Image Classification

Jing Liu

Content：

1 Introduction to Image Classification

1. What is Image Classification + The Process of Image Classification Approach
2. The Importance of Image Classification
3. The Challenges in Image Classification Area
4. The Introduction to Some Benchmark Datasets

2 Image Classifiers

Before 2012: Traditional Image Classification Methods

1. Nearest Neighbor
2. K-Nearest Neighbor
3. SVM
4. Random Forest

After 2012: Deep Learning Algorithms

1. Introduction to CNN

   LeNet-5

2. Benchmark Networks

   AlexNet； VGG； GoogLeNet； ResNet

3. SENet； PyramidNet

3 The Main Research Direction of Image Classification

1. Fine-grained Image Classification

2. Imbalanced Image Classification

- Classical Method:

  Data-level Approach

  Algorithm-level Approach

- Methods about Convolutional Neural Network

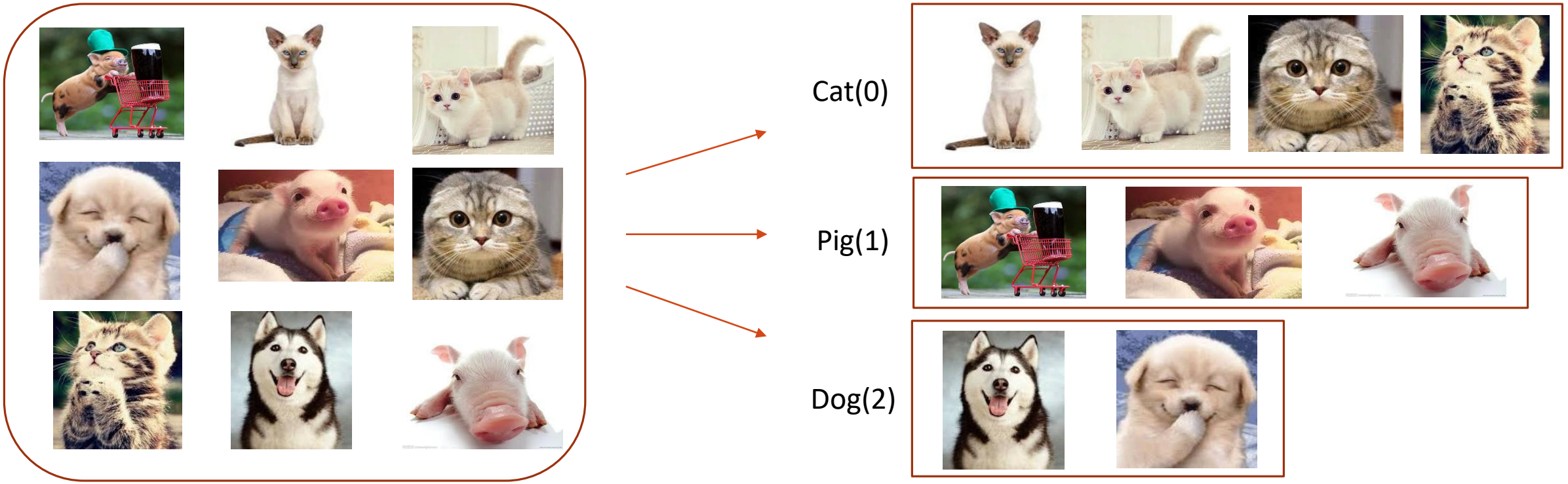
  Use GAN to Generate Minority Samples

  Optimize the Loss of CNN Algorithms

1 Introduction to Image Classification

What is Image Classification in Machine Learning?

Image classification: an image **processing** method that distinguishes different kinds of **targets** according to different **features** reflected in the image information(semantic information).

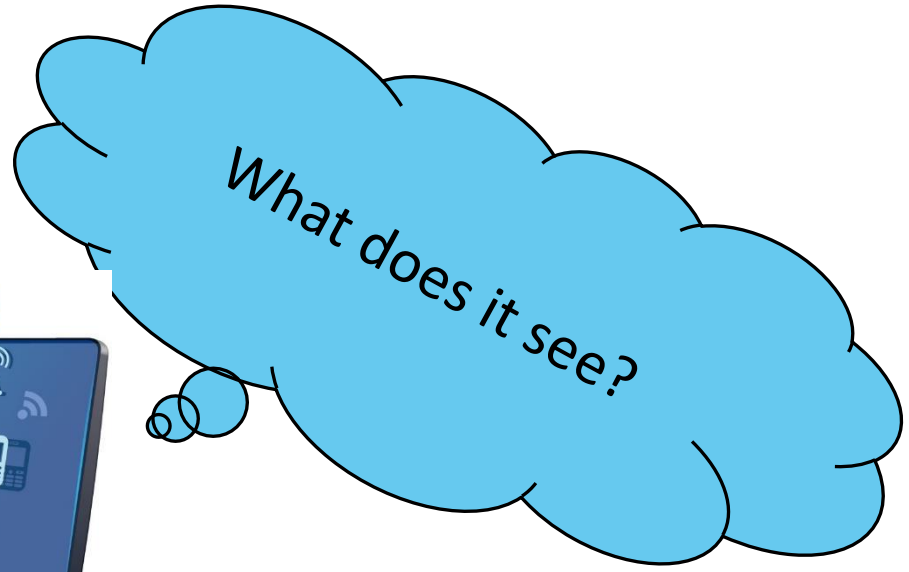


Baidu Encyclopedia

This seems like a really easy problem because of your own visual system in your brain.

But this is actually a really hard problem for a machine.

What we see



An image is just a big grid of numbers between [0, 255]:

(194, 153, 151)

194	230	250	254	232	206	211	232	229	179	189	186	158	132	132	135	144	153	147	144	162	86	49	65	30	31	41	26
213	243	235	215	189	188	227	238	234	233	181	135	128	112	129	138	136	127	139	139	116	103	60	57	58	68	55	39
237	243	241	209	217	247	226	242	217	153	132	146	120	128	151	165	153	152	154	127	108	120	134	39	57	34	18	55
253	245	238	203	244	222	214	226	168	122	128	121	102	100	149	187	177	167	182	123	109	115	117	89	42	43	49	33
230	233	205	236	235	196	206	203	121	147	112	90	94	111	152	188	186	191	202	118	100	89	104	129	75	54	35	44
220	212	202	241	227	183	192	166	138	150	154	75	53	119	176	181	174	179	195	84	86	93	95	150	131	58	65	36
182	184	196	242	201	169	187	132	152	175	176	88	29	150	190	193	163	177	150	53	102	154	126	144	142	90	39	34
187	158	211	235	193	168	225	135	118	175	161	172	143	170	192	219	208	186	181	124	157	148	188	186	167	87	37	27
168	167	195	221	181	152	182	158	75	131	163	172	187	177	173	177	179	179	194	179	183	189	190	191	158	114	66	58
161	157	154	179	157	165	171	175	114	80	110	128	126	137	157	169	160	167	153	149	161	174	180	148	97	146	74	58
148	168	142	145	148	202	186	161	158	105	94	100	101	92	102	132	127	130	106	121	127	104	96	87	132	173	84	58
154	180	167	158	158	178	173	145	136	127	93	97	85	55	83	75	64	81	68	76	87	74	67	103	236	178	110	61
128	151	168	169	161	141	124	104	101	97	106	88	74	82	93	84	113	124	100	66	53	63	107	200	236	175	86	48
92	91	114	127	107	84	89	156	182	183	175	153	145	148	160	153	150	142	132	136	126	103	122	175	233	184	109	70

What we see

What the computer sees

Image Classification

92% Pig

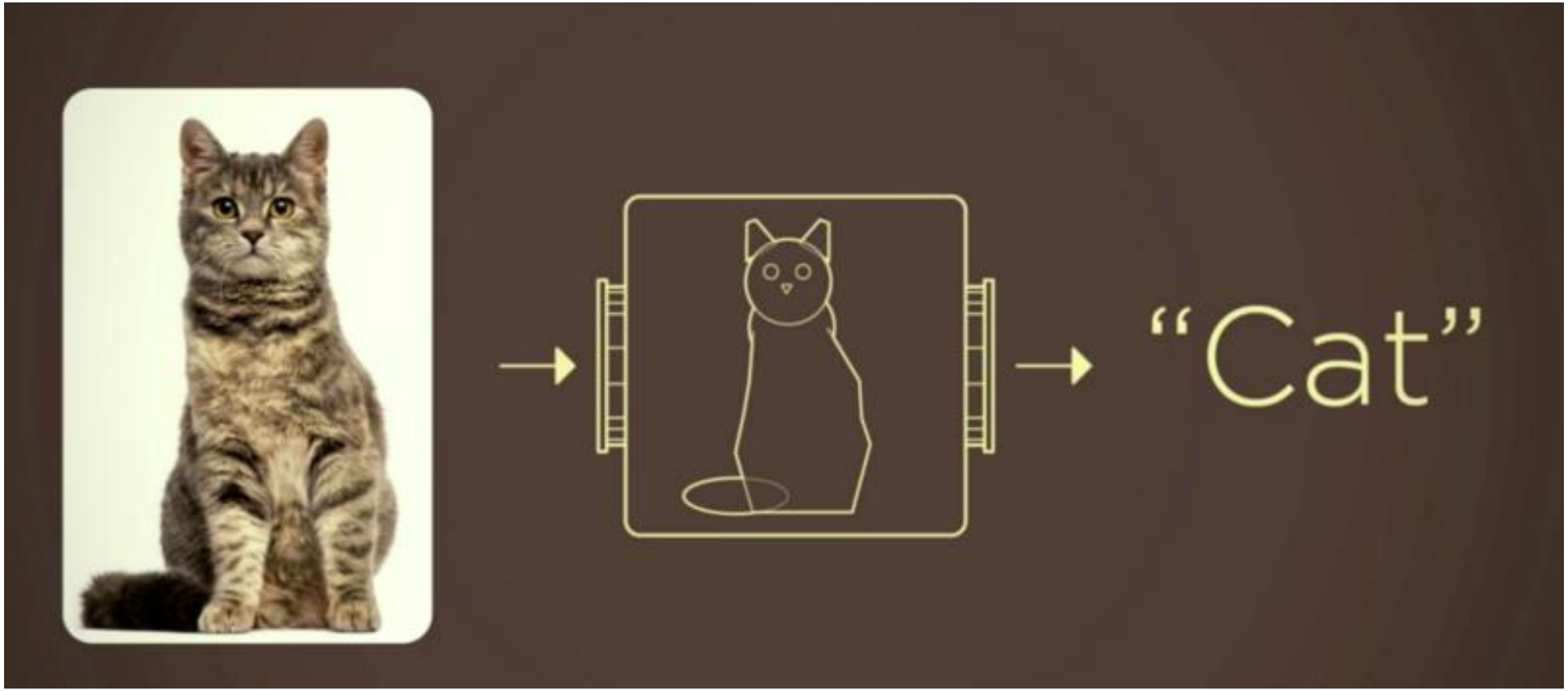
4.8% Cat

3.2% Dog

What the computer outputs

The Process of Image Classification Approach

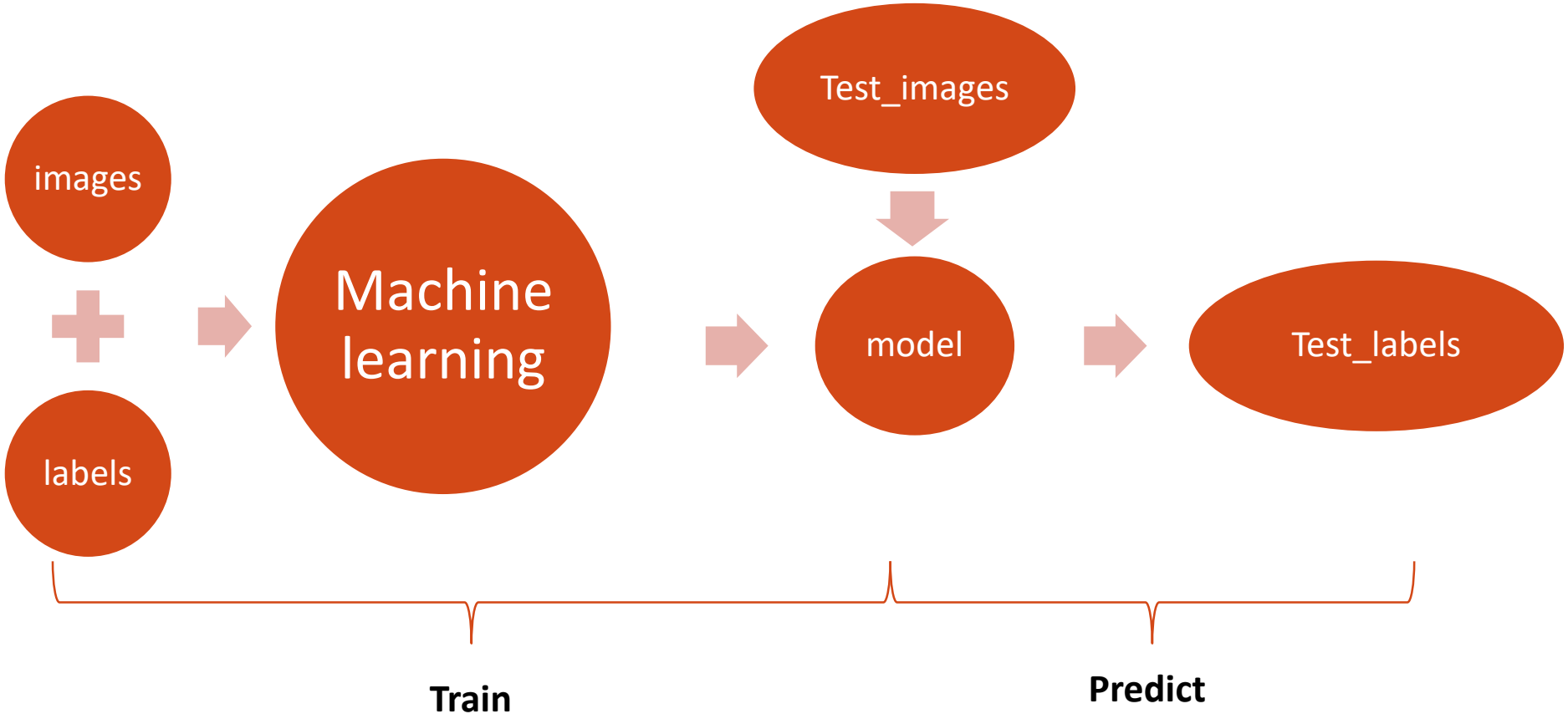
Traditional Approach: Hand-designed feature description and detection.

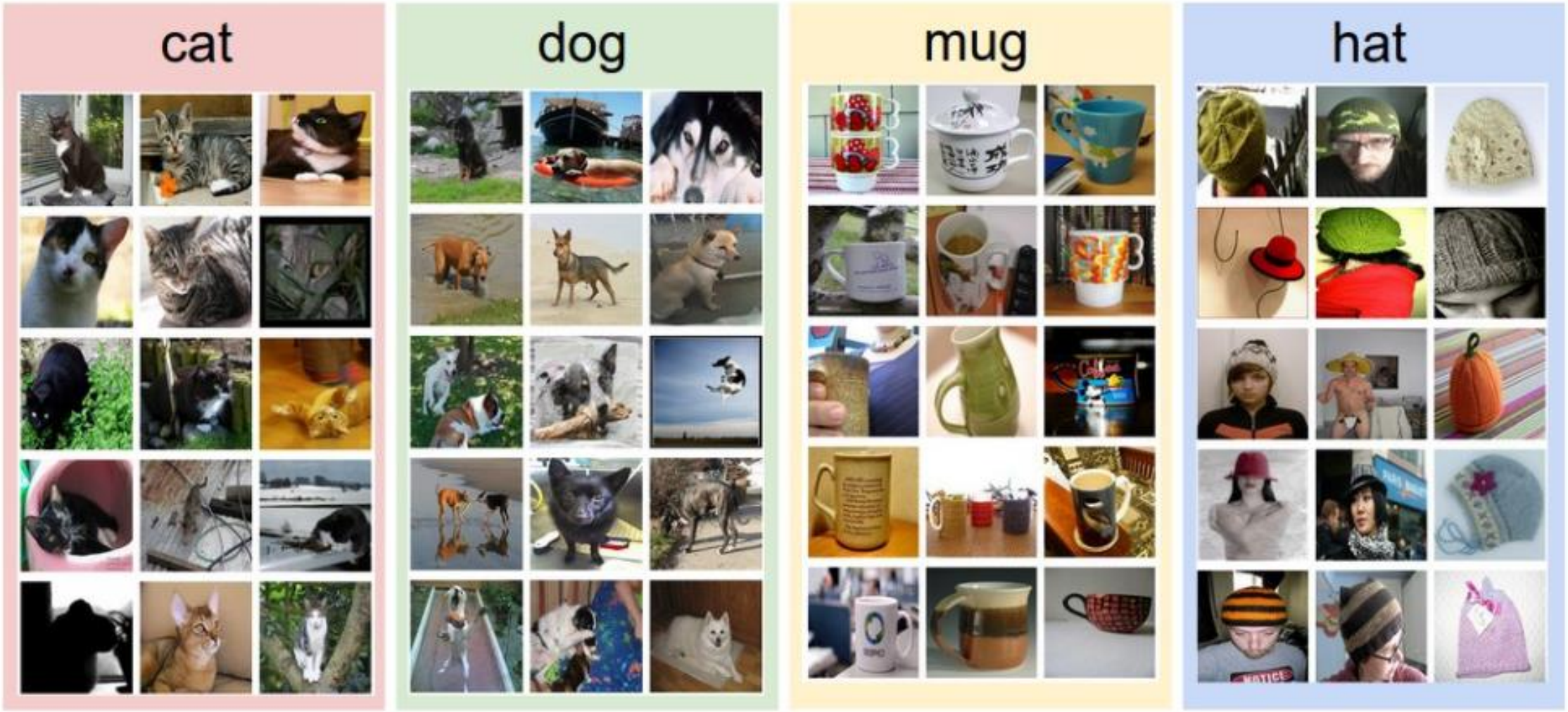


CS231n

Data-Driven Approach:

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images





An example training set for four visual categories. In practice we may have thousands of categories and hundreds of thousands of images for each category.

CS231n

The Importance of Image Classification

Foundation of :

1. Image detection
2. Image segmentation
3. Object tracking

...

Application:

1. Public security system: Fingerprint identification & Face recognition
2. Medical system: Assisted medical care
3. Commercial: Trademark management, Online shopping, Searching…

...

The Challenges in Image Classification Area

Challenges: Viewpoint Variation

CS231n

Challenges: Scale Variation

CS231n

Challenges: Deformation




CS231n

Challenges: Occlusion

CS231n

Challenges: Illumination condition

CS231n

Challenges: Intra-class variation

CS231n

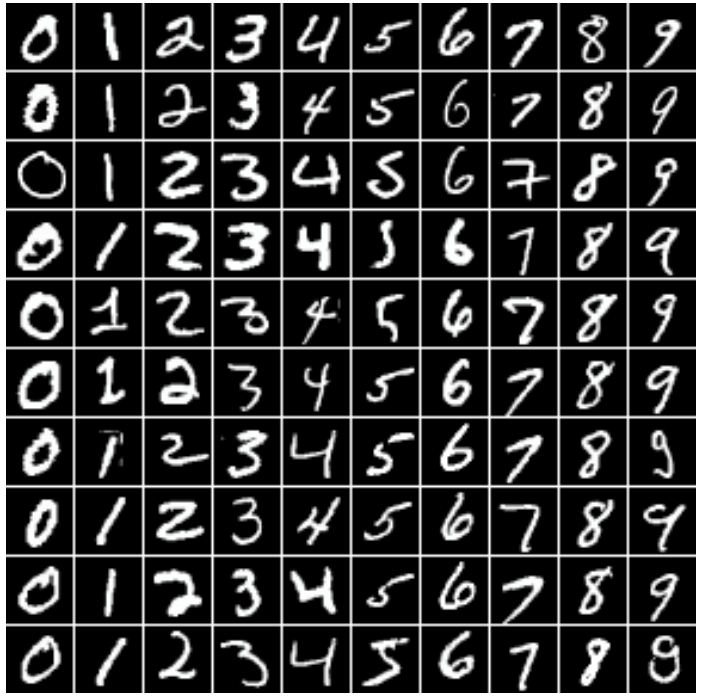
Challenges: Background Clutter

CS231n

The Introduction to Some Benchmark Datasets

MNIST：

Number of training images： 60000
Number of testing images： 10000
Number of categories: 10

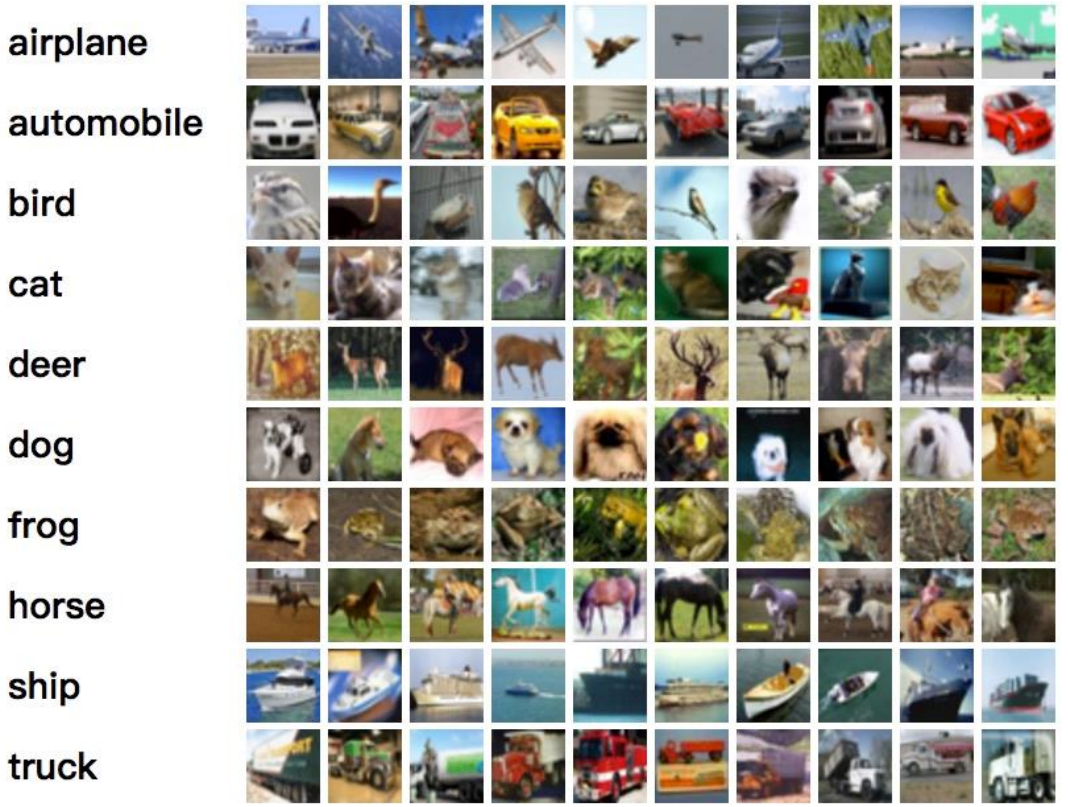


http://yann.lecun.com/exdb/mnist/

CIFAR-10：

Number of training images：50000
Number of testing images：10000
Number of categories: 10

