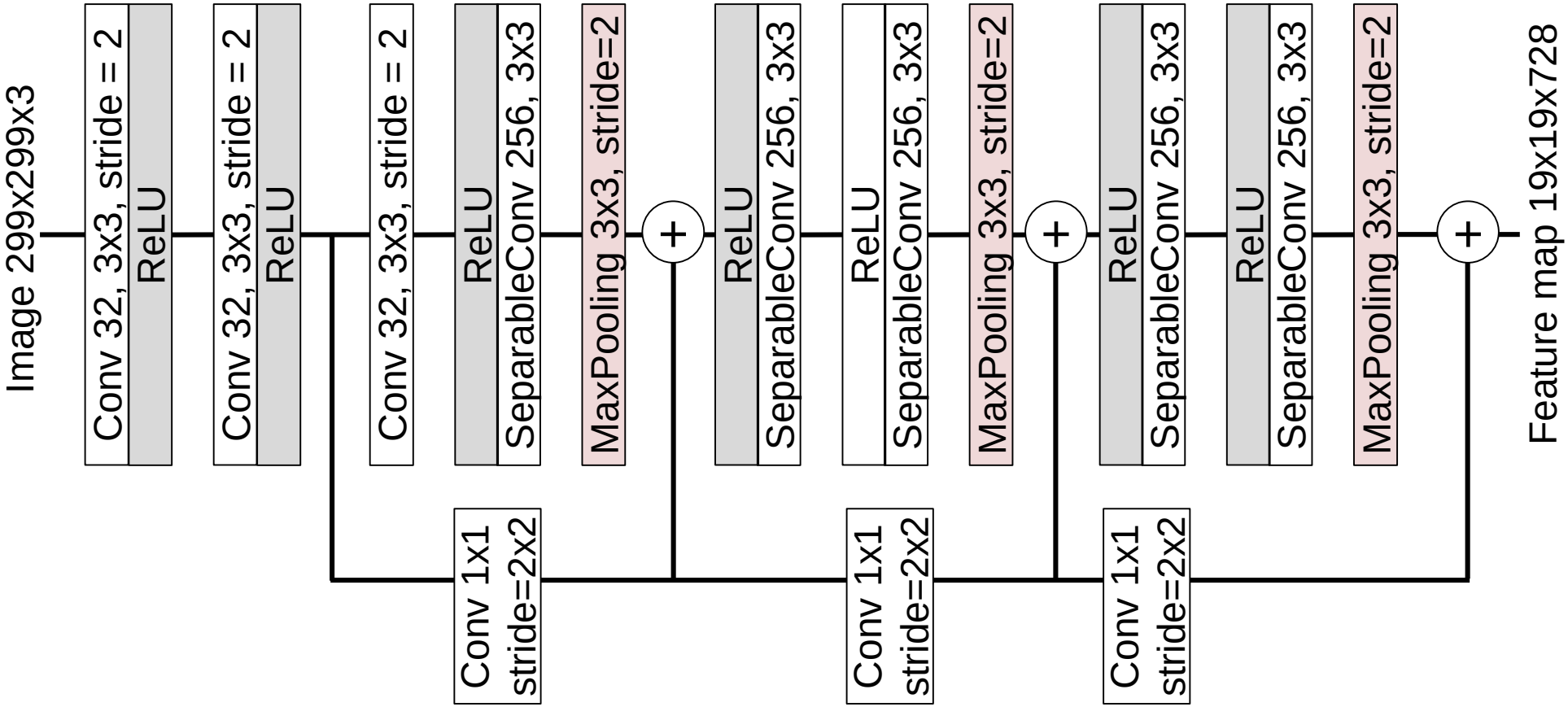
* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Xception (6)

- The “extreme” version of the inception module is identical to the modified depthwise separable convolution considered earlier
- The Xception model is constructed using “extreme” inception blocks and skip connections, which allow using feature maps from different levels of detail

Xception (7.1)

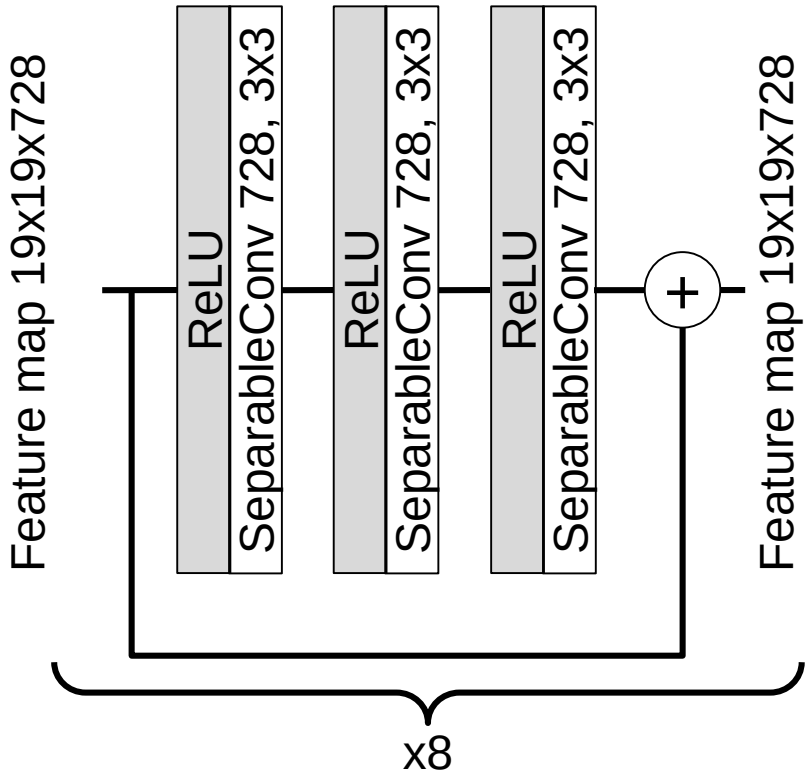
- The Xception model (Entry Flow):



* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Xception (7.2)

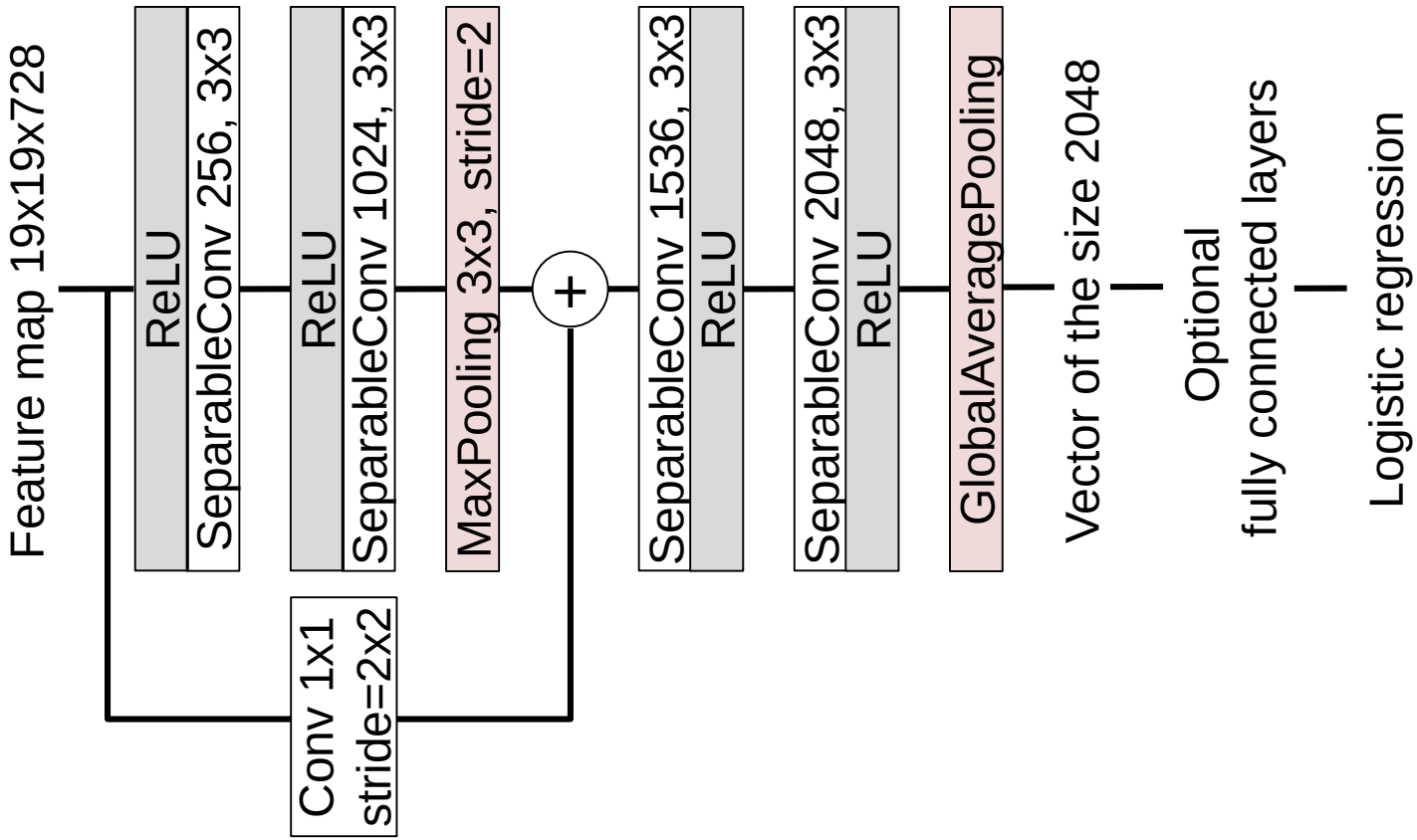
- The Xception model (Middle Flow):



* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

Xception (7.3)

- The Xception model (Exit Flow):



* Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].