CIFAR-100:
Number of training images：50000
Number of testing images：10000
Number of categories: 100



http://www.cs.toronto.edu/~kriz/cifar.html

Caltech-101:

Number of images：9144
Number of categories: 102

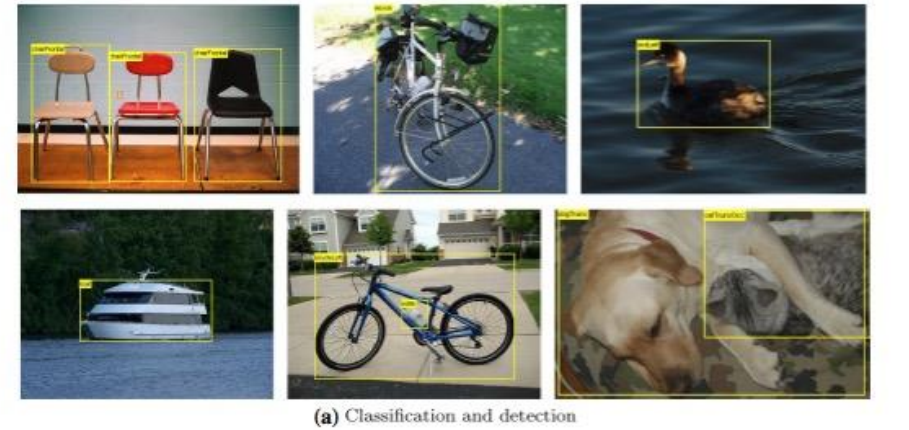
Caltech-256:

Number of images：30607
Number of categories: 257

http://www.vision.caltech.edu

PASCAL VOC:

Number of images：11000
Number of categories: 20
Number of Object instances: 27000
Number of segmentation: 7000



(a) Classification and detection

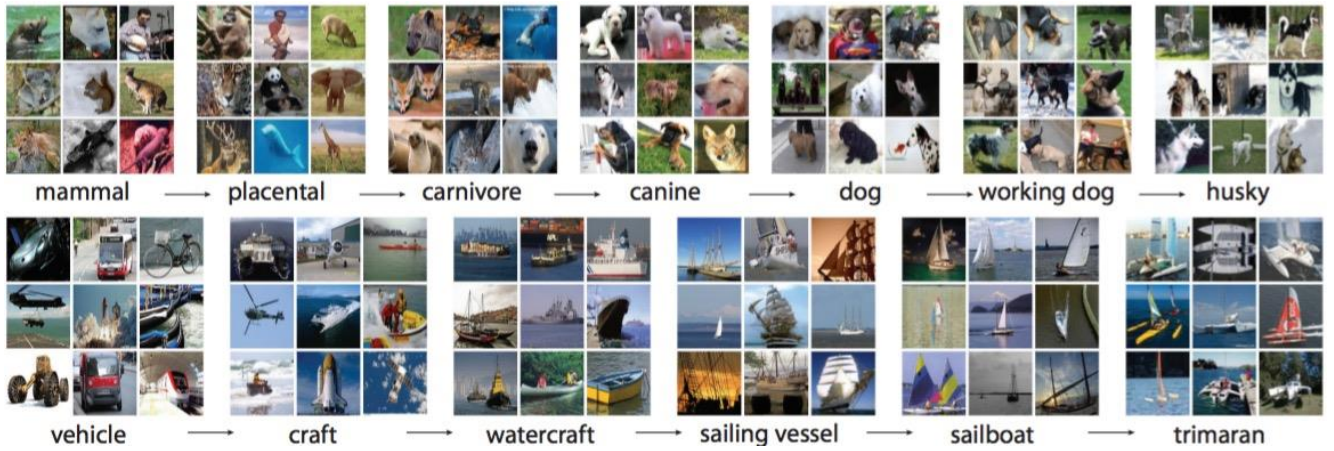
(b) Segmentation

(c) Action classification

(d) Person layout

http://host.robots.ox.ac.uk/pascal/VOC/

ImageNet-1000:

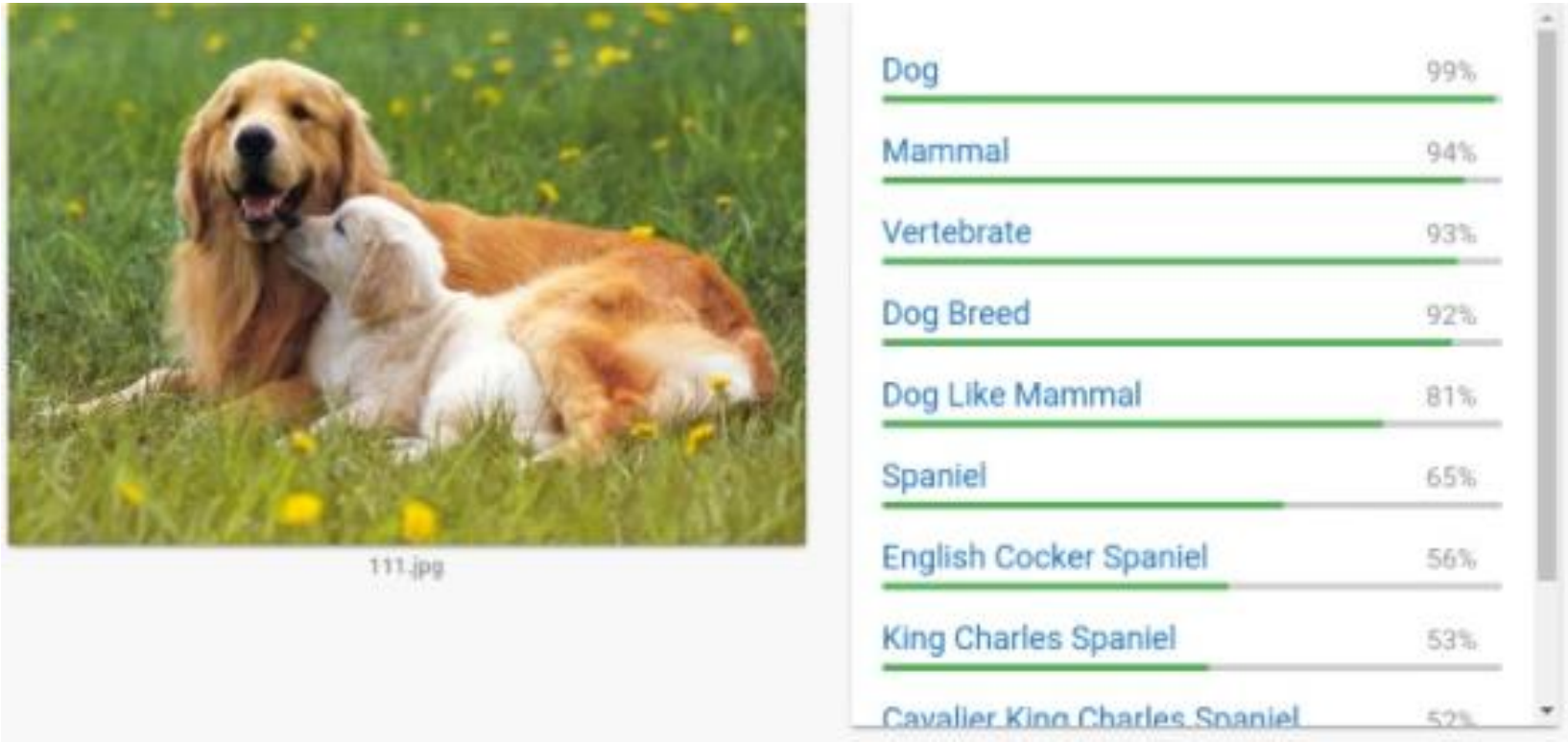


Number of training images：1.3M
Number of testing images：100K
Number of categories: 1000

http://image-net.org/

Google Open Image dataset:

9 million images
Over 6000 categories



https://github.com/openimages/dataset

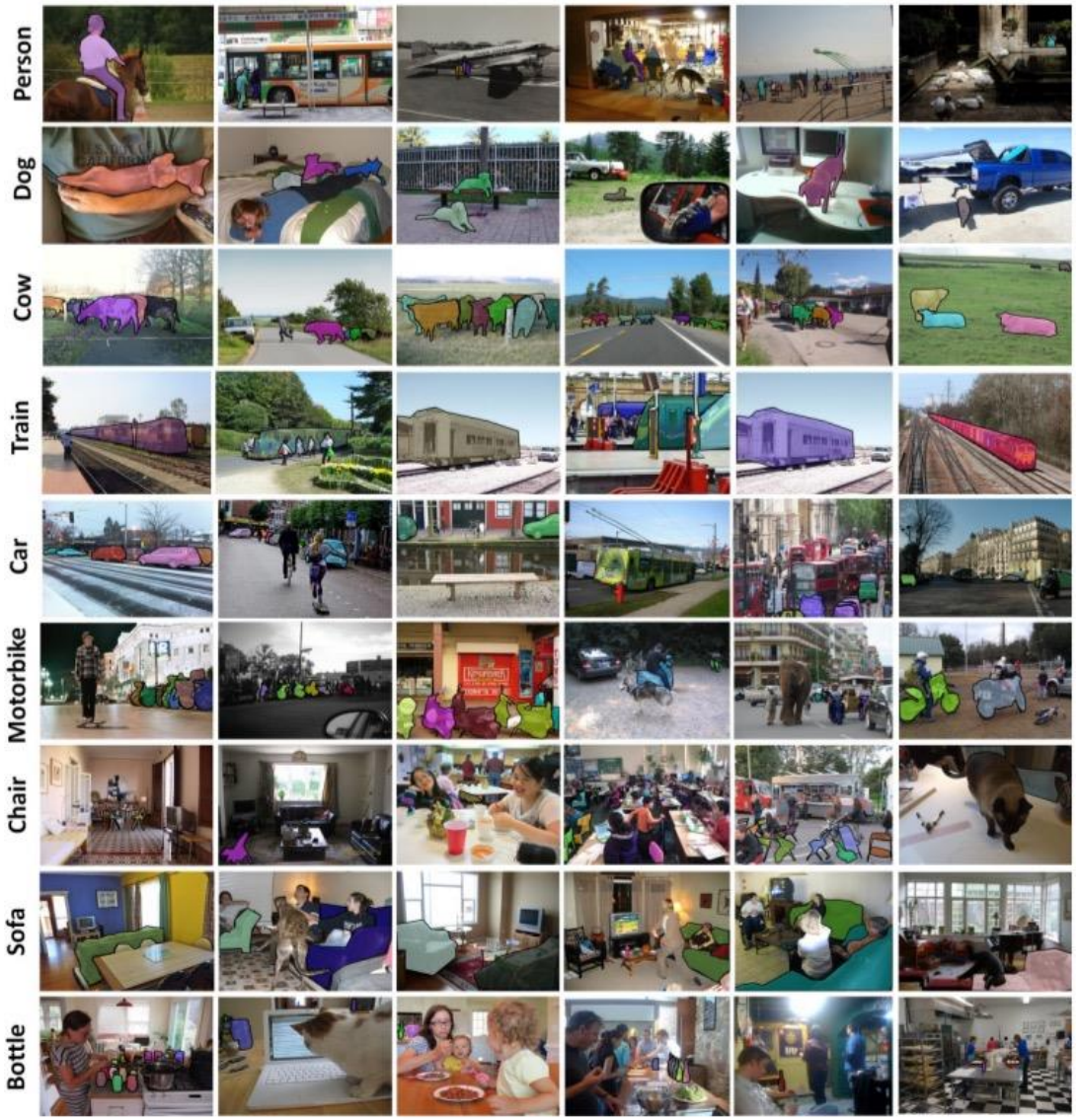
MS COCO:

Number of training images: 165482

Number of validation images: 81208

Number of testing images: 81434

Number of categories: 102

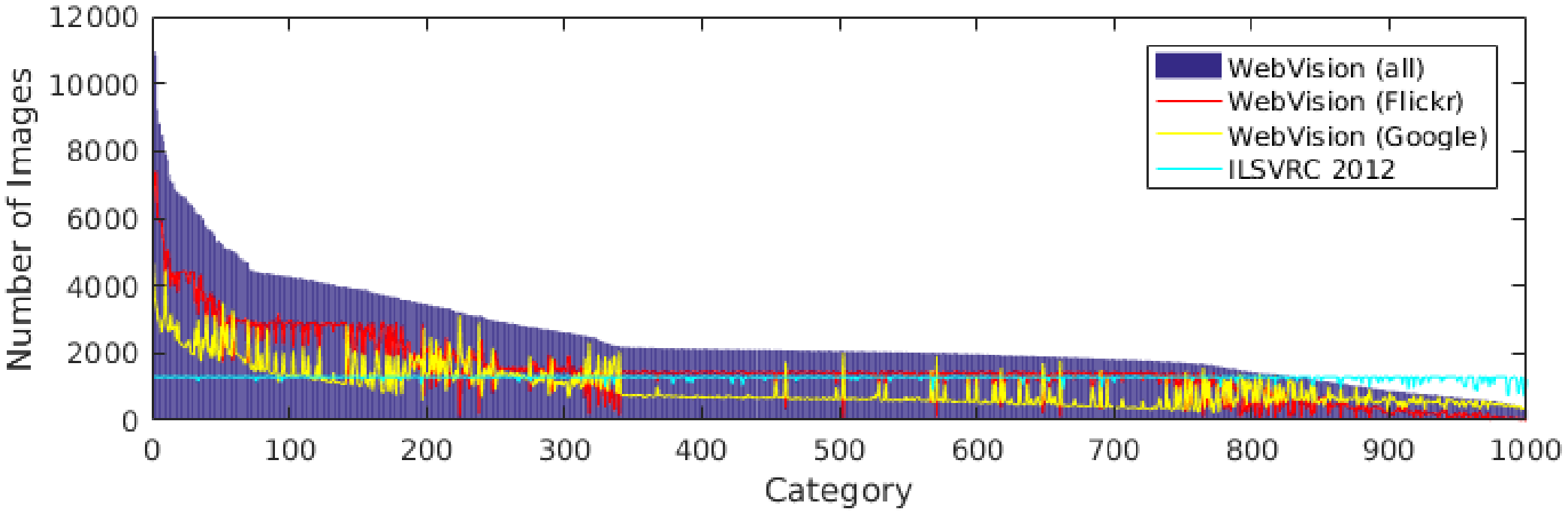


http://cocodataset.org/

WebVision

http://www.vision.ee.ethz.ch/webvision/

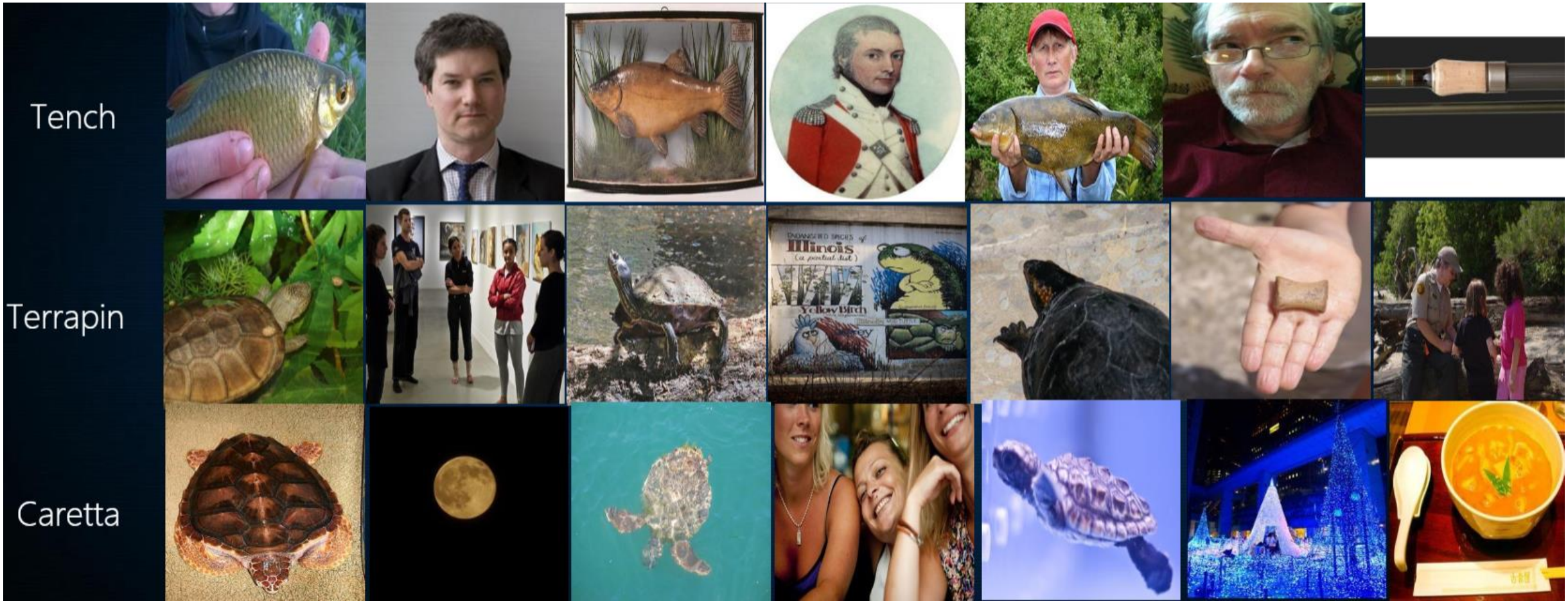
1. The WebVision dataset contains more than **2.4 million** of images crawled from the Flickr website and Google Images search.
2. A validation set contains 50,000 images (50 images per category) is provided to facilitate the algorithmic development.



The number of images per category for our dataset is shown, which varies from several hundreds to more than 10,000.

http://www.vision.ee.ethz.ch/webvision/

Label noise：



http://www.vision.ee.ethz.ch/webvision/

2 Image Classifiers

Before 2012: Traditional Image Classification Methods

Before 2012: Traditional Image Classification Methods

Credited to Prof. Songchun Zhu

- **Image processing：** Image enhancement, Image restoration, Image segmentation
- **Feature extraction：** SIFT, SURF, HOG, LBP, FAST, LoG, DoG
- **Feature representation：** Fisher Vector, Vector quantization, Soft quantification, FMM, LCC
- **Feature selection：** Search measurement based, Evaluation criteria based
- **Dimensionality reduction：** PCA, LDA, LLE, Feature hashing
- **Classifier：** SVM, K-nearest, Random forest, adaboost

Disadvantages:

- Hand-designed features
  - High costs
  - Subjective
  - Poor effectiveness
- Artificial integrated system

First classifier: Nearest Neighbor

1. Train:

   Just memorize all of the training data.

2. Predict:

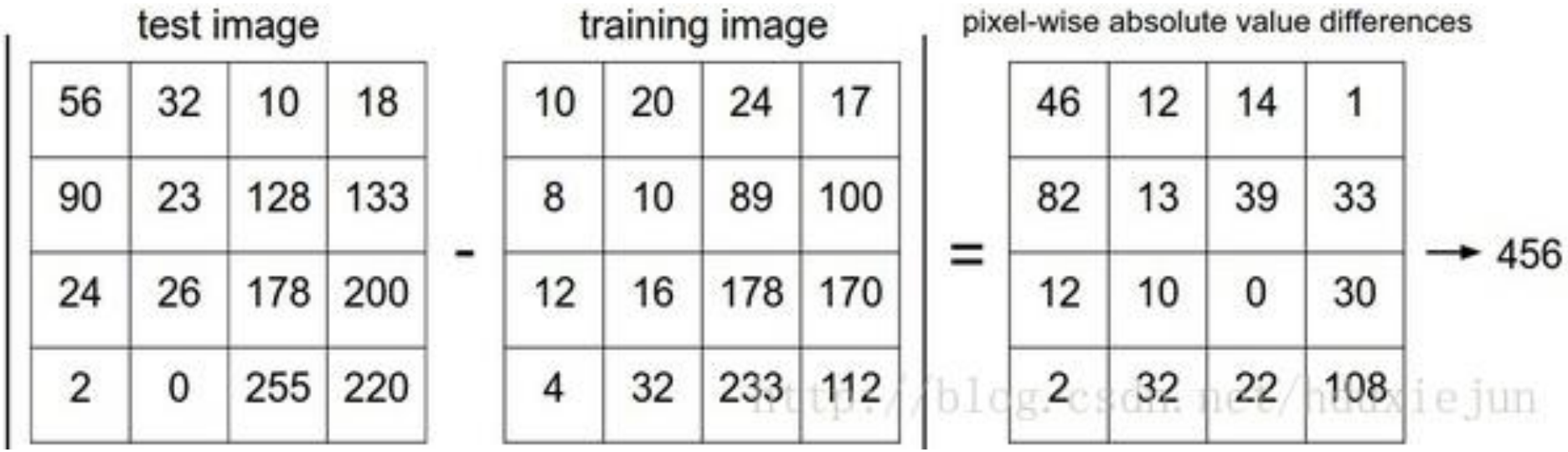
   Find the most similar image in the training data to that new image.

How to calculate the distance: L1 distance: sometimes called the Manhattan distance.