MobileNet (1)

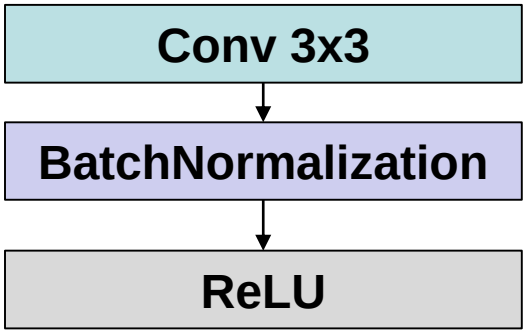
- MobileNets is a family of efficient deep neural networks for mobile and embedded devices
  - Small model size – fewer parameters
  - Low computational complexity – fewer multiplication and addition operations
- The models are based on depthwise separable convolutions in order to construct lightweight deep neural networks
- An algorithm for selecting a model of the correct size is introduced by searching for a compromise between the model complexity and accuracy

* Howard A.G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – 2017. – [https://arxiv.org/pdf/1704.04861.pdf].

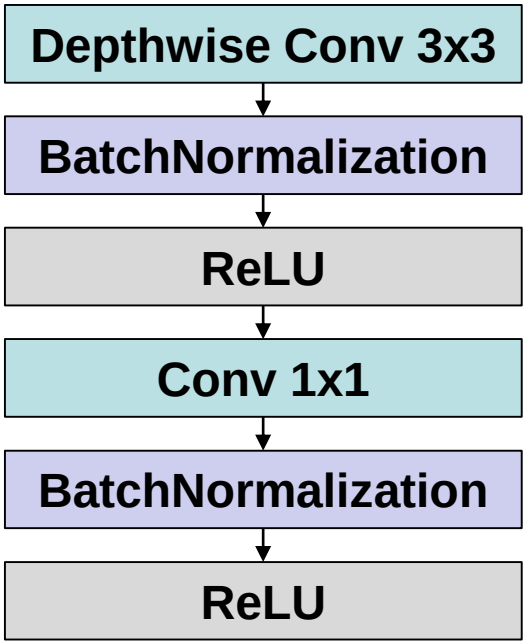
MobileNet (2)

- The basic convolutional layer of the MobileNet model:

Standard convolutional layer with batch normalization



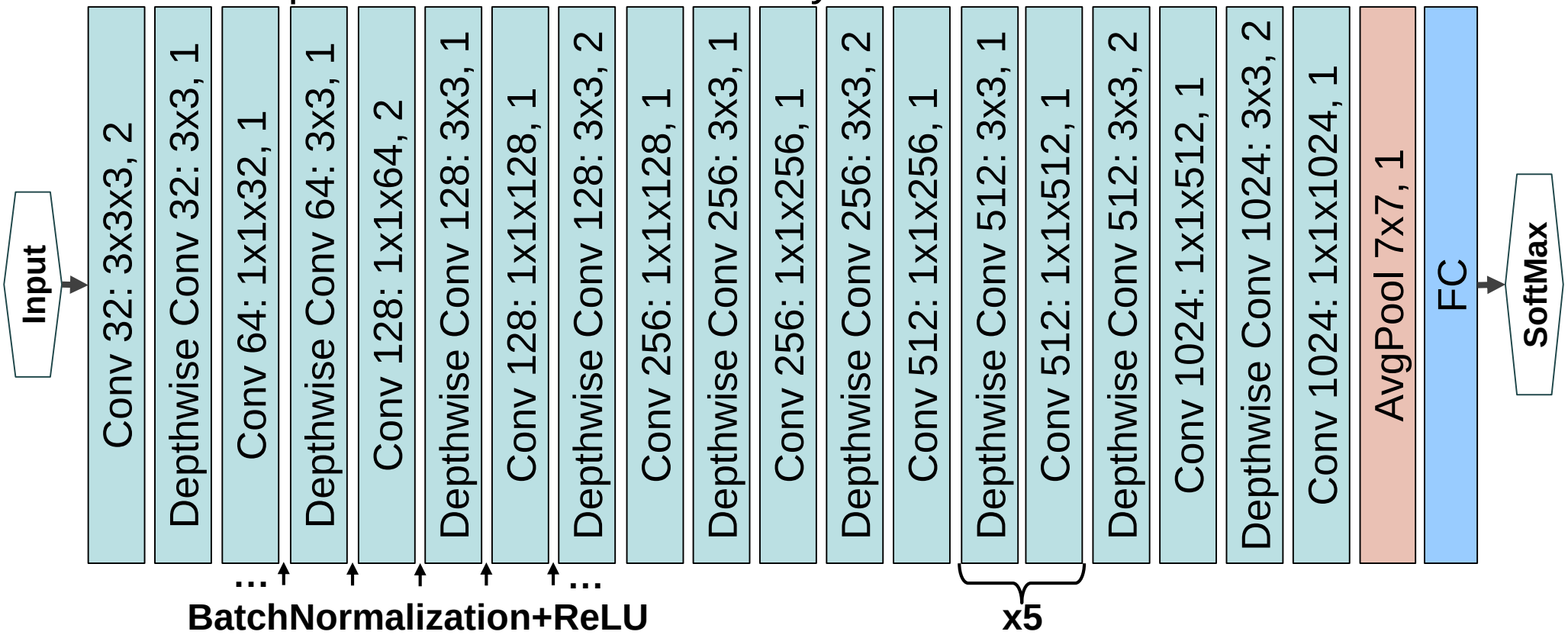
Depthwise separable convolution with depthwise and pointwise layers



* Howard A.G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – 2017. – [https://arxiv.org/pdf/1704.04861.pdf].

MobileNet (3)

- The sequence of the MobileNet layers:



* Howard A.G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – 2017. – [https://arxiv.org/pdf/1704.04861.pdf].

MobileNet (4)

- The MobileNet features:
  - It contains 28 layers (counting depthwise and pointwise convolutions as separate layers)
  - There is no pooling after convolutional layers
  - To reduce the dimension of feature maps, convolutions with the stride of 2 are used
  - Each convolutional layer is followed by batch normalization and the ReLU activation function

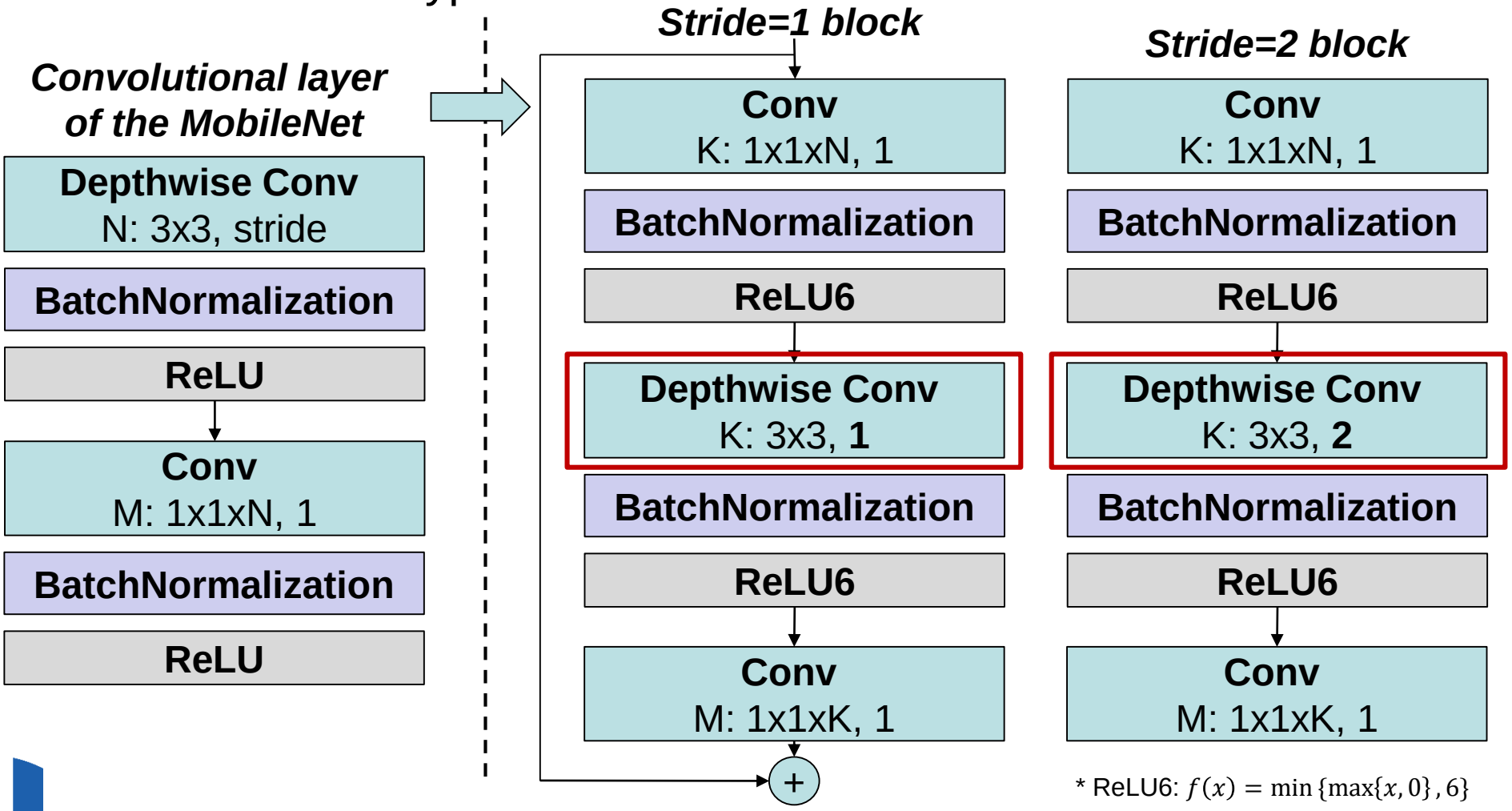
MobileNetV2 (1)