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

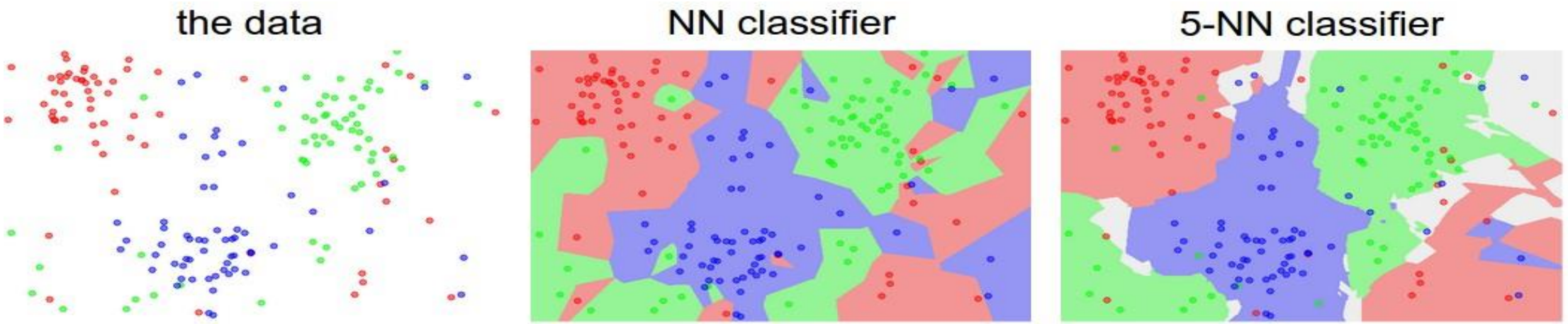
CS231n

For example(suppose that our test image is maybe just a tiny four by four image of pixel values):



CS231n

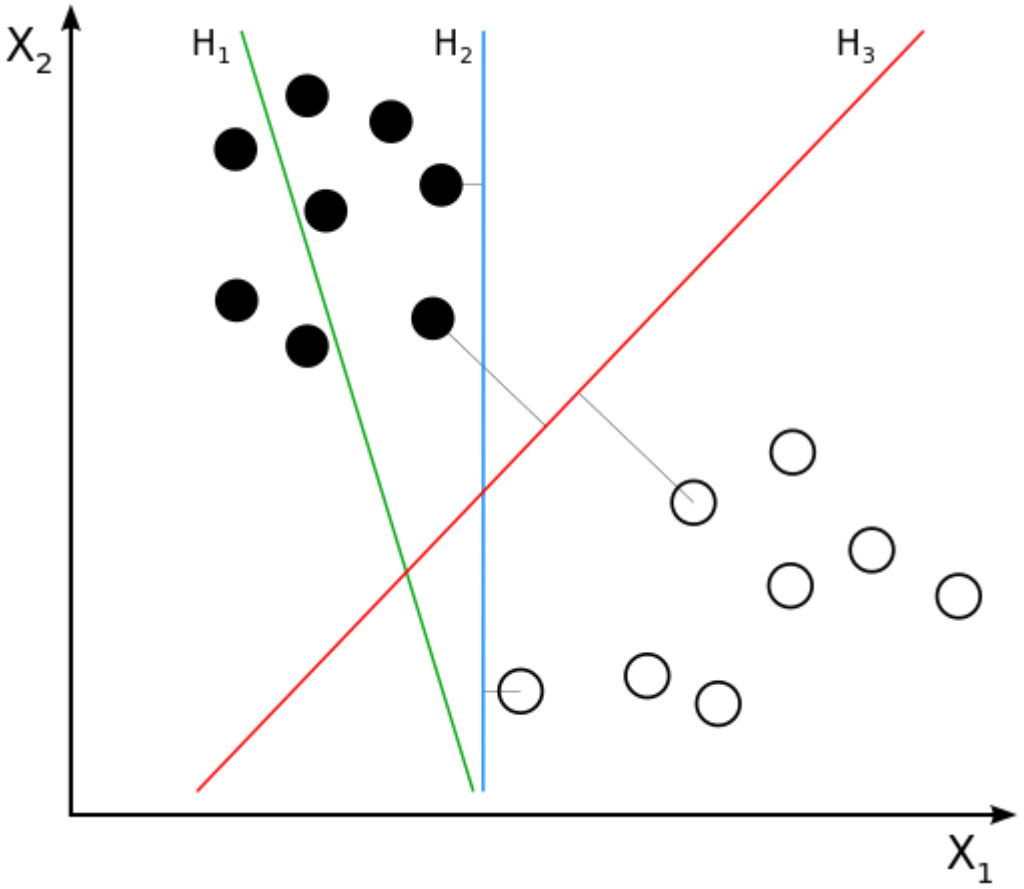
How does the Nearest Neighbors algorithm operate in practice?
Here we use the picture that we call the decision regions of the classifiers.



CS231n

SVM

Proposed: 1992



wikimedia

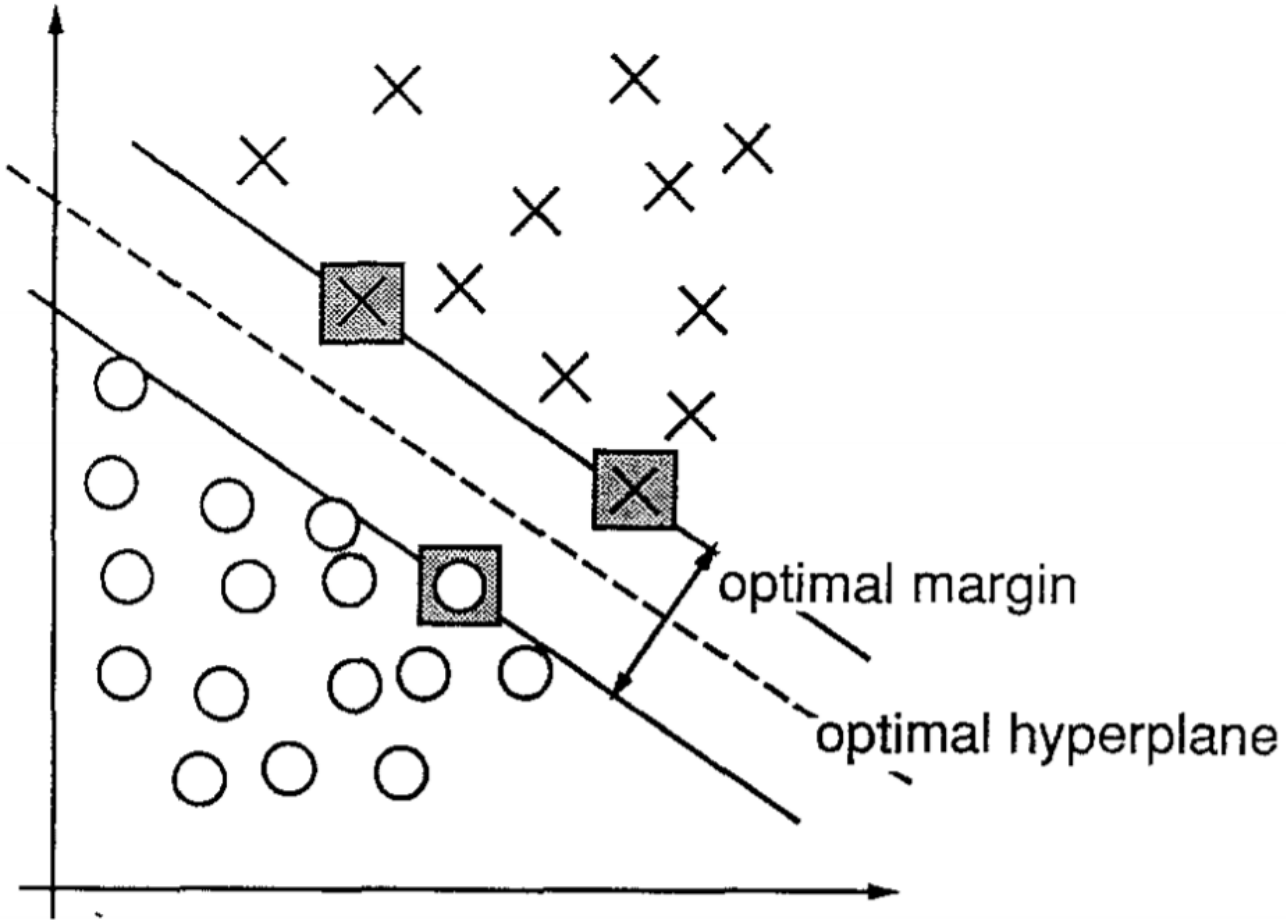
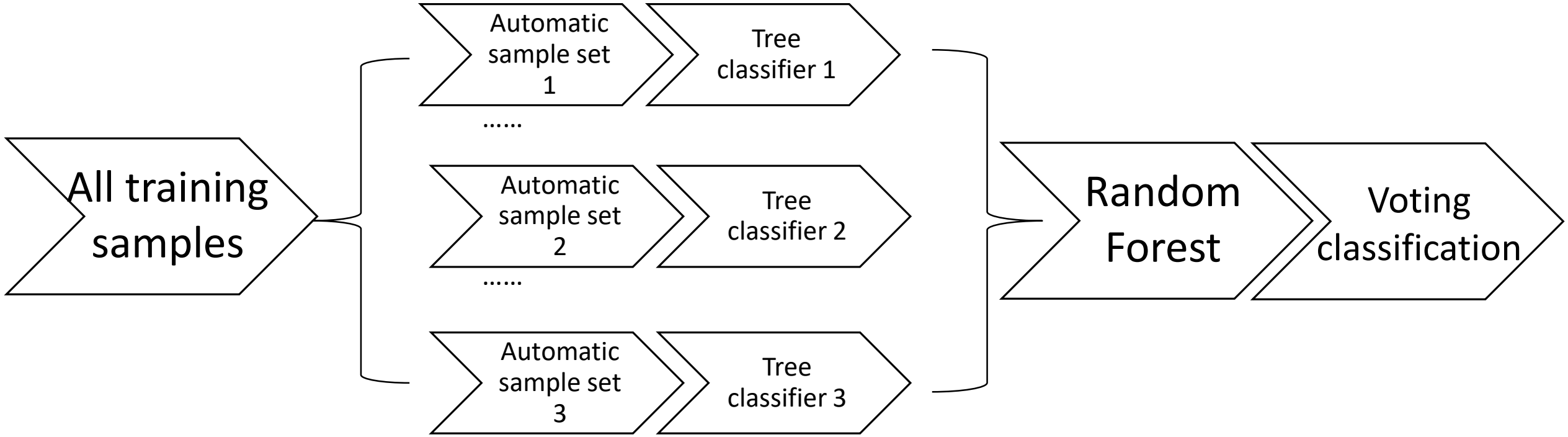


Figure 2. An example of a separable problem in a 2 dimensional space. The support vectors, marked with grey squares, define the margin of largest separation between the two classes.

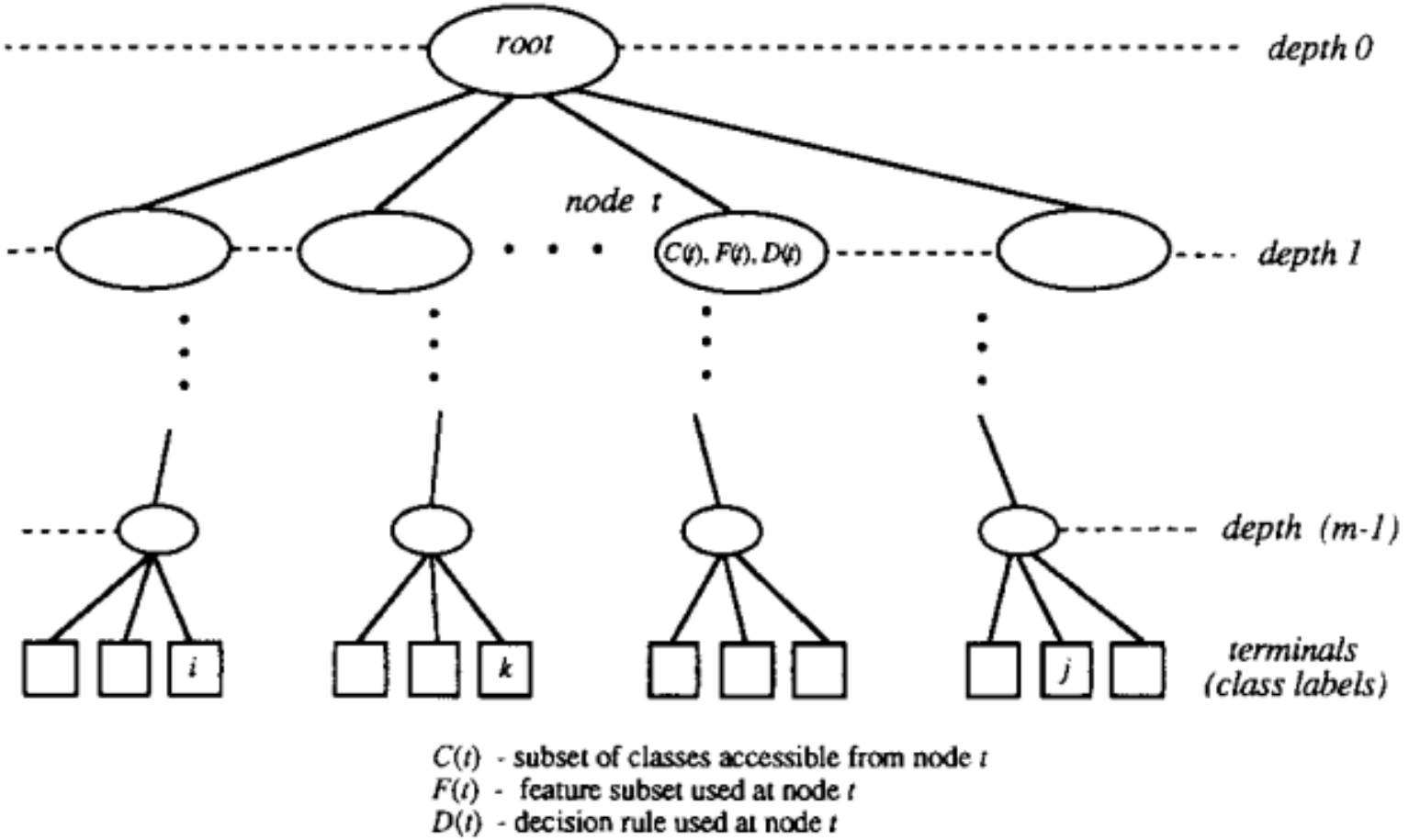
Cortes C, Vapnik V. Support-Vector Networks[J]. Machine Learning, 1995, 20(3):273-297.

Random Forest



L. Breiman. Random forests. Mach. Learning, 45(1):5–32, 2001

Decision Tree

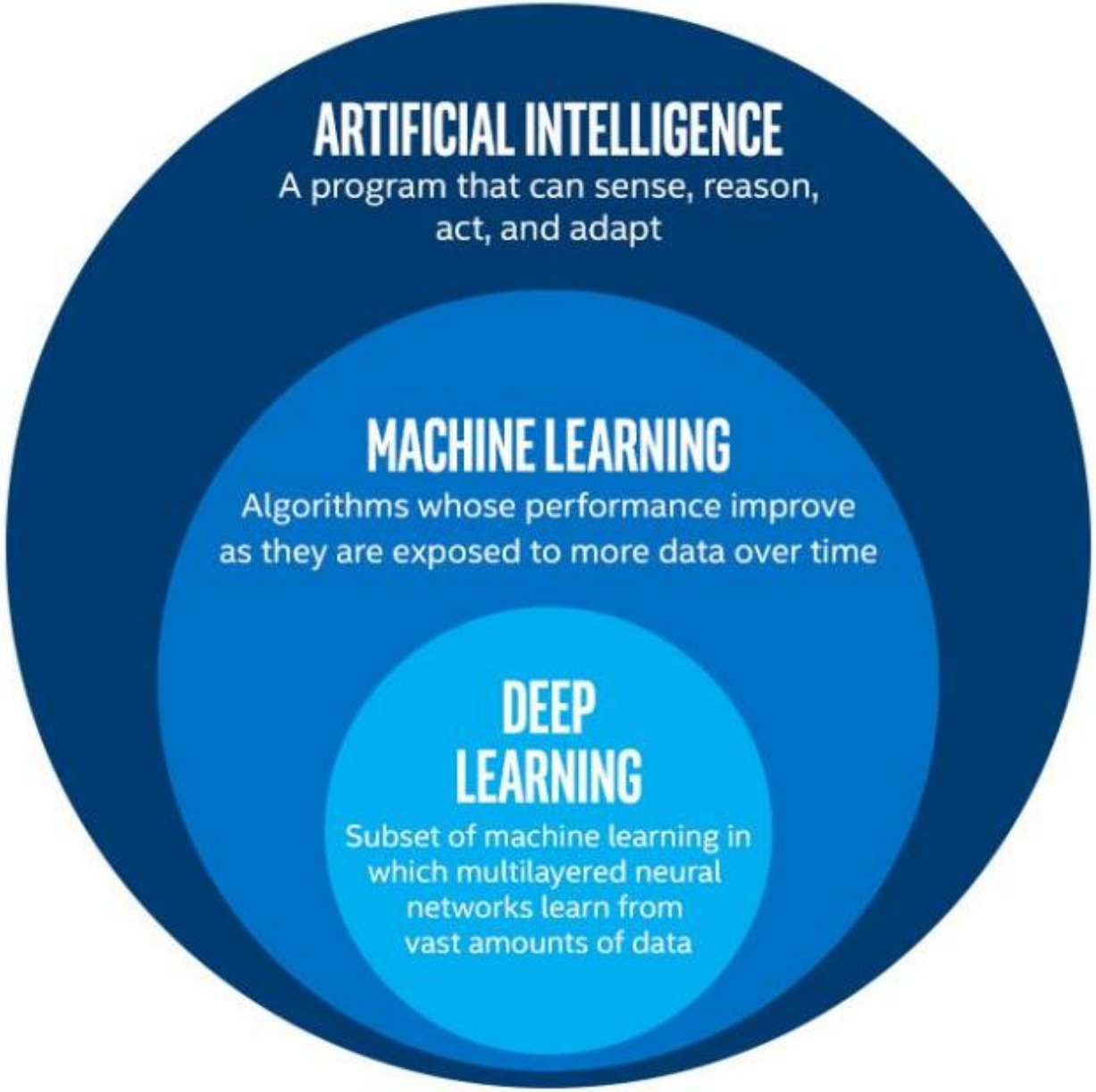


$C(t)$ - subset of classes accessible from node t
$F(t)$ - feature subset used at node t
$D(t)$ - decision rule used at node t

Safavian S R, Landgrebe D. A survey of decision tree classifier methodology[J]. IEEE transactions on systems, man, and cybernetics, 1991, 21(3): 660-674.

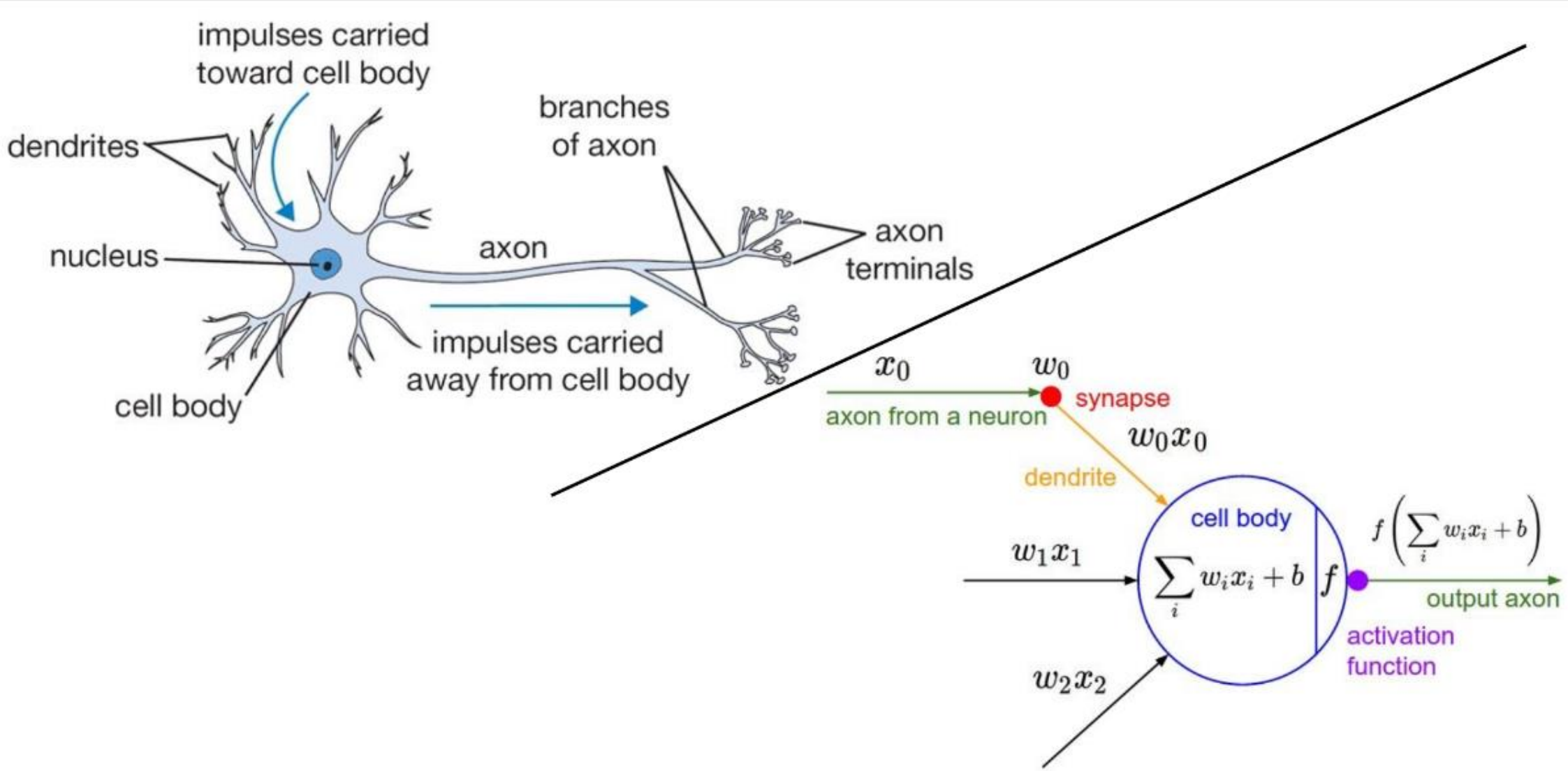
After 2012: Deep Learning Algorithms

What is Deep Learning?