- MobileNetV2 is a modification of the MobileNet model, in which an inverted residual block is introduced
- Two types of inverted residual blocks are introduced:
  - “Stride=1 block” is a bottleneck block
  - “Stride=2 block” is a sequence of convolutional layers that reduce size of a feature map
- Each block contains 3 convolutional layers:
  - 1x1 convolution and the ReLU activation function
  - Depthwise convolution
  - 1x1 convolution (!without activation function)

* Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. – 2018. – [https://arxiv.org/pdf/1801.04381.pdf].

MobileNetV2 (2)

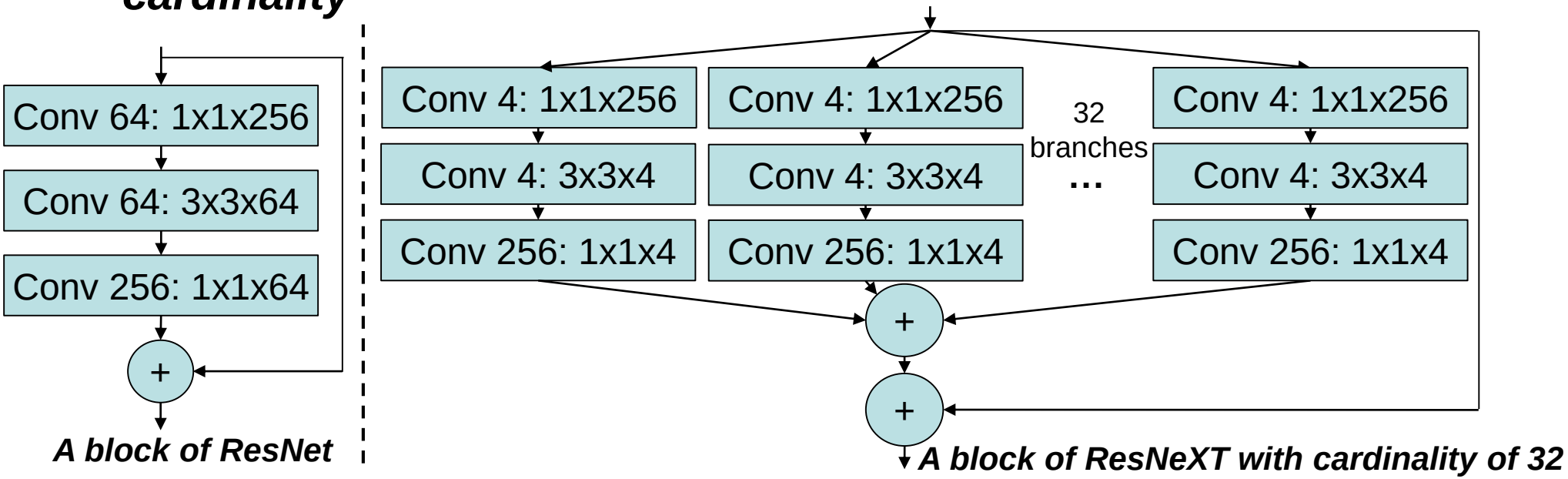
- The structure of typical blocks:



* ReLU6: $f(x) = \min\{\max\{x, 0\}, 6\}$

ResNeXT

- ResNeXT is a deep convolutional network consisting of repeating residual blocks that aggregate a set of transformations with the same topology
- The number of branches with the same topology is called ***cardinality***



* Xie S., Girshick R., Dollar P., Tu Z., He K. Aggregated Residual Transformations for Deep Neural Networks. – 2017. – [https://arxiv.org/pdf/1611.05431v2.pdf].