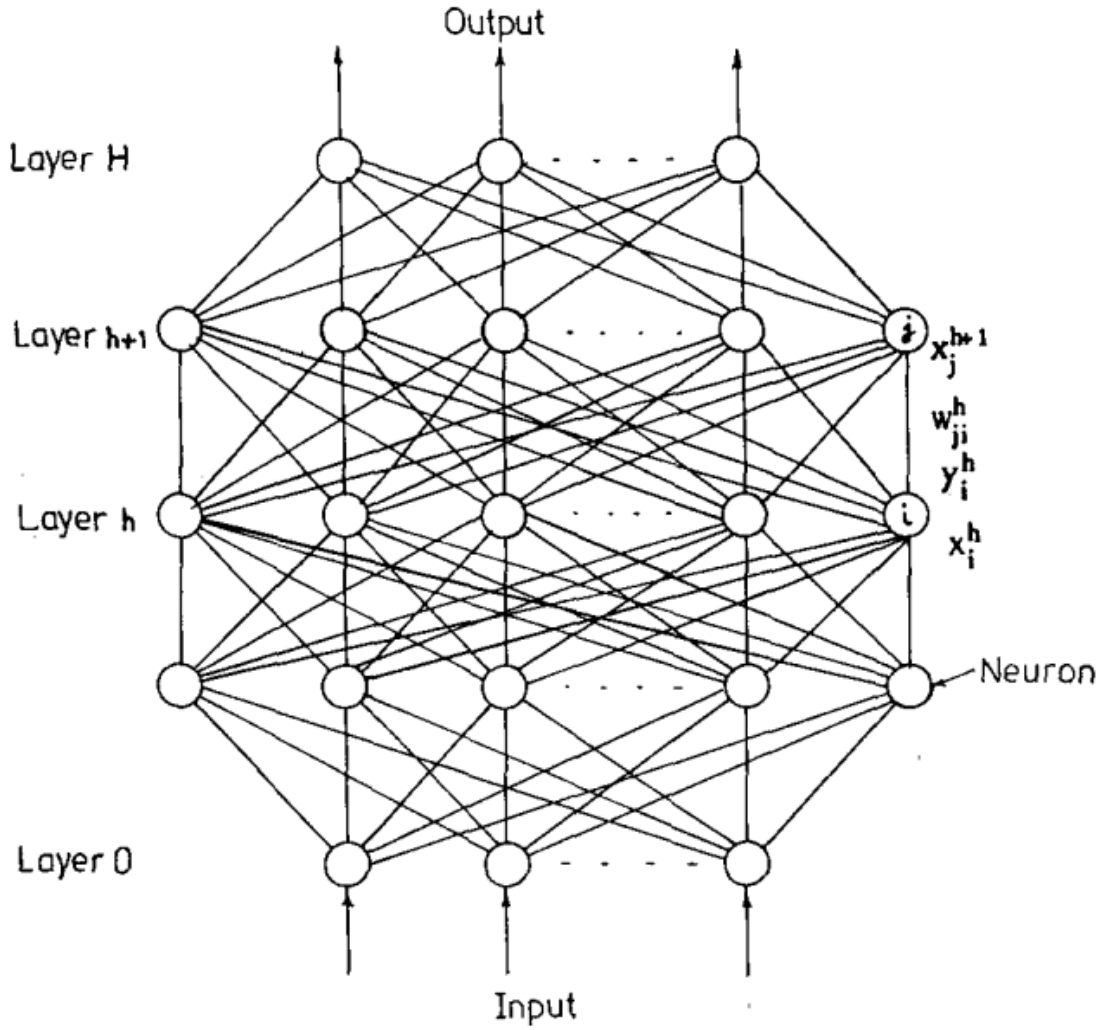
Intel

Basic Neural Network:



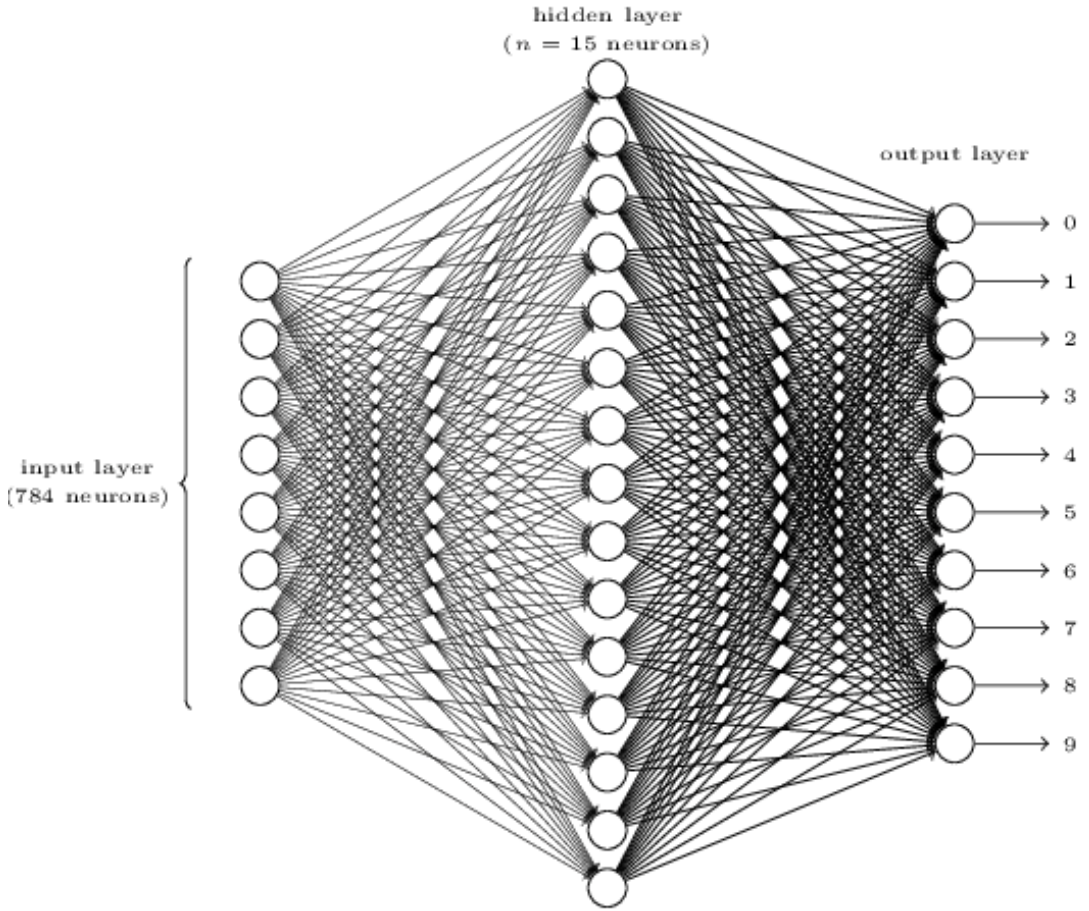
CS231n

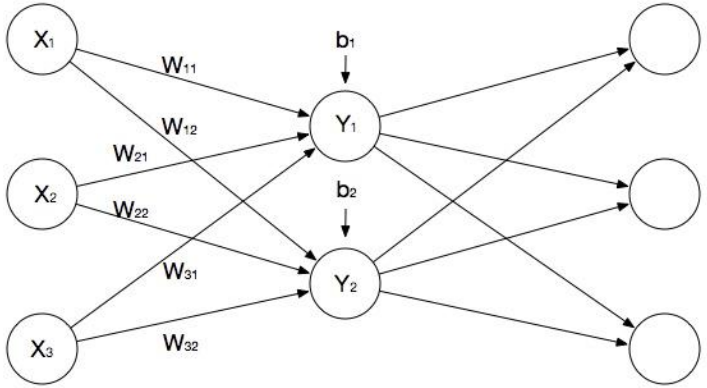
MLP



Pal S K, Mitra S. Multilayer perceptron, fuzzy sets, and classification[J]. IEEE Transactions on neural networks, 1992, 3(5): 683-697.

Multilayer Perceptron：



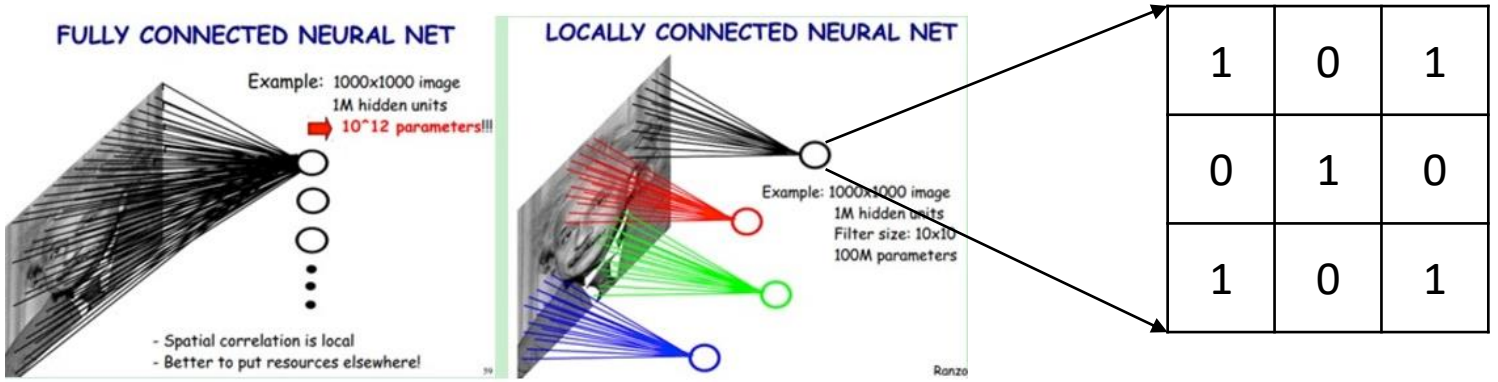


$$Forward:\ y = W^T x + b,\quad y \in R^{m\times 1}, x \in R^{n\times 1}, W \in R^{m\times n}$$

$$Backward:\ \frac{\partial L}{\partial x} = W \times \frac{\partial L}{\partial y}, \frac{\partial L}{\partial y} = x \times (\frac{\partial L}{\partial y})^T$$

Pal S K, Mitra S. Multilayer perceptron, fuzzy sets, and classification[J]. IEEE Transactions on neural networks, 1992, 3(5): 683-697.

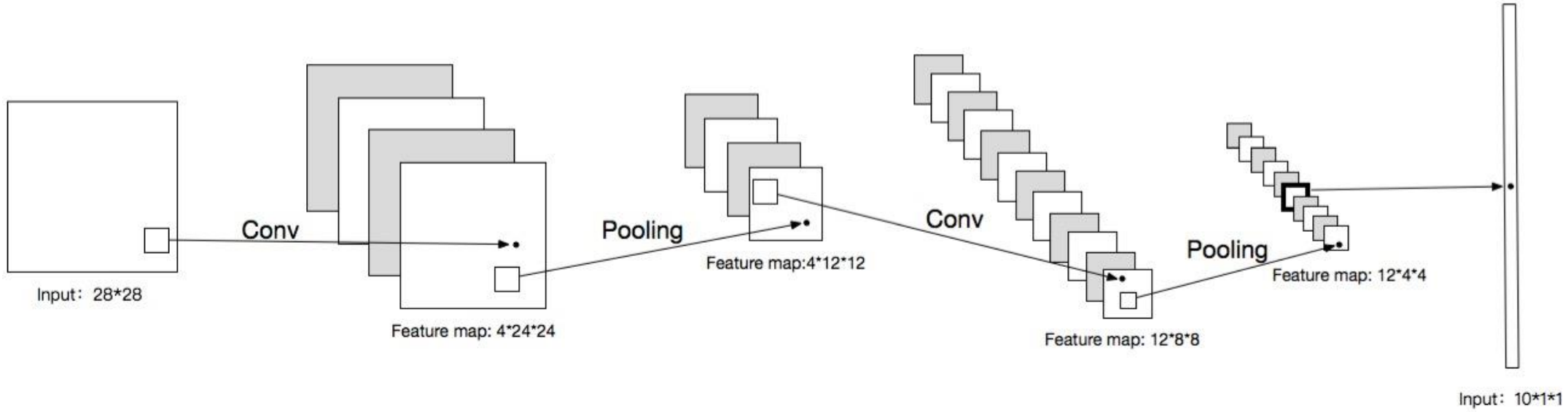
CNN



- Sparse Connectivity
- Shared Weights

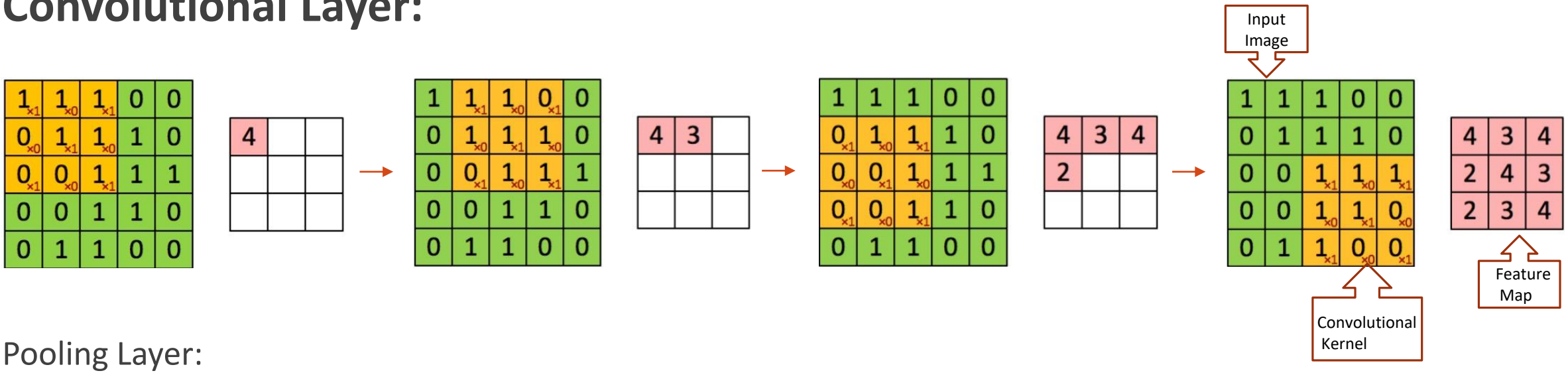
Large-Scale Visual Recognition With Deep Learning, Marc'Aurelio Ranzato, Sunday 23 June 2013.

CNN



Convolutional Neural Networks | Coursera

Convolutional Layer:

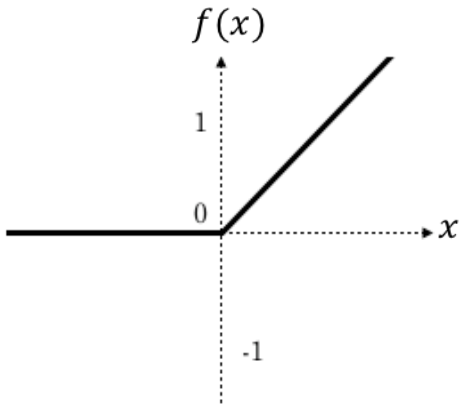


Pooling Layer:

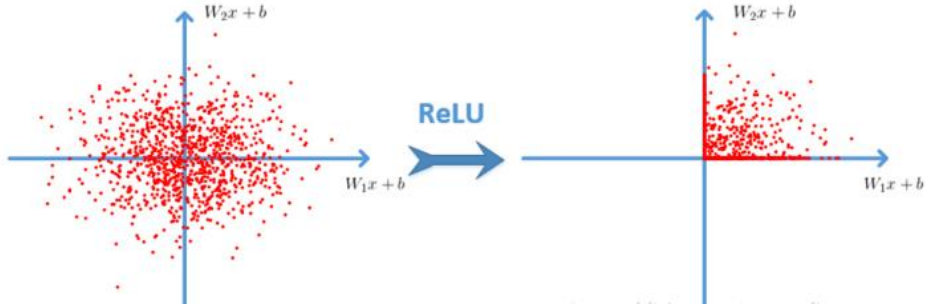
CS231n: Convolutional Neural Networks for Visual Recognition

CNN

ReLU:



$$f(x) = max(0, x)$$



Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]//Aistats. 2011, 15(106): 275.

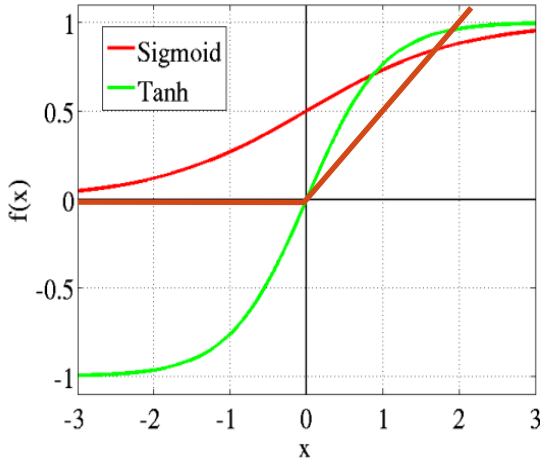
CNN

Advantages of ReLU：

1、Faster

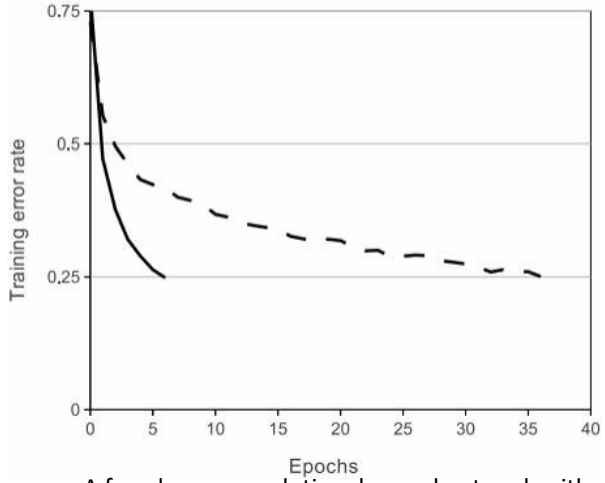
2、Simple Gradient

3、Sparsity



Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$

Tanh: $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$



A four-layer convolutional neural network with ReLUs (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons (dashed line).

Deep Learning: Revolution

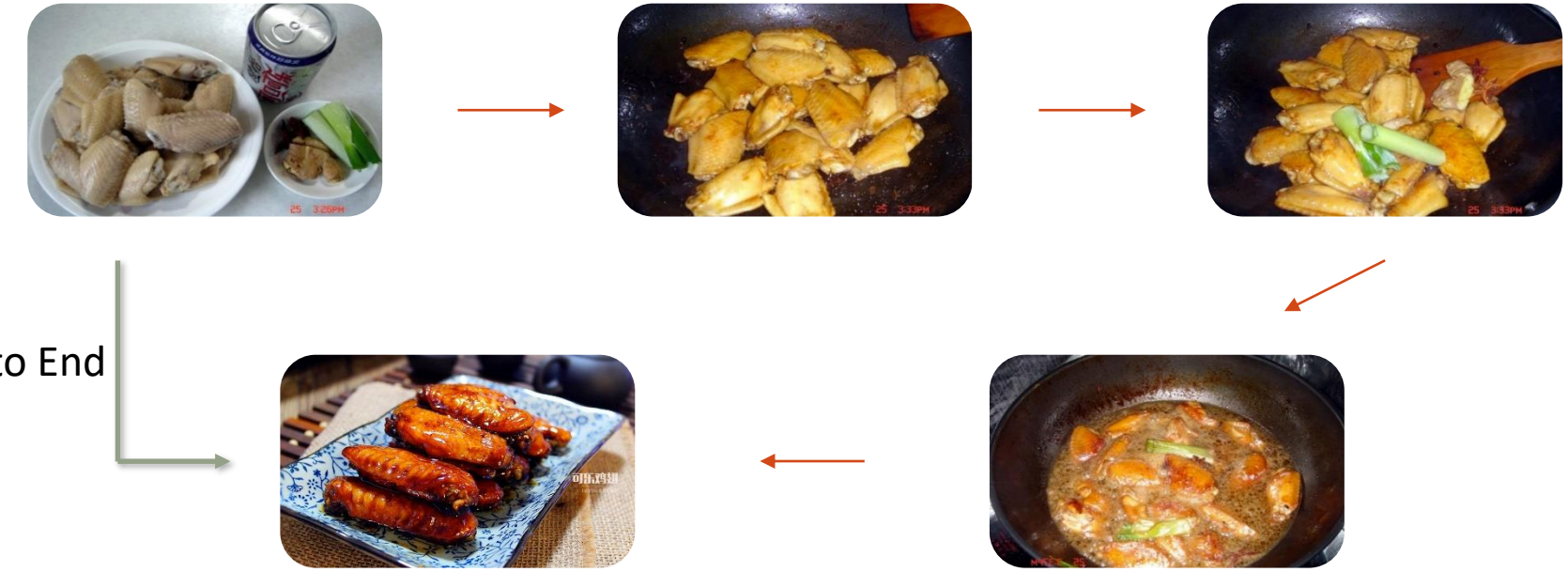
ILSVRC：

- Good classification effect
- Transportability
- End to End
- Hierarchical features

Year	Top-5 error	Methods
2010 Champion	28.20%	HOG + LBP + SVM
2011 Champion	25.70%	FV + SVM
2012 Champion	16.42%	DCNN: AlexNet (8 layers)
2013 Champion	11.74%	DCNN: Network visualization (AlexNet based)
2014 Champion	6.66%	DCNN: GoogLeNet (22 layers + Inception)
2014 Second	7.32%	DCNN: VGGNet (19 layers)
2015 Champion	3.6 %	DCNN: ResNet (152 layers)

Deep Learning | Coursera

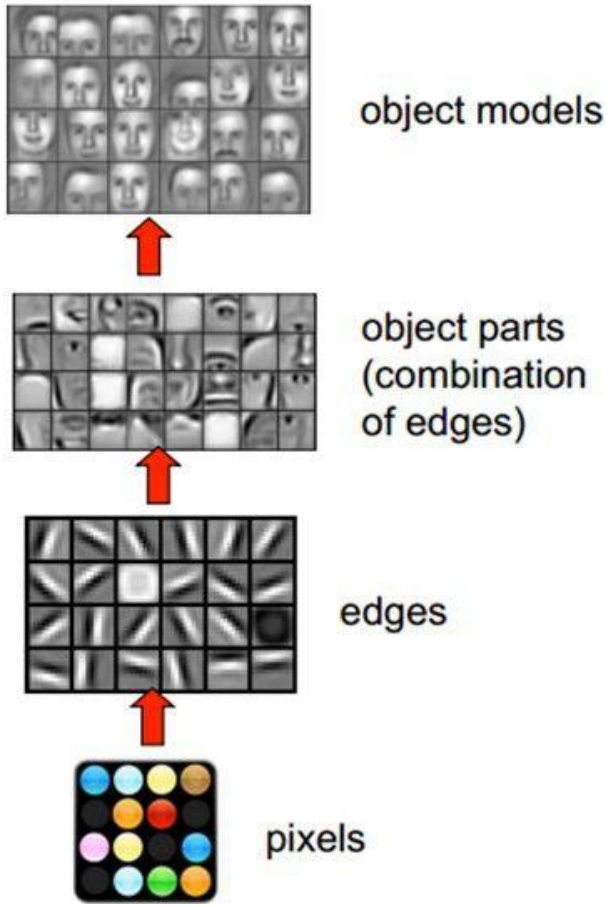
End - to - End:

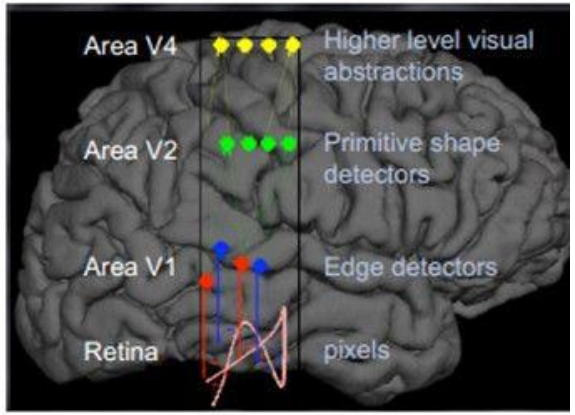


Deep Learning | Coursera

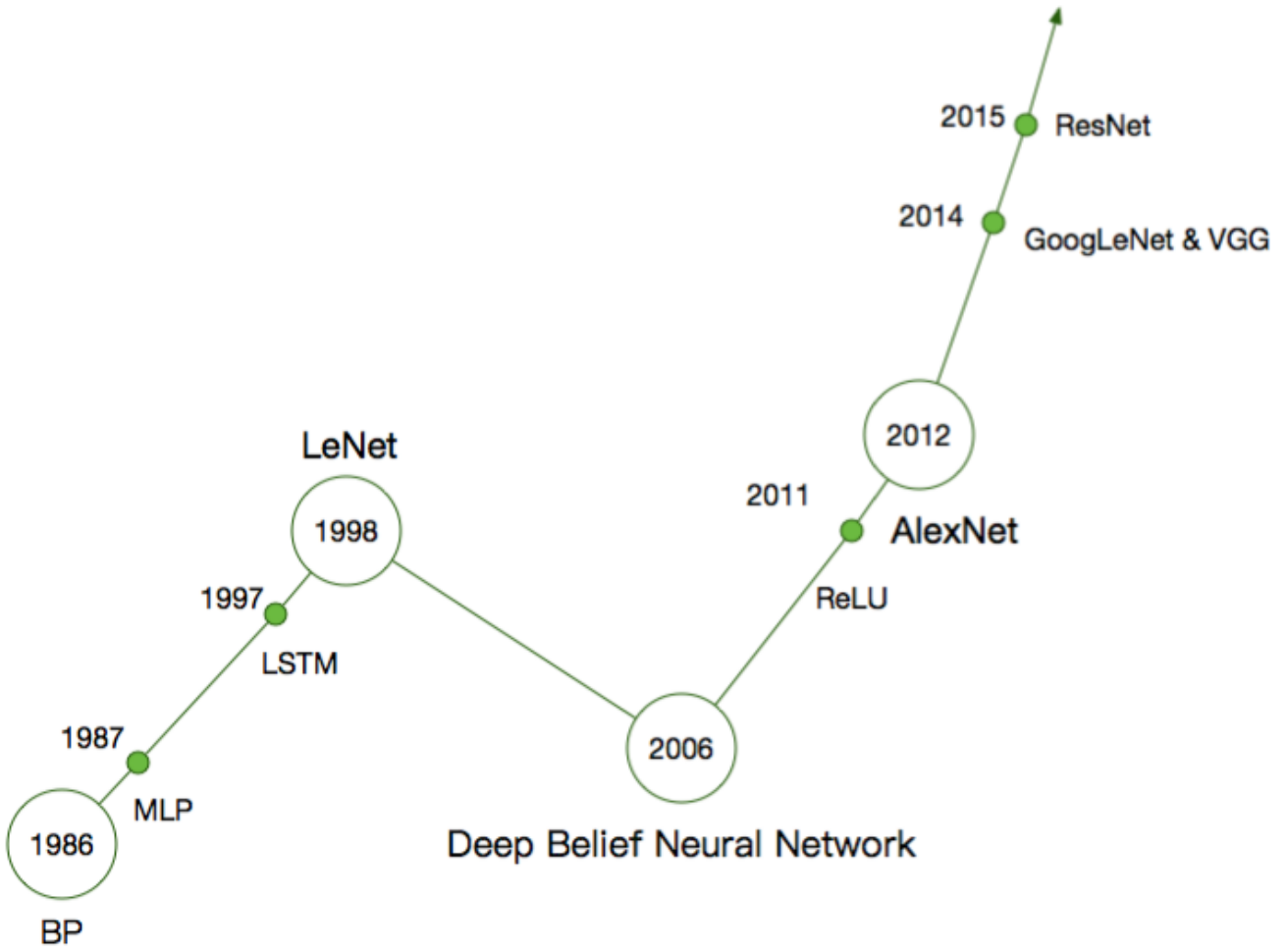
Hierarchical features：

Pixel => Edge => Texton => Motif => Part => Object





Deep Learning | Coursera

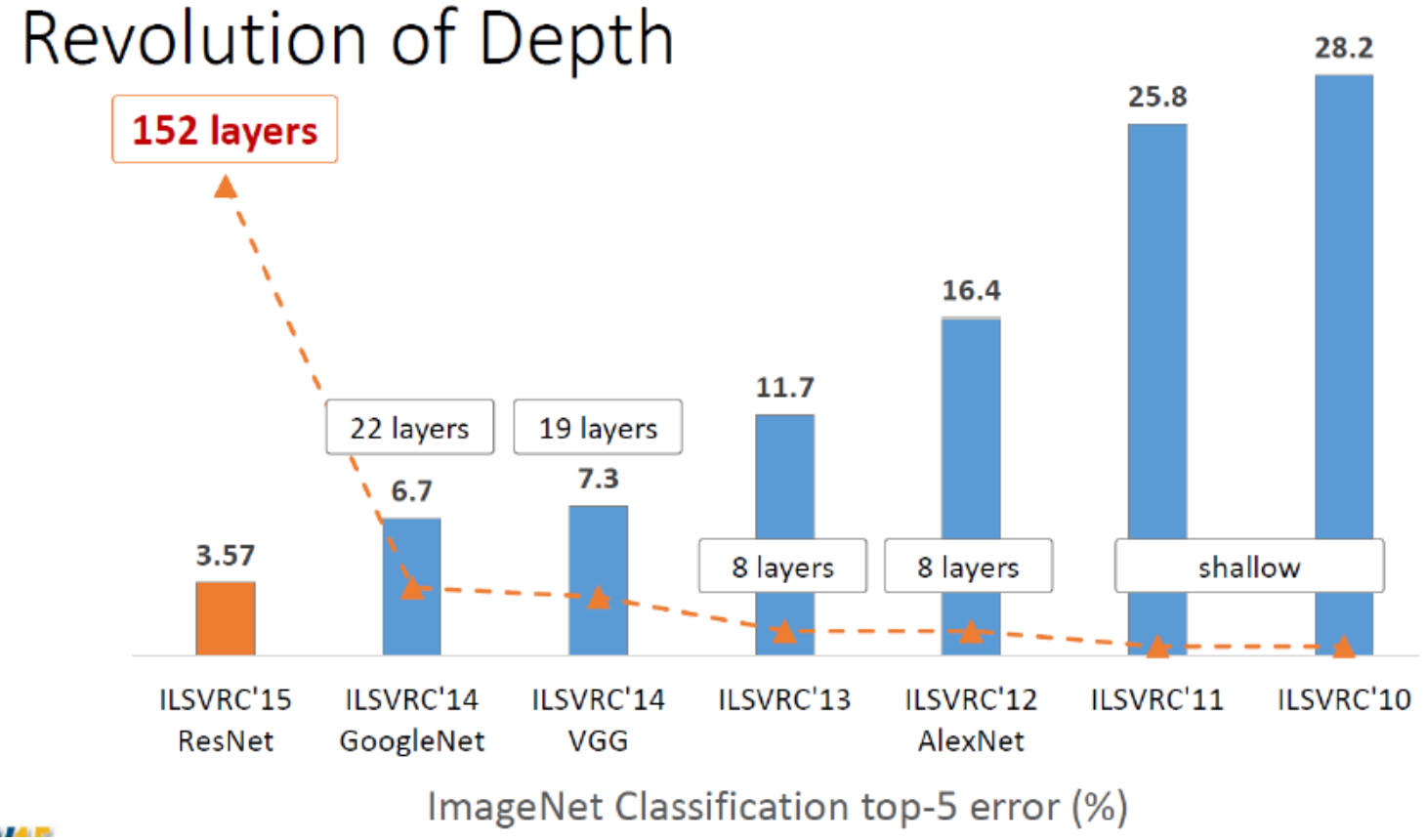


Credited to Prof. Meina Kan, PHD Student Xin Liu and Shuzhe Wu

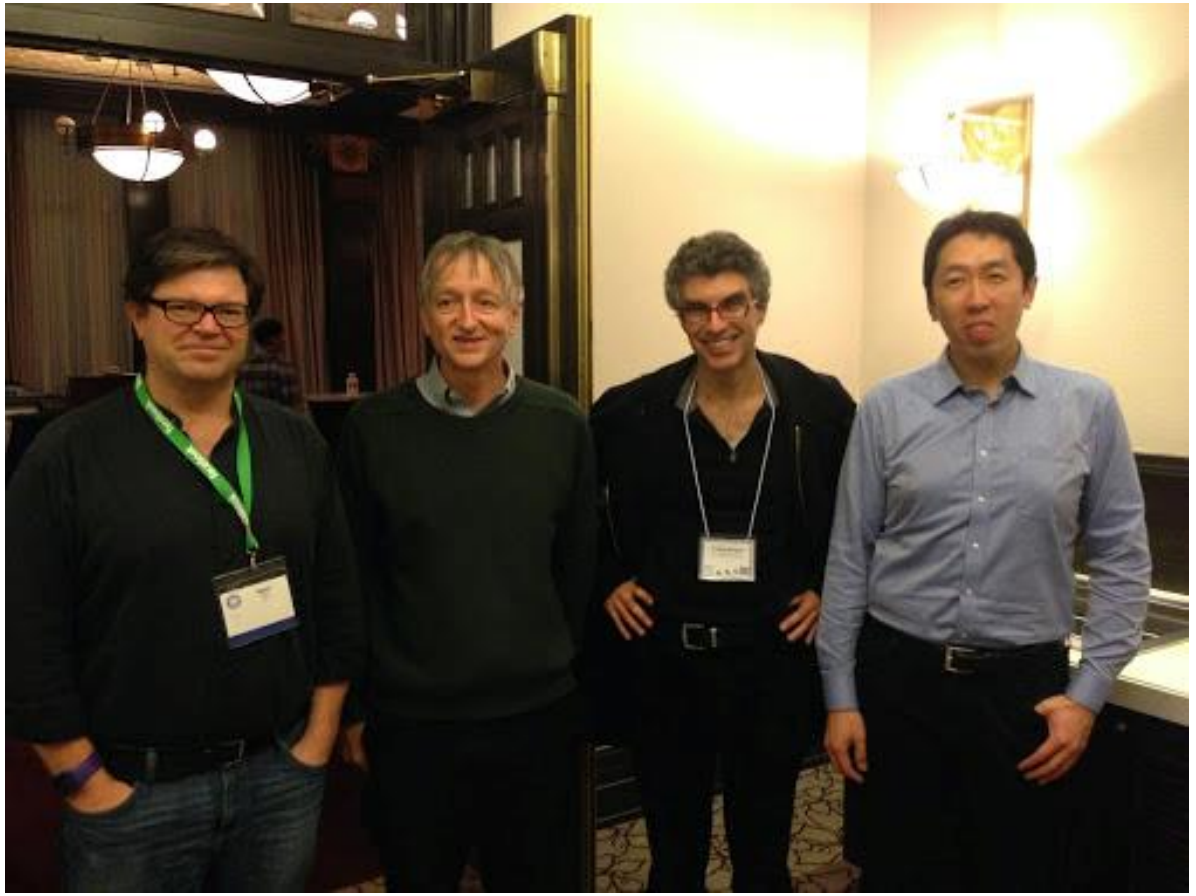
The evolution of networks:

Networks: AlexNet → VGG → GoogLeNet → ResNet

Depth: 8 → 19 → 22 → 152

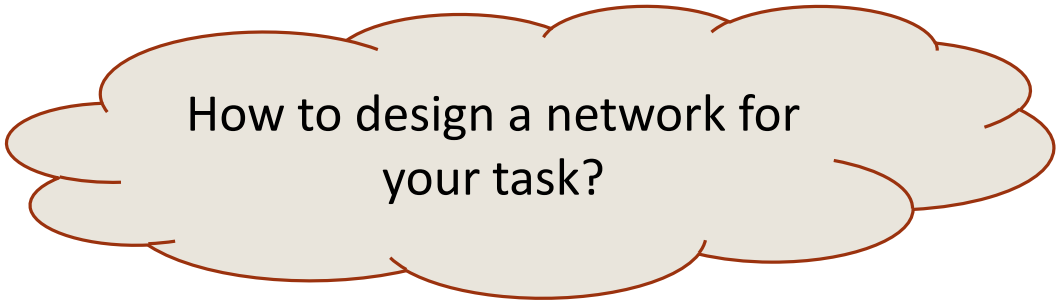


Credited to Prof. Shiguang Shan with modified



From the left：

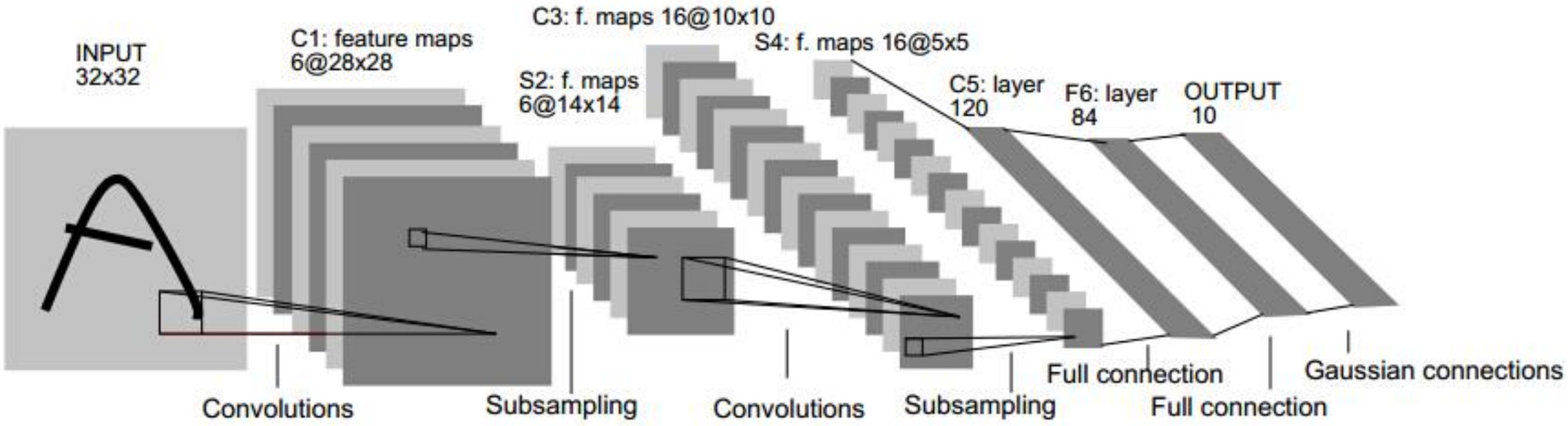
LeCun
Hinton
Bengio
NG



Look for the already open implementation and pretrained model for fine-tuning.

LeNet-5

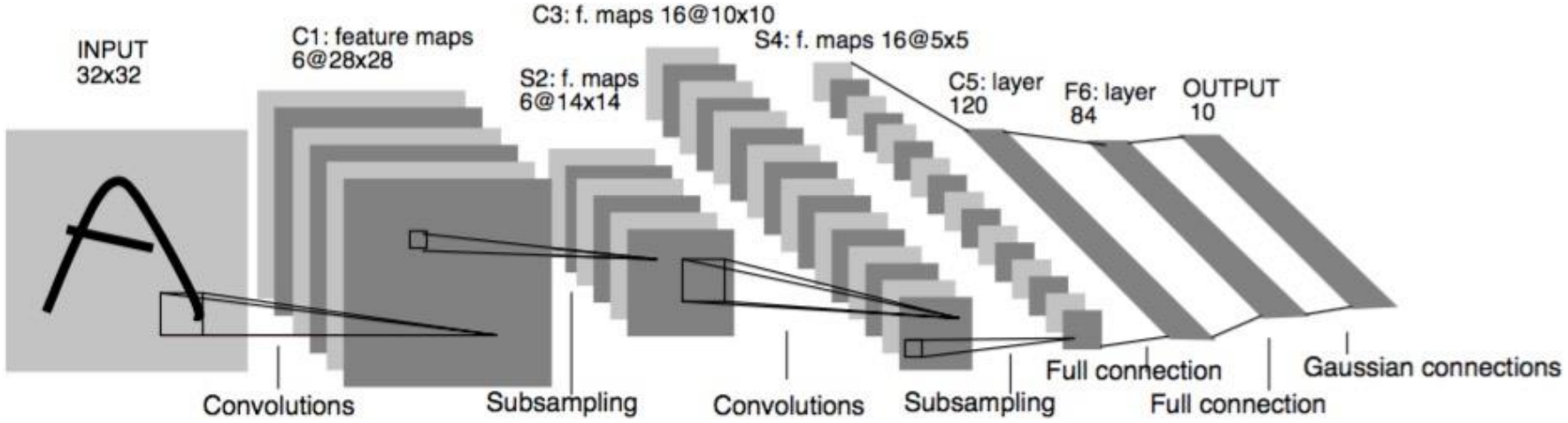
This is a classic CNN architecture, mainly for handwriting recognition, and also a network that needs to be learned to learn.



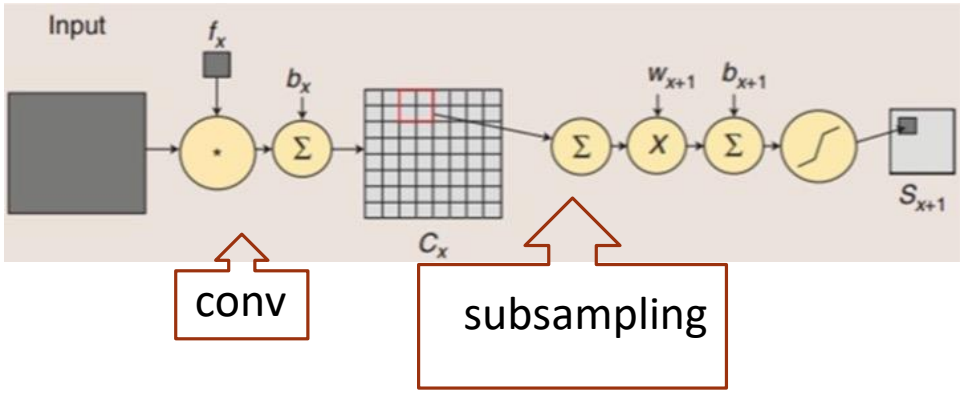
LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

LeNet-5:

Network Architecture:



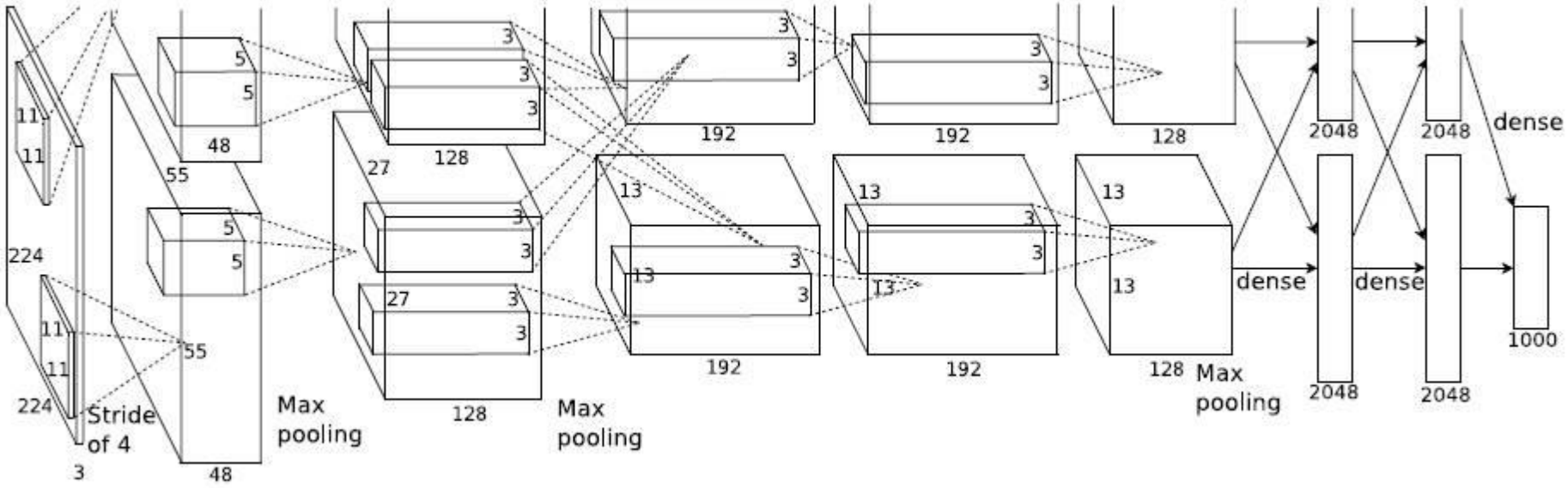
Process of Convolutions & Subsumpling:



LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

AlexNet

ILSVRC 2012's champion network. The basic network architecture: conv1 (96) (256) - > pool1 - > conv2 - > pool2 - > conv3 - > conv4 (384) (384) (256) - > conv5 - > pool5 - > FC6 - > fc7 (4096) (4096) (1000) - > softmax - > fc8.



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

Novel：

1. Data Augmentation
2. ReLU Nonlinearity
3. Local Response Normalization
4. Overlapping Pooling
5. Dropout
6. Training on Multiple GPUs

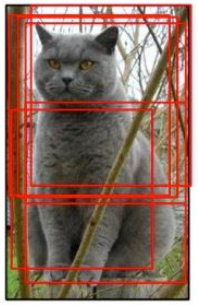
Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

CNN: AlexNet

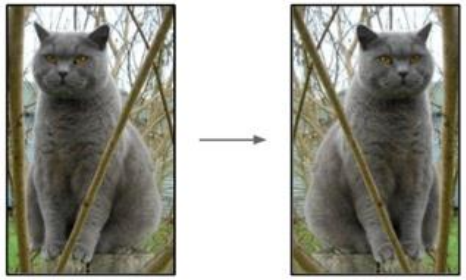
Data Augmentation：

1. Image crop and horizontal reflections.

2. Alter the intensities of the RGB channels in training images.



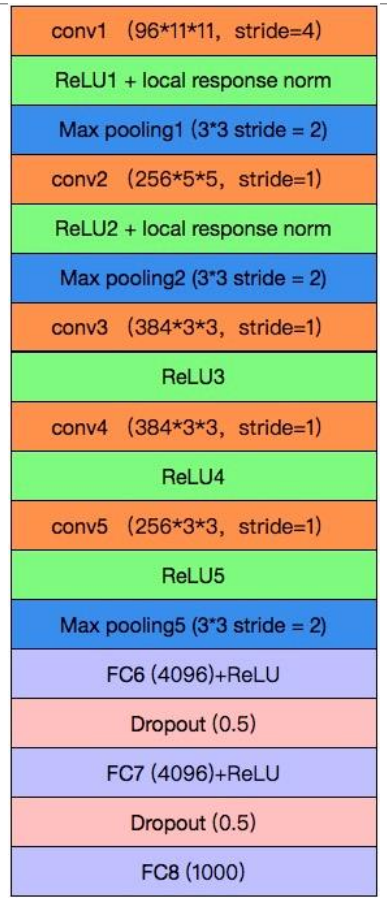
crop



reflection

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

CNN: AlexNet

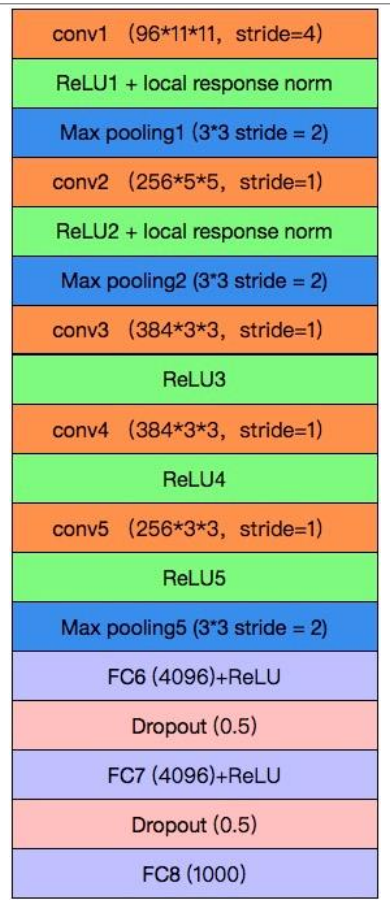


Novel:

1. Data Augmentation
2. ReLU Nonlinearity
3. Local Response Normalization
4. Overlapping Pooling
5. Dropout
6. Training on Multiple GPUs

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

CNN: AlexNet

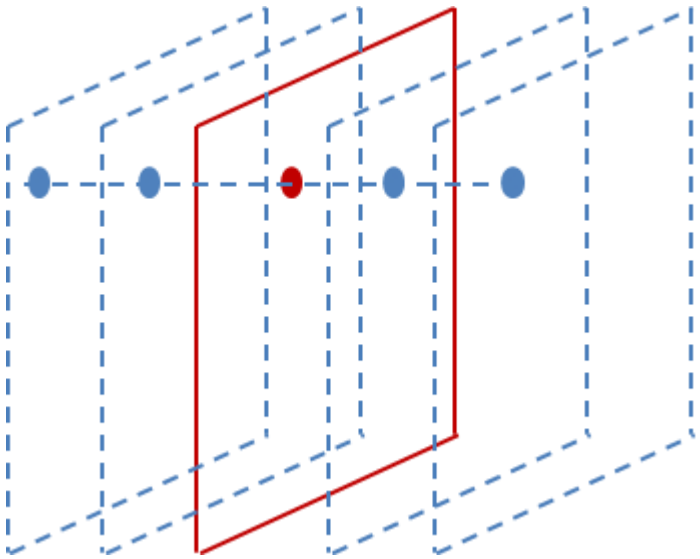


Novel:

1. Data Augmentation
2. ReLU Nonlinearity
3. Local Response Normalization
4. Overlapping Pooling
5. Dropout
6. Training on Multiple GPUs

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

Alexnet has a special layer of computing, the LRN layer, and does something to smooth the output results of the current layer.



A few layers of the front and back (the point of the corresponding position) make a smooth constraint on the middle layer. The calculation method is:

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2 \right)^{\beta}$$

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

CNN: AlexNet

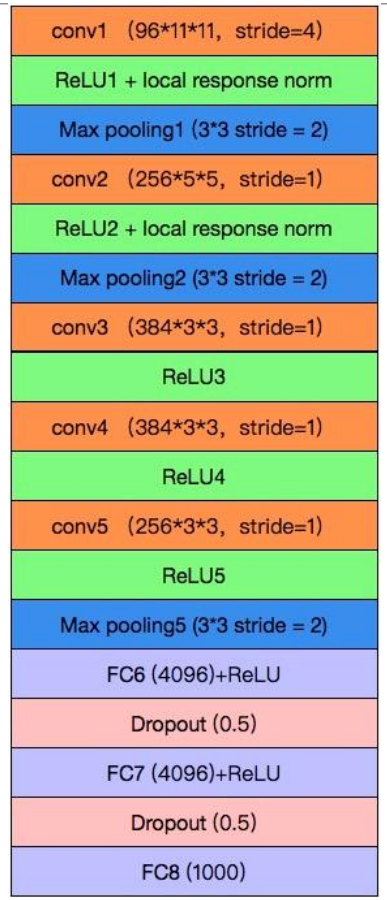
Local Response Normalization：

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2 \right)^{\beta}$$

Effect: Reduces top-1 error and top-5 error by 1.4% and 1.2%

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

CNN: AlexNet

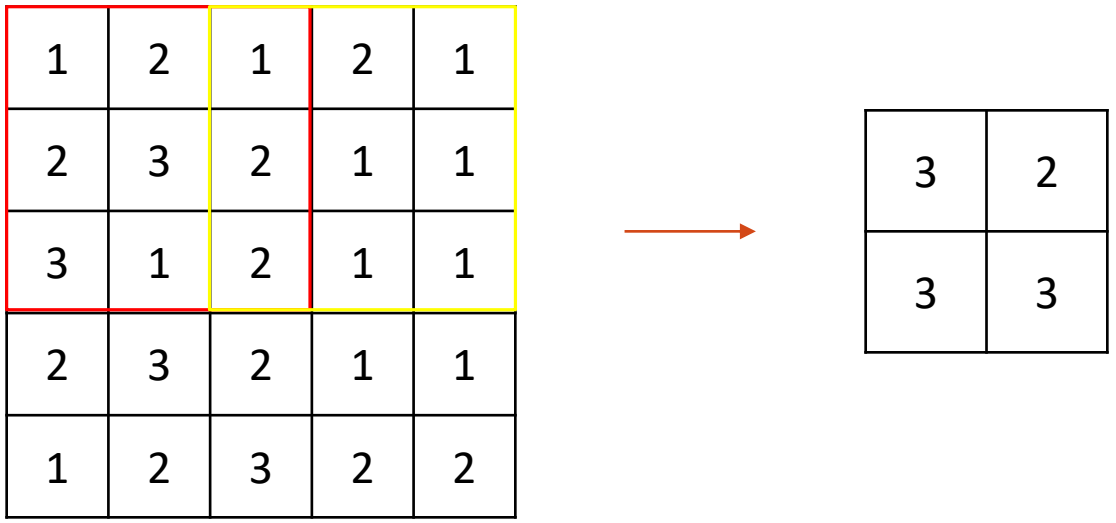


Novel:

1. Data Augmentation
2. ReLU Nonlinearity
3. Local Response Normalization
4. Overlapping Pooling
5. Dropout
6. Training on Multiple GPUs

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

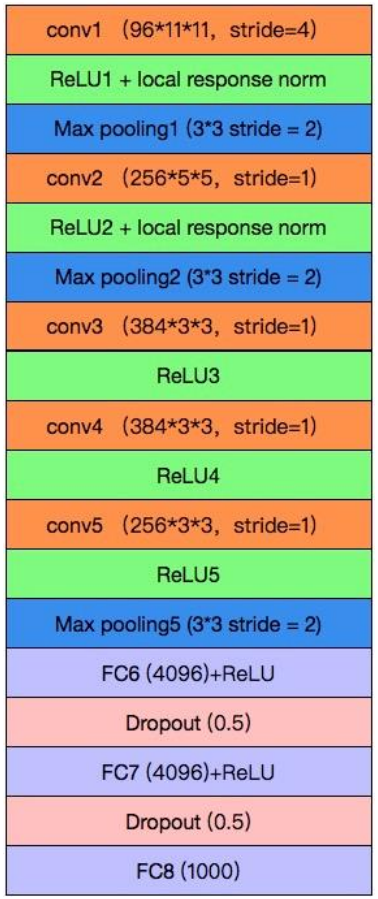
CNN: AlexNet



Effect: Reduces top-1 error and top-5 error by 0.4% and 0.3%

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

AlexNet



Novel:

1. Data Augmentation
2. ReLU Nonlinearity
3. Local Response Normalization
4. Overlapping Pooling
5. Dropout
6. Training on Multiple GPUs

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Nips. 2012.

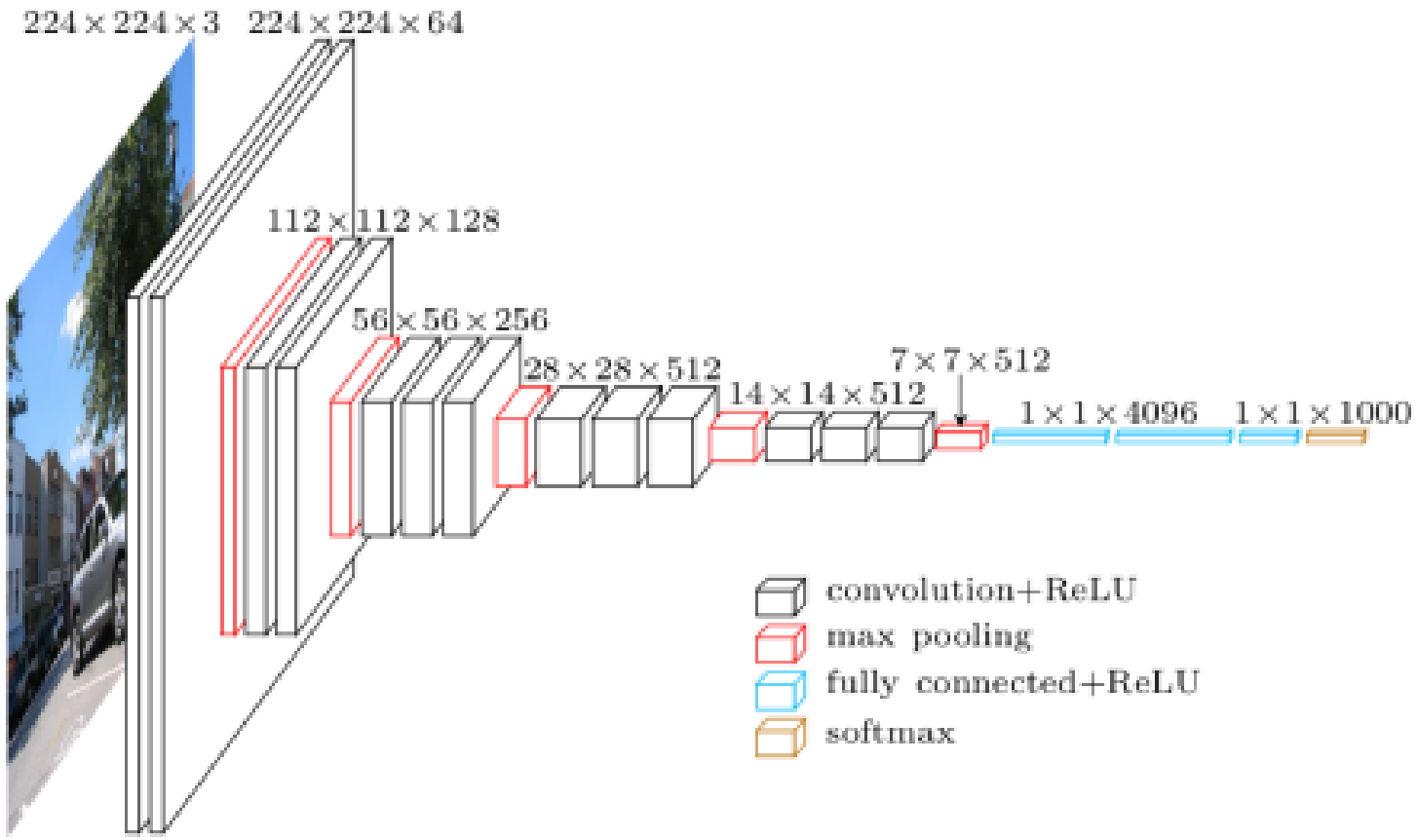
CNN: VGGNet(2014)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

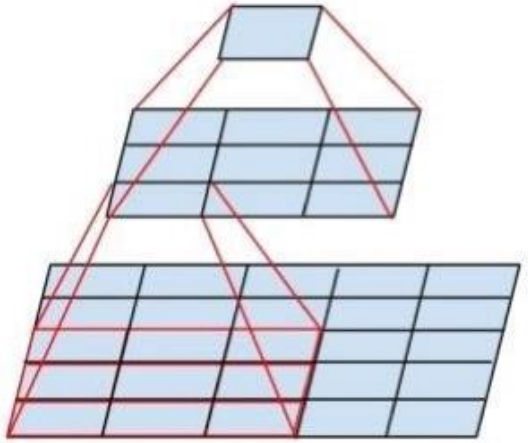
- More layers
- Non-overlapping pooling
- No LRN
- Smaller kernel size and stride

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

VGG-16/VGG-19



Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.



	5*5	3*3 + 3*3
Parameter Amount	5*5+1=26	2*(3*3+1)=20

Advantages：

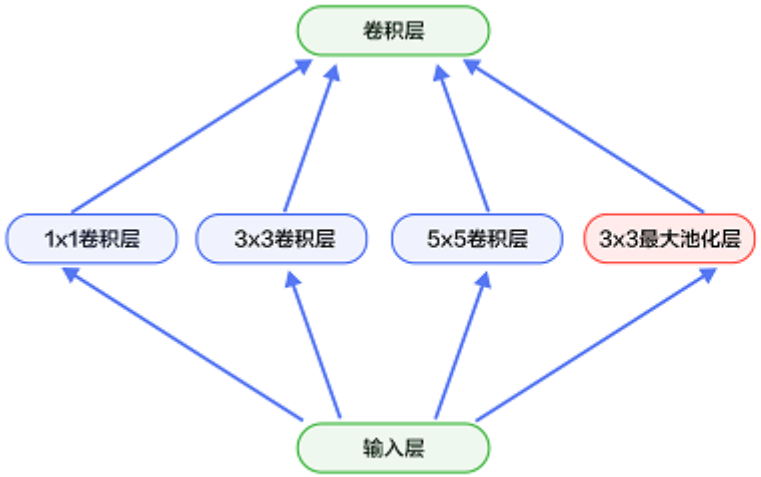
1. Less parameters
2. Multiple 3*3 kernels can replace much bigger kernel
3. More nonlinear

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

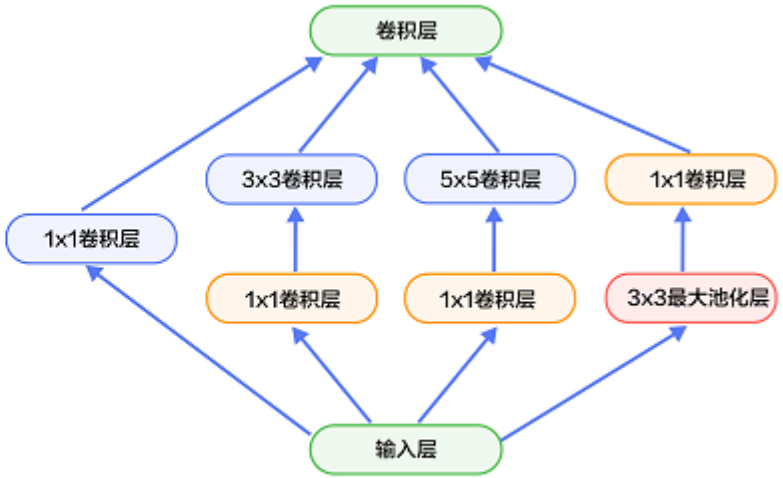
ILSVRC 2014's champion network

Szegedy, Christian, et al. "Going deeper with convolutions." Cvpr, 2015.

The Inception module



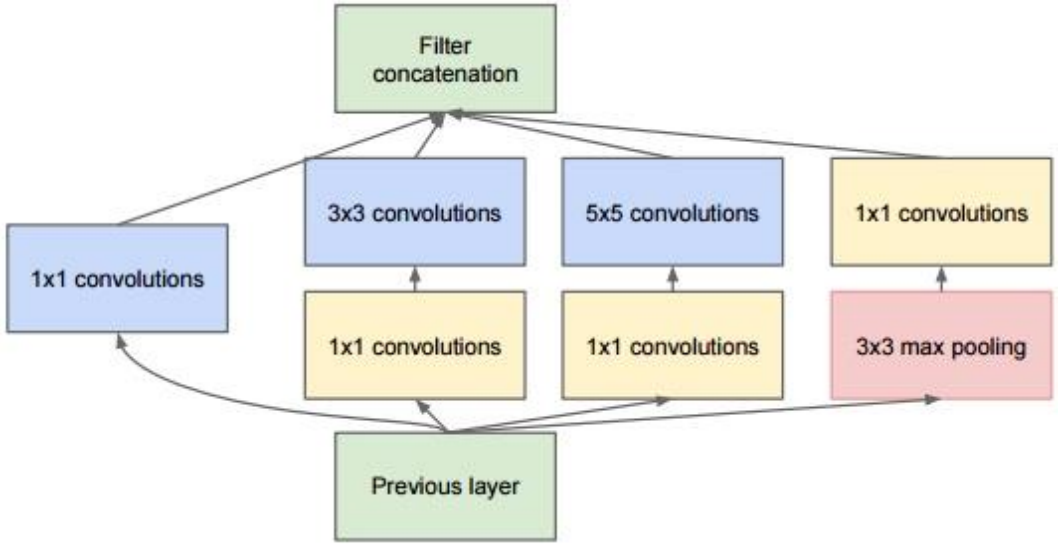
*Inception*简单模块



*Inception*含降维模块

Szegedy, Christian, et al. "Going deeper with convolutions." Cvpr, 2015.

Inception Architecture：

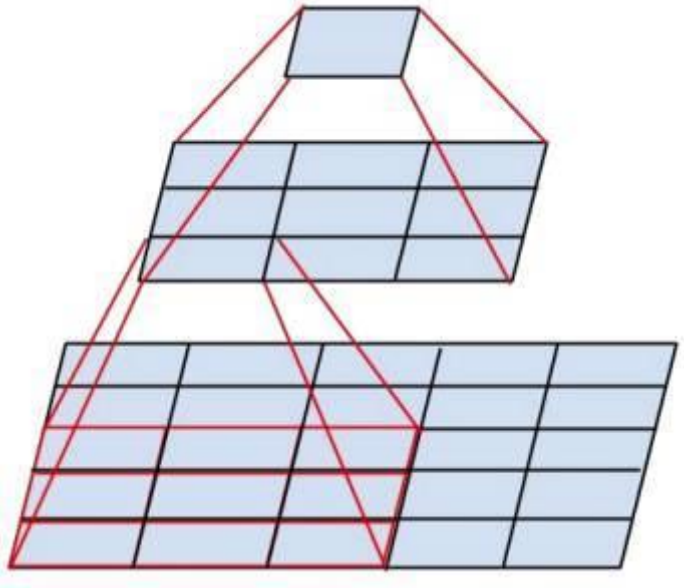


Main idea: consider how an optimal local sparse structure of a convolutional vision network can be approximated and covered by readily available dense components

Szegedy, Christian, et al. "Going deeper with convolutions." Cvpr, 2015.

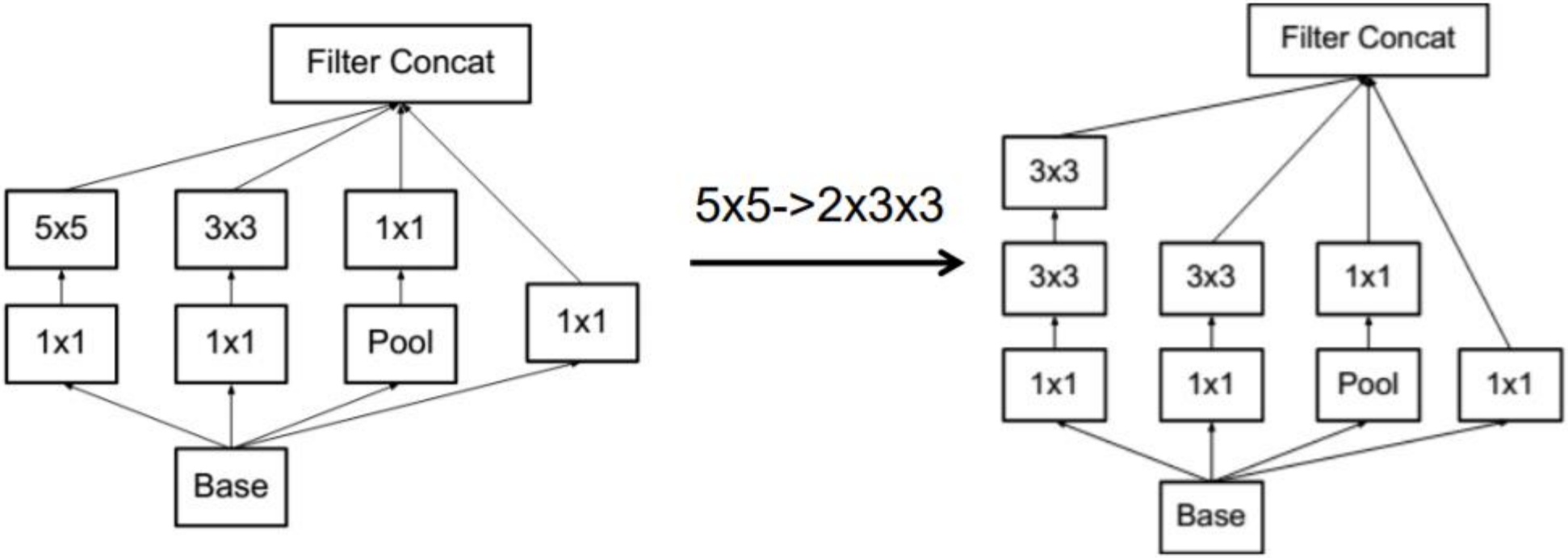
Inception v2

1. The BN layer is added.
2. Uses 2 3x3 conv instead of 5x5 in the inception module



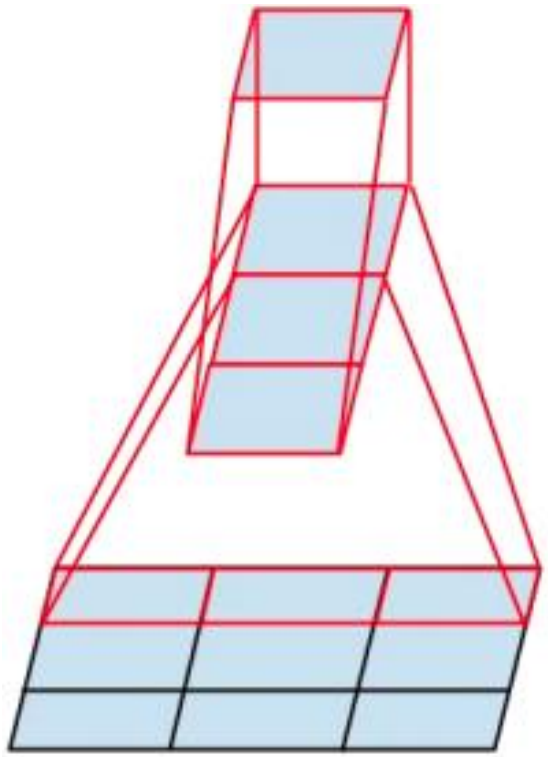
Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

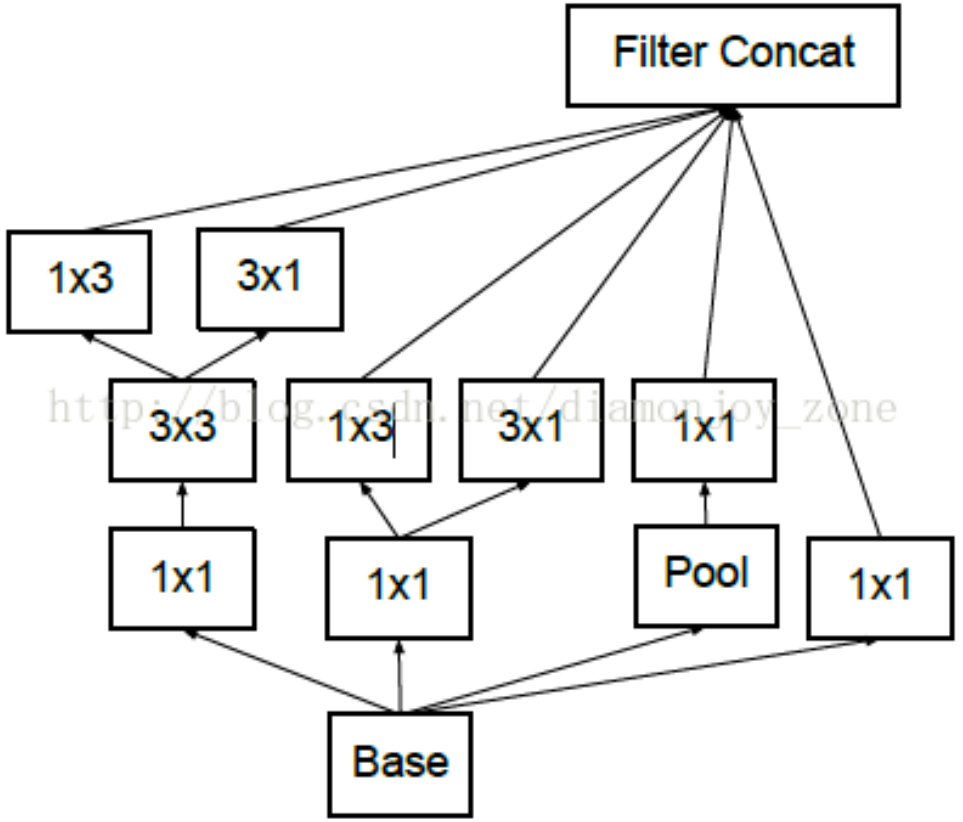
Inception v2



Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

Inception v3





Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.