EfficientNet (1)

- EfficientNets is a class of models whose development goal is to preserve high accuracy of solving a problem and increase efficiency of a model (reduce the number of parameters and reduce computational complexity)
- Scaling any network dimension (depth, resolution of the input image, width is the number of channels in feature maps) can lead to higher accuracy while designing network correctly
- It is important to balance all network dimensions (depth, resolution, and width) while scaling the network to obtain high accuracy and efficiency

* Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. – 2019. – [https://arxiv.org/pdf/1905.11946.pdf].

EfficientNet (2)

- The EfficientNet authors propose ***a compound scaling method***
- A compound coefficient ϕ is introduced for uniform scaling of depth, width and resolution:
  - Depth: $d = \alpha^{\phi}$
  - Width: $\omega = \beta^{\phi}$
  - Resolution: $r = \gamma^{\phi}$
- Ratio: $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2,\ \alpha \geq 1,\ \beta \geq 1,\ \gamma \geq 1$

$$FLOPS \cong O(d\omega^2 r^2)$$
$$FLOPS < 2^{\phi}$$

- ϕ is a user-specified coefficient that controls how many computational resources are available for model scaling
- α, β, γ specify how to assign these resources to network width, depth, and resolution respectively

EfficientNet (3)

- EfficientNet-B0 is a basic neural network which is constructed using inverted residual blocks introduced in MobileNetV2
- EfficientNet-B1, ..., B7 were obtained by searching for the optimal ratio of depth, width and resolution using the proposed scaling method

COMPARISON OF CLASSIFICATION ACCURACY AND COMPLEXITY OF DEEP MODELS ON THE IMAGENET DATASET

Test dataset

- A comparison of classification accuracy is represented on the test dataset of ImageNet
- Measurements are compiled by researchers based on the results of the ILSVRC contest and published on the Internet [https://paperswithcode.com/sota/image-classification-on-imagenet]

Quality metrics

- Supposed N is the number of image categories
- For every image $I_j, j = \overline{1, S}$ in the dataset, classification model predicts a vector of confidences $p^j = \left(p_1^j, p_2^j, \dots, p_N^j\right)$, where p_i^j is the confidence of the assumption that the image I_j belongs to the class i
- ***Top-K accuracy*** is as follows:

$$topK = \frac{\sum_{j=1}^{S} 1_{\left\{i_1^j, i_2^j, \dots, i_K^j\right\}}\left(l_j\right)}{S},$$

where $\left\{i_1^j, i_2^j, \dots, i_K^j\right\} \subseteq \{1,2, \dots, N\}$, $p_{i_1^j}^j, p_{i_2^j}^j, \dots, p_{i_K^j}^j$ are K maximal confidences, l_j is the class to which the image I_j belongs according to the groundtruth, $1_{\left\{i_1^j, i_2^j, \dots, i_K^j\right\}}\left(l_j\right)$ is an indicator function

Comparison of classification accuracy and complexity of deep models (1)

Model	Year	top-1,%	top-5,%	Number of parameters, million
AlexNet	2012	63.3	84.6	60
OverFeat	2013	66.04	86.76	–
VGG-16	2014	74.4	91.9	138
GoogLeNet	2014	69.8	89.9	5
ResNet-101	2015	78.25	93.95	40
Inception-v2	2015	74.8	92.2	11.2
Inception-v3	2015	78.8	94.4	23.8
DenseNet-201	2016	78.54	94.46	20
Xception	2016	79	94.5	22.8
MobileNet-224	2017	70.6	89.5	–
ResNeXT-101 64x4	2017	80.9	95.6	83.6
EfficientNet-B0	2019	76.3	93.2	**5.3**
EfficientNet-B7	2019	**84.4**	**97.1**	66