Inception V4, Inception-ResNet

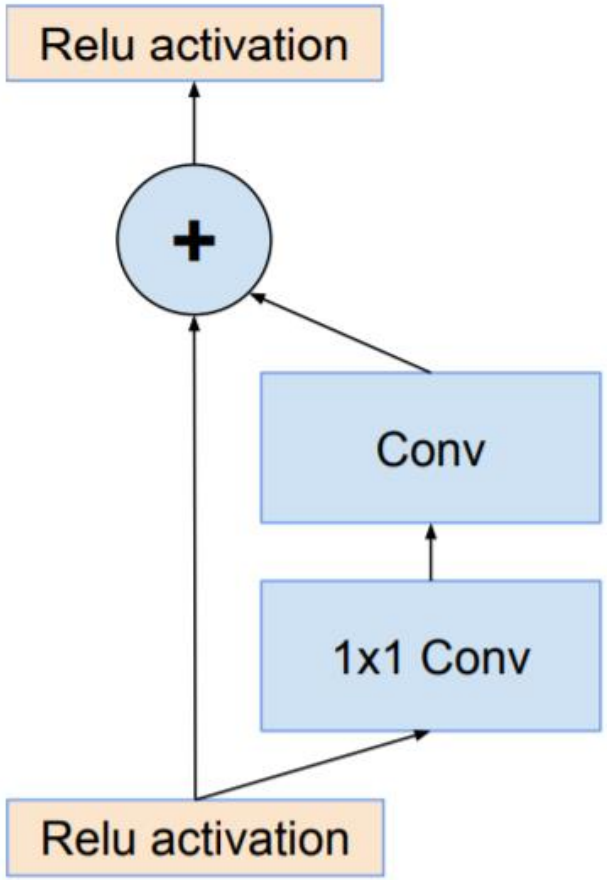
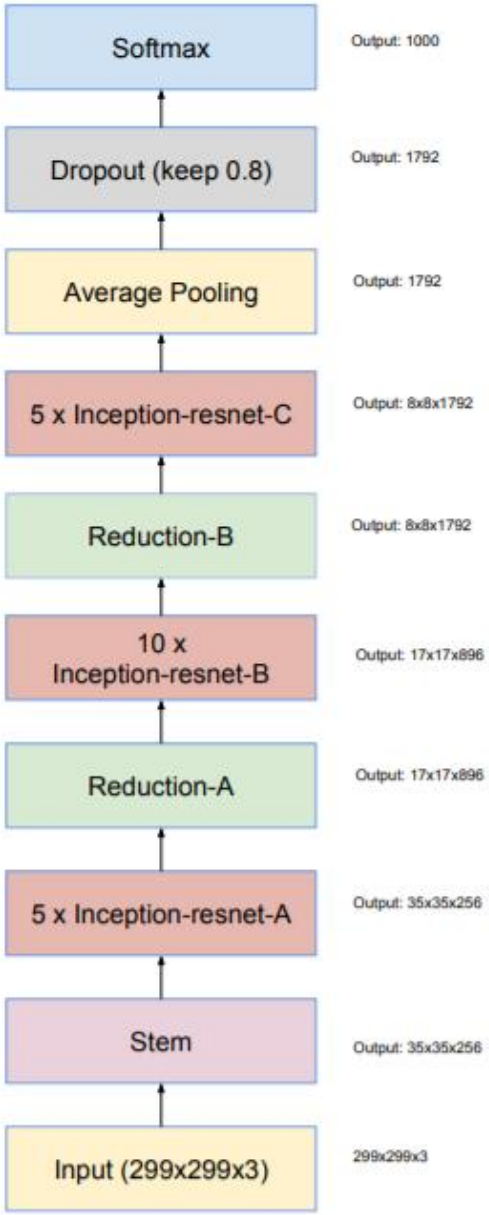


Figure 2. Optimized version of ResNet connections by [5] to shield computation.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

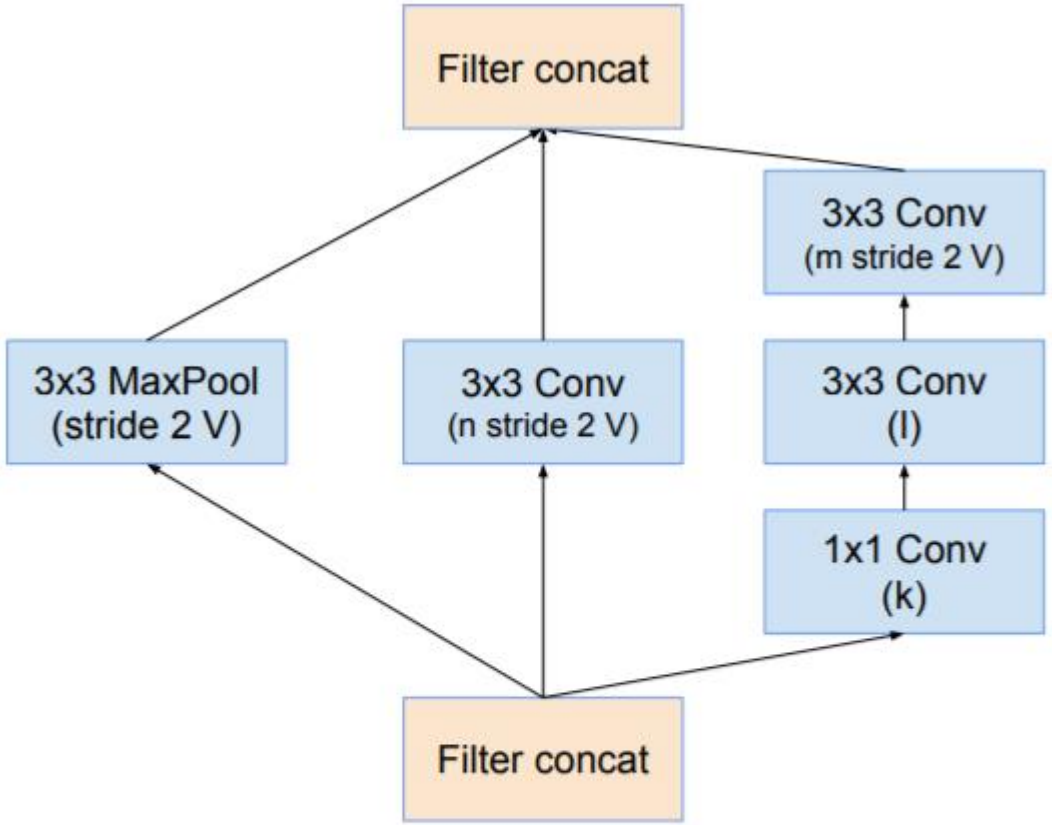


Figure 7. The schema for 35×35 to 17×17 reduction module. Different variants of this blocks (with various number of filters) are used in Figure 9, and 15 in each of the new Inception(-v4, -ResNet-v1, -ResNet-v2) variants presented in this paper. The k, l, m, n numbers represent filter bank sizes which can be looked up in Table 1.

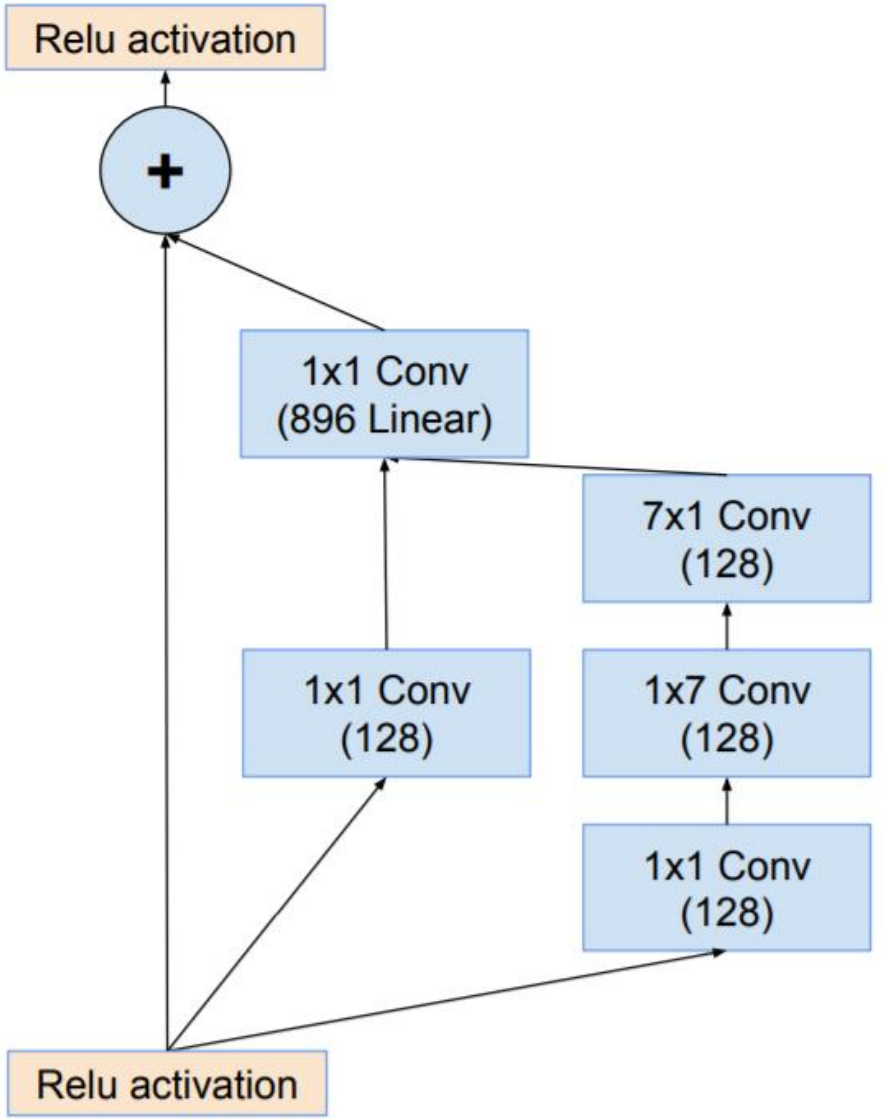


Figure 11. The schema for 17×17 grid (Inception-ResNet-B) module of Inception-ResNet-v1 network.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

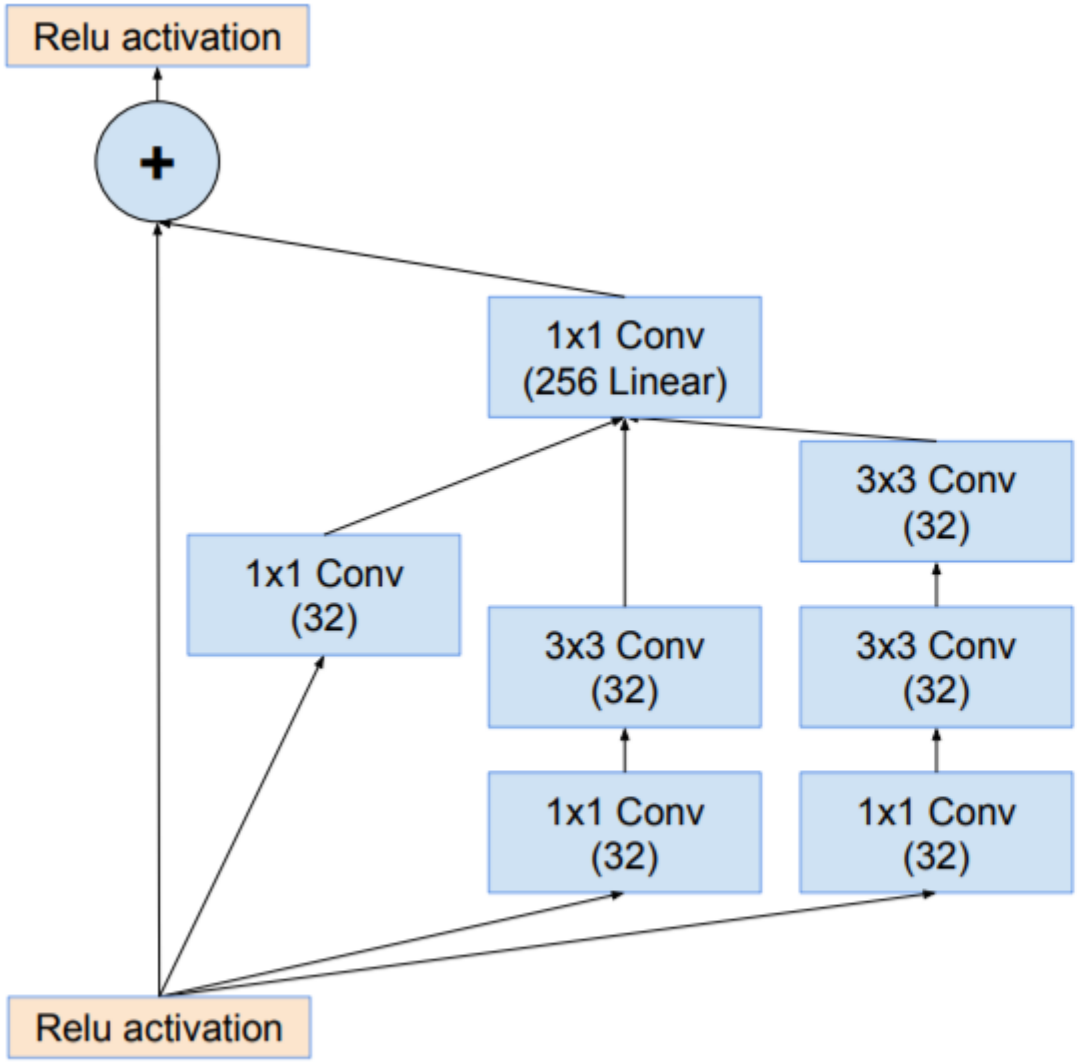


Figure 10. The schema for 35 × 35 grid (Inception-ResNet-A) module of Inception-ResNet-v1 network.

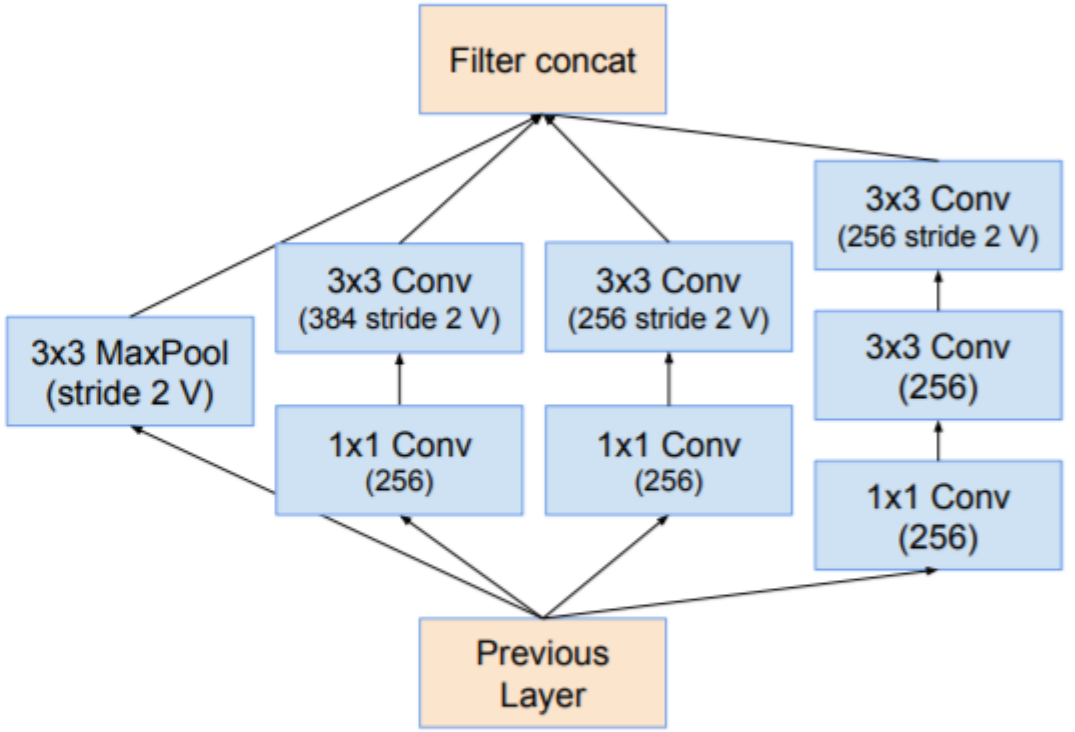


Figure 12. "Reduction-B" 17 × 17 to 8 × 8 grid-reduction module. This module used by the smaller Inception-ResNet-v1 network in Figure 15.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

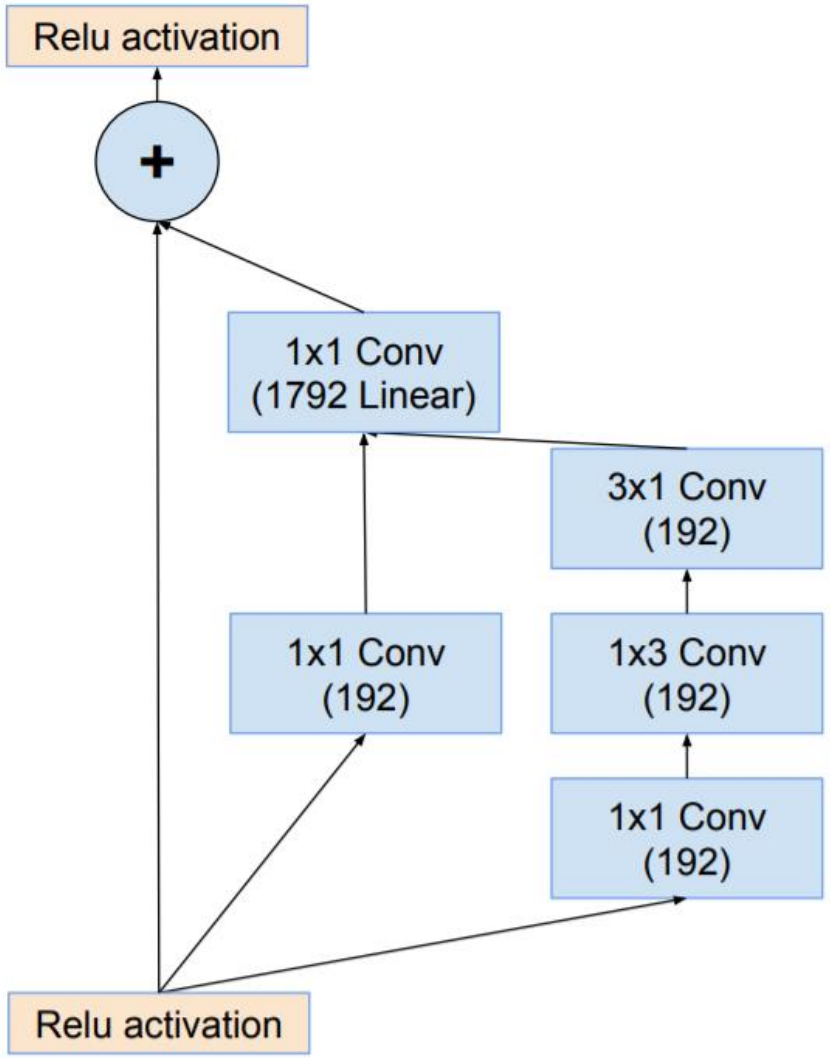


Figure 13. The schema for 8×8 grid (Inception-ResNet-C) module of Inception-ResNet-v1 network.