Increasing accuracy and number of parameters

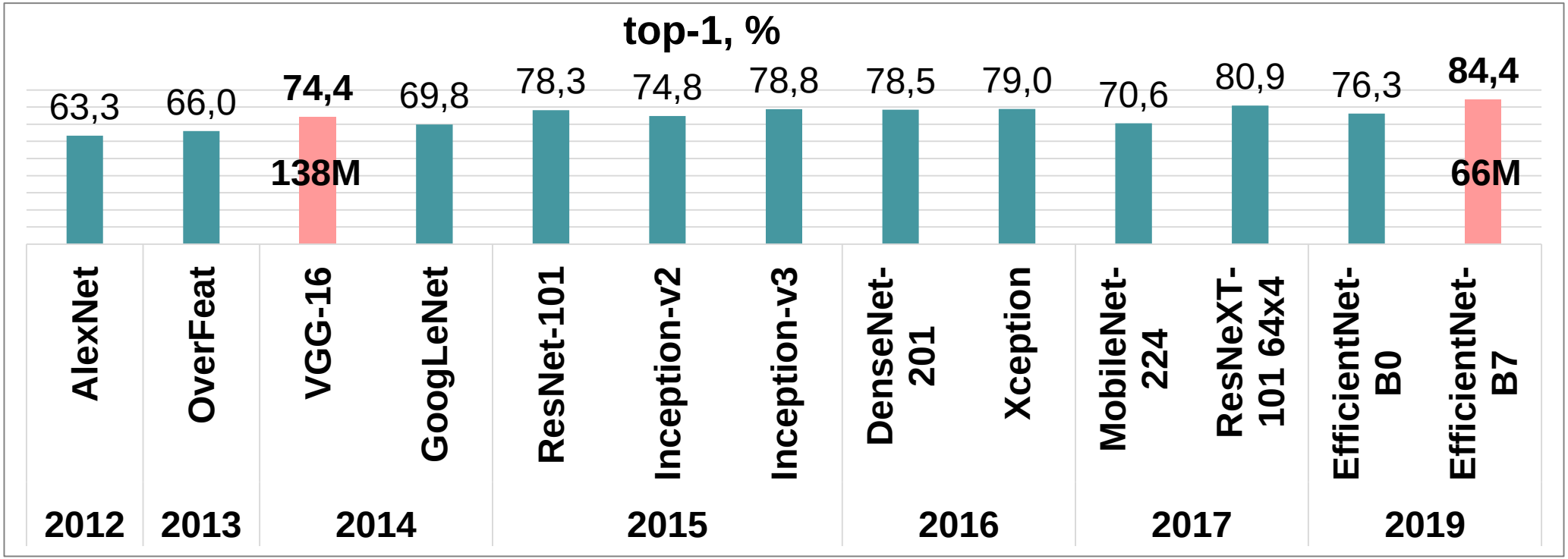
Increasing accuracy and decreasing number of parameters

Reducing model complexity and searching for the best model

* Image Classification on ImageNet [https://paperswithcode.com/sota/image-classification-on-imagenet].

Comparison of classification accuracy and complexity of deep models (2)

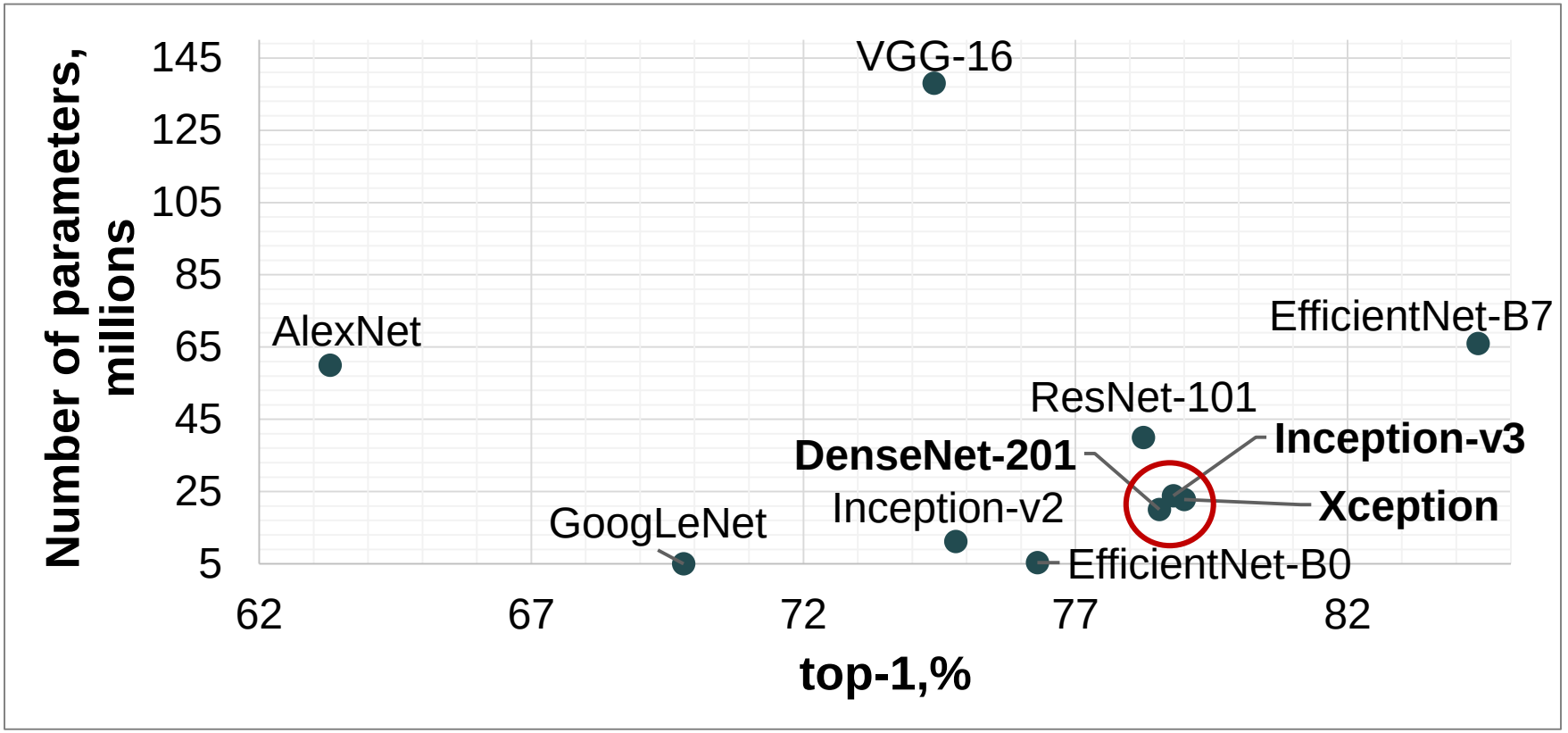
- Changing the top-1 accuracy on the ImageNet dataset for the selected models:



- ***Over 5 years, the top-1 accuracy increased by 10%, and the number of parameters decreased by ~2 times***

Comparison of classification accuracy and complexity of deep models (3)

- Until 2014, the goal of model development is to improve the accuracy of solving a problem; from 2015, the goal is to increase the efficiency of the model and ensure quality growth (compromise)



Comparison of classification accuracy and complexity of deep models (4)

- Notes:
  - Improving the model efficiency means reducing the computational complexity of the model (the number of performed operations) and reducing the model size (number of parameters)
  - The model complexity is not directly related to the number of parameters
  - In practice, complexity is usually much more important

Conclusion

- Deep models for image classification are not limited to those discussed in this lecture; there are many modifications of basic architectures
- Nowadays a large number of models for solving problems from other problem areas use the described architectures based on transfer learning approach, or use the basic building blocks of the considered models (it will be shown later in the lectures)
- ***The optimal model is a compromise between accuracy and complexity***
  - The accuracy is determined by the requirements for solving a practical problem
  - Complexity is determined by available computational resources and runtime requirements

Literature (1)

- Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in neural information processing systems. – 2012. – [http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf].
- Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. – 2013. – [https://arxiv.org/pdf/1312.6229.pdf].
- Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. – 2014. – [https://arxiv.org/pdf/1409.1556.pdf].

Literature (2)

- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going Deeper with Convolutions. – 2014. – [https://arxiv.org/pdf/1409.4842.pdf].
- He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. – 2015. – [https://arxiv.org/pdf/1512.03385.pdf].
- Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. – 2015. – [https://arxiv.org/pdf/1502.03167.pdf].
- Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. – 2015. – [https://arxiv.org/pdf/1512.00567.pdf], [https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf].

Literature (3)

- Huang G., Liu Z., Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. – 2016. – [https://arxiv.org/pdf/1608.06993.pdf].
- Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. – 2016. – [https://arxiv.org/pdf/1610.02357.pdf].
- Howard A.G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – 2017. – [https://arxiv.org/pdf/1704.04861.pdf].
- Xie S., Girshick R., Dollar P., Tu Z., He K. Aggregated Residual Transformations for Deep Neural Networks. – 2017. – [https://arxiv.org/pdf/1611.05431v2.pdf], [https://ieeexplore.ieee.org/document/8100117].

Literature (4)

- Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. – 2018. – [https://arxiv.org/pdf/1801.04381.pdf], [https://ieeexplore.ieee.org/document/8578572].
- Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. – 2019. – [https://arxiv.org/pdf/1905.11946.pdf].

Authors

- **Turlapov Vadim Evgenievich,** Dr., Prof., department of computer software and supercomputer technologies
  vadim.turlapov@itmm.unn.ru
- **Vasiliev Engeny Pavlovich,** lecturer, department of computer software and supercomputer technologies
  evgeny.vasiliev@itmm.unn.ru
- **Getmanskaya Alexandra Alexandrovna,** lecturer, department of computer software and supercomputer technologies
  alexandra.getmanskaya@itmm.unn.ru
- **Kustikova Valentina Dmitrievna**
  Phd, assistant professor, department of computer software and supercomputer technologies
  valentina.kustikova@itmm.unn.ru