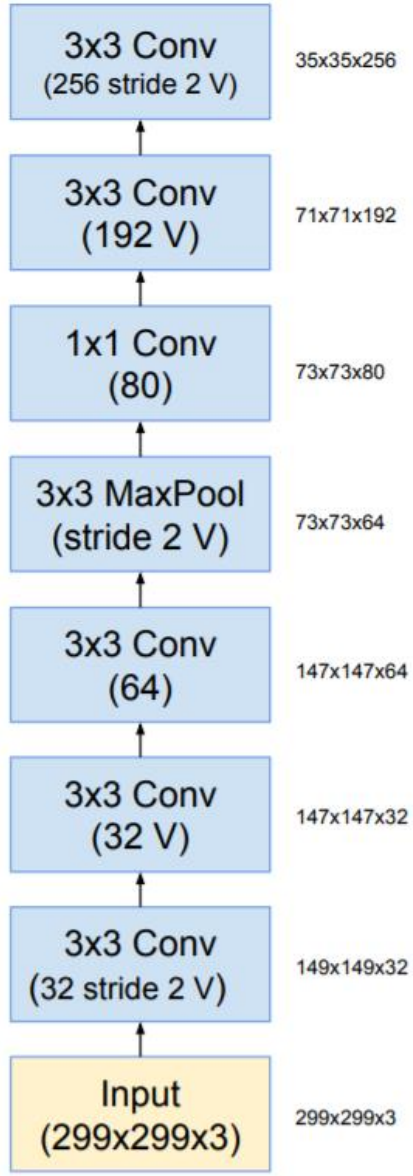
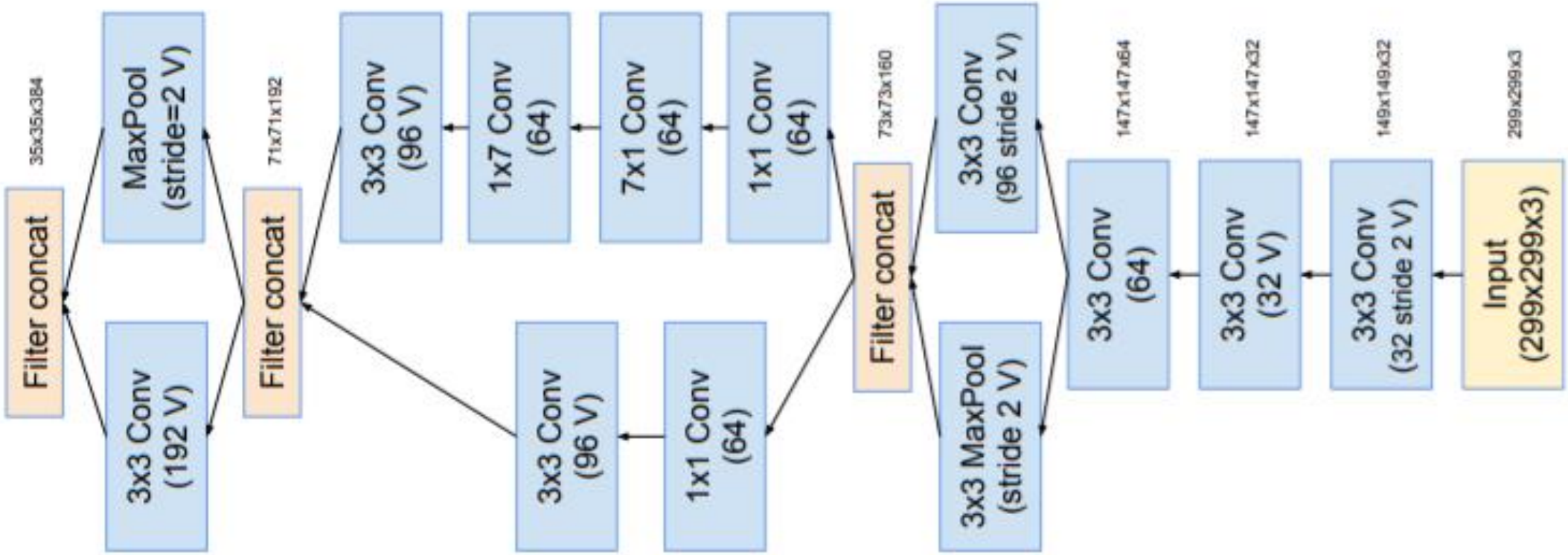


Figure 14. The stem of the Inception-ResNet-v1 r

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

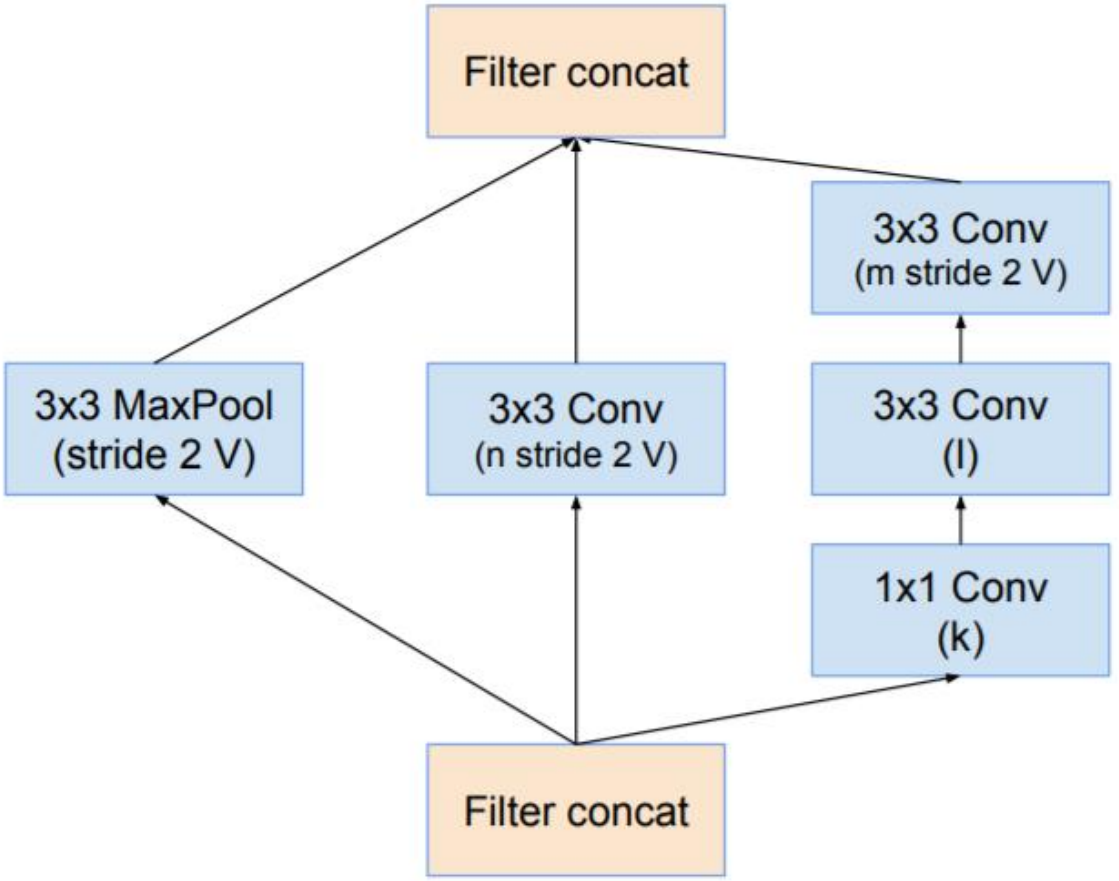


Figure 7. The schema for 35×35 to 17×17 reduction module. Different variants of this blocks (with various number of filters) are used in Figure 9, and 15 in each of the new Inception(-v4, -ResNet-v1, -ResNet-v2) variants presented in this paper. The k, l, m, n numbers represent filter bank sizes which can be looked up in Table 1.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

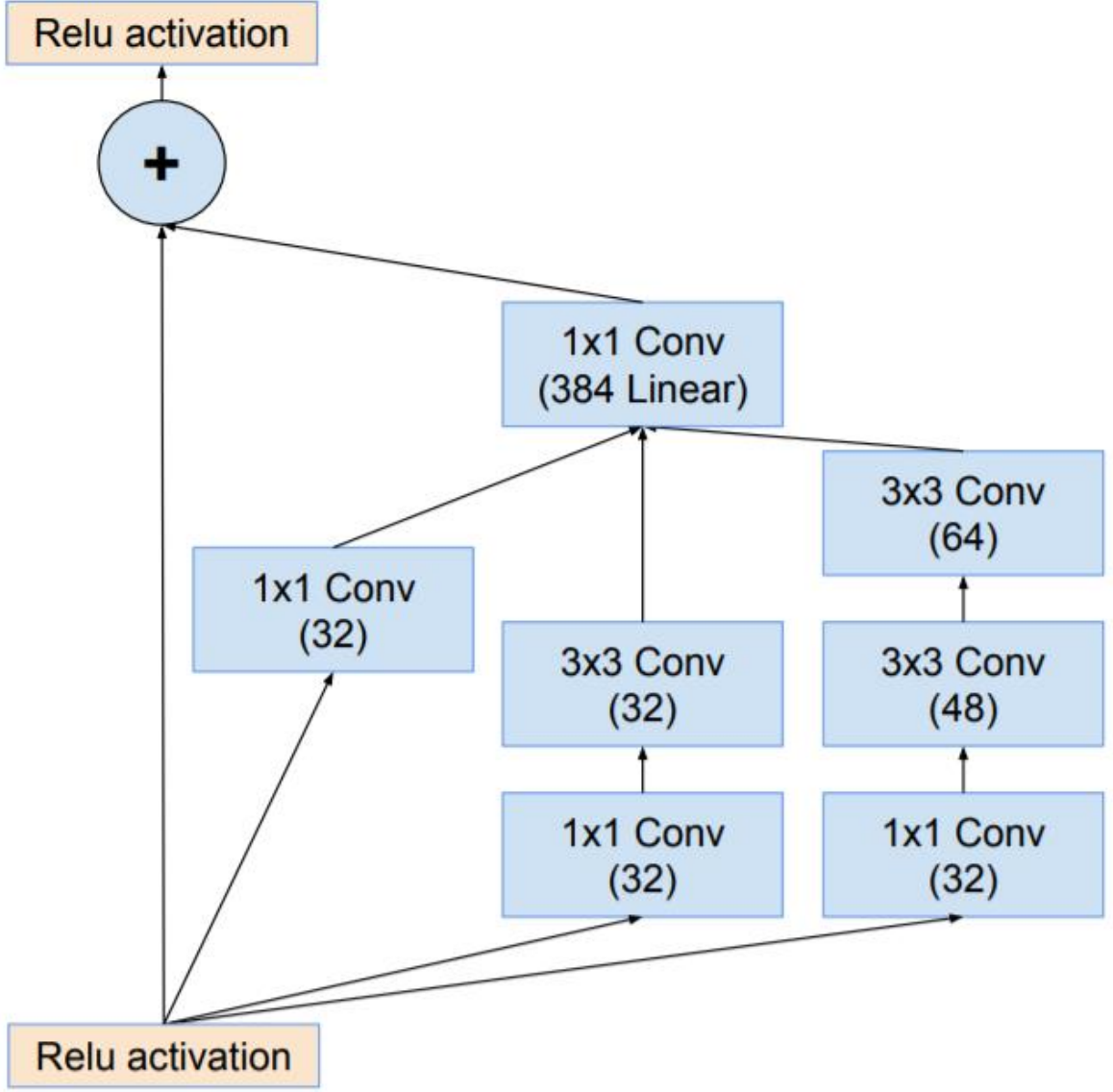
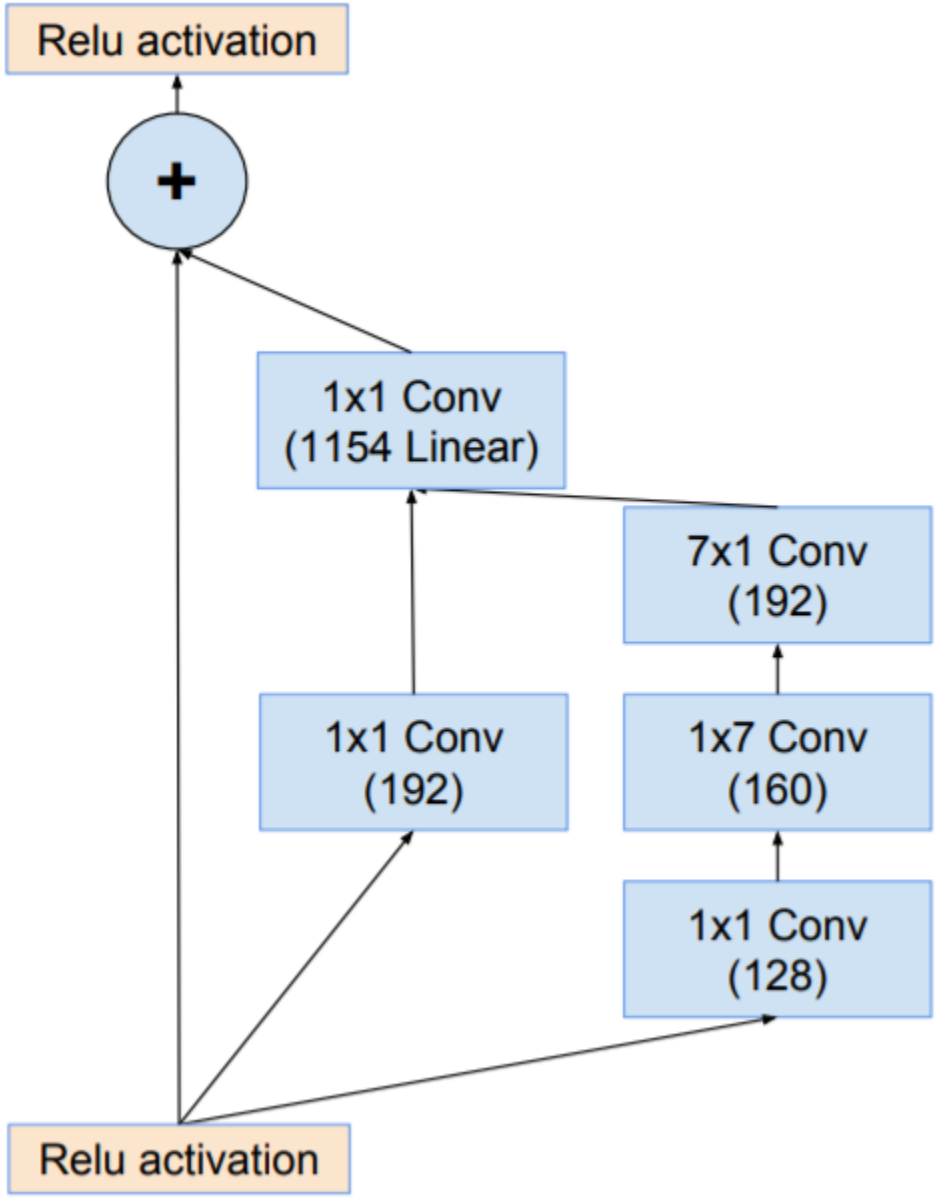
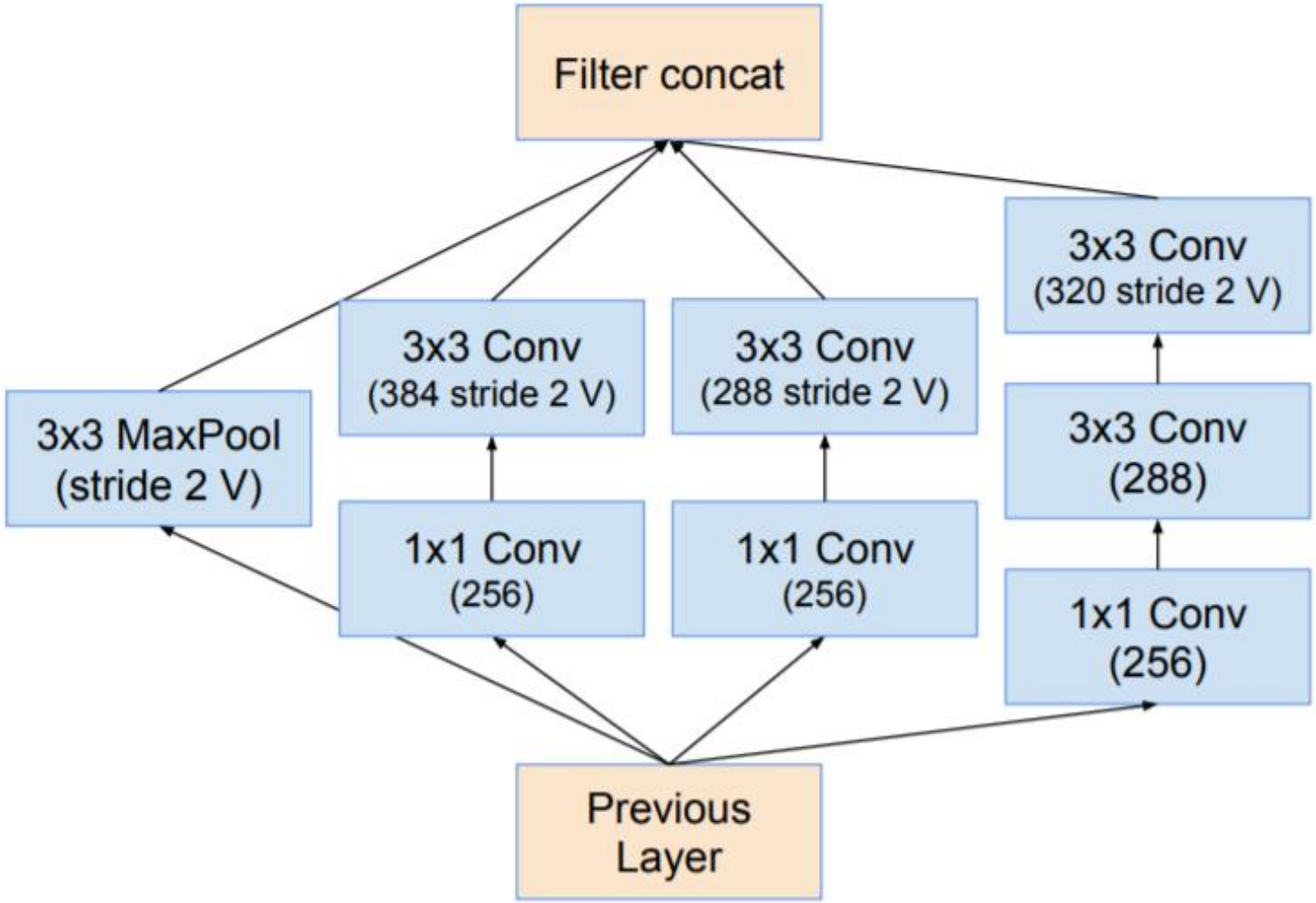


Figure 16. The schema for 35×35 grid (Inception-ResNet-A) module of the Inception-ResNet-v2 network.

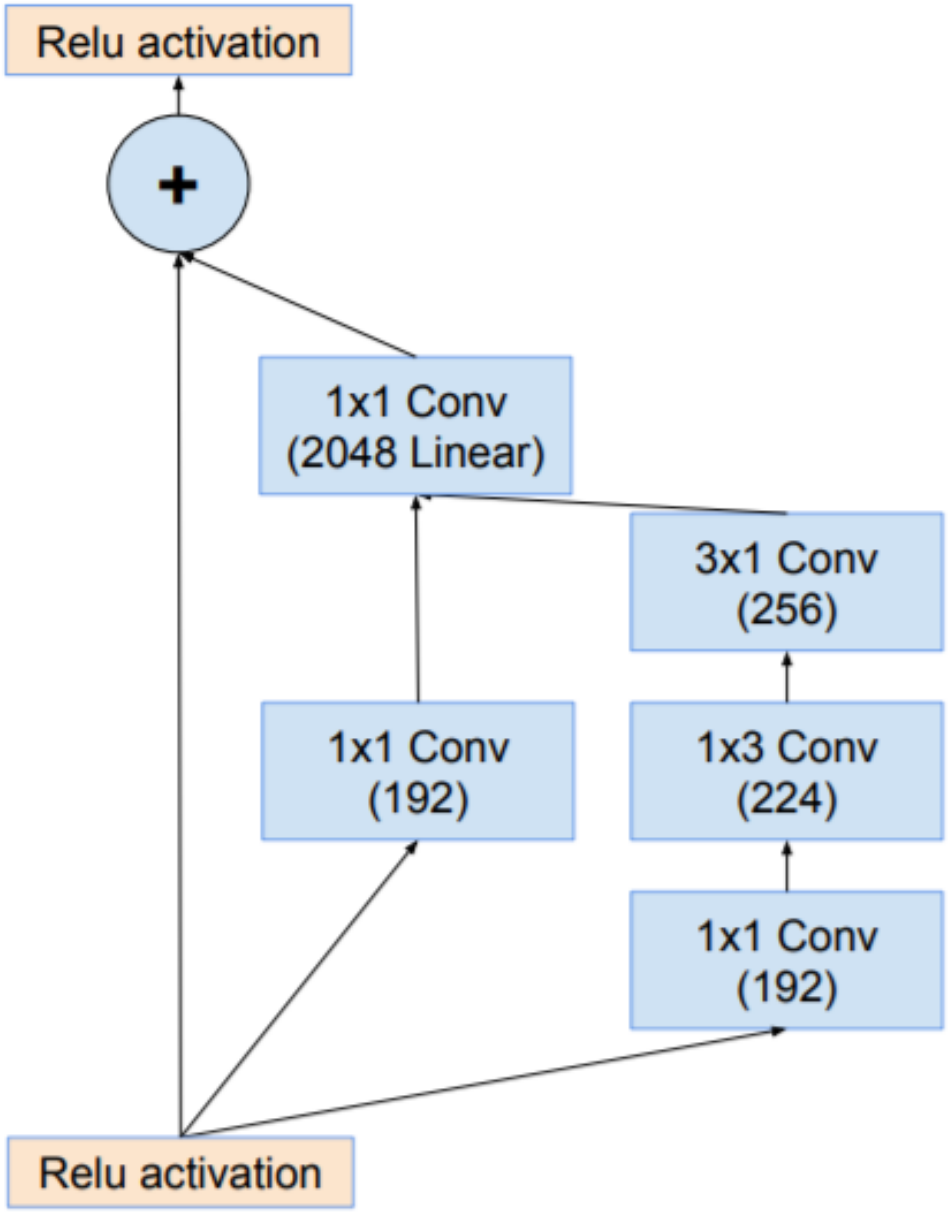
Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



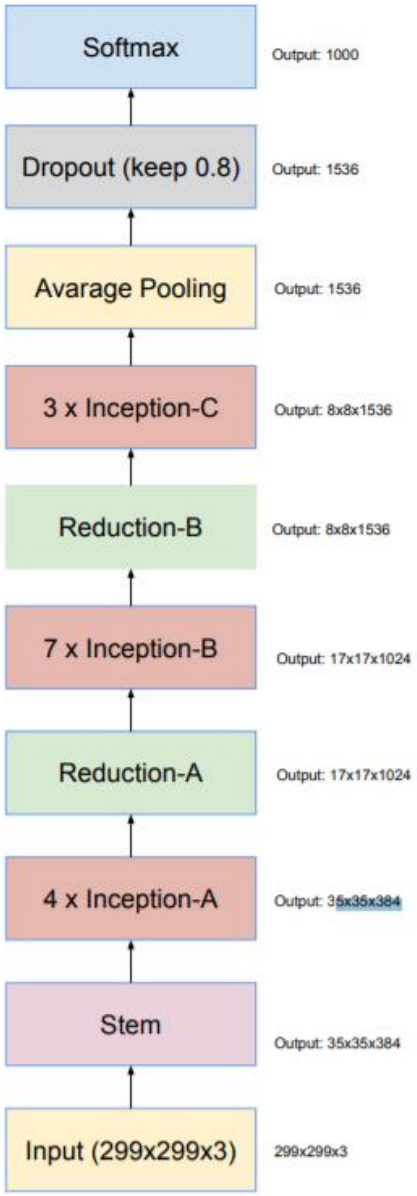
Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.



Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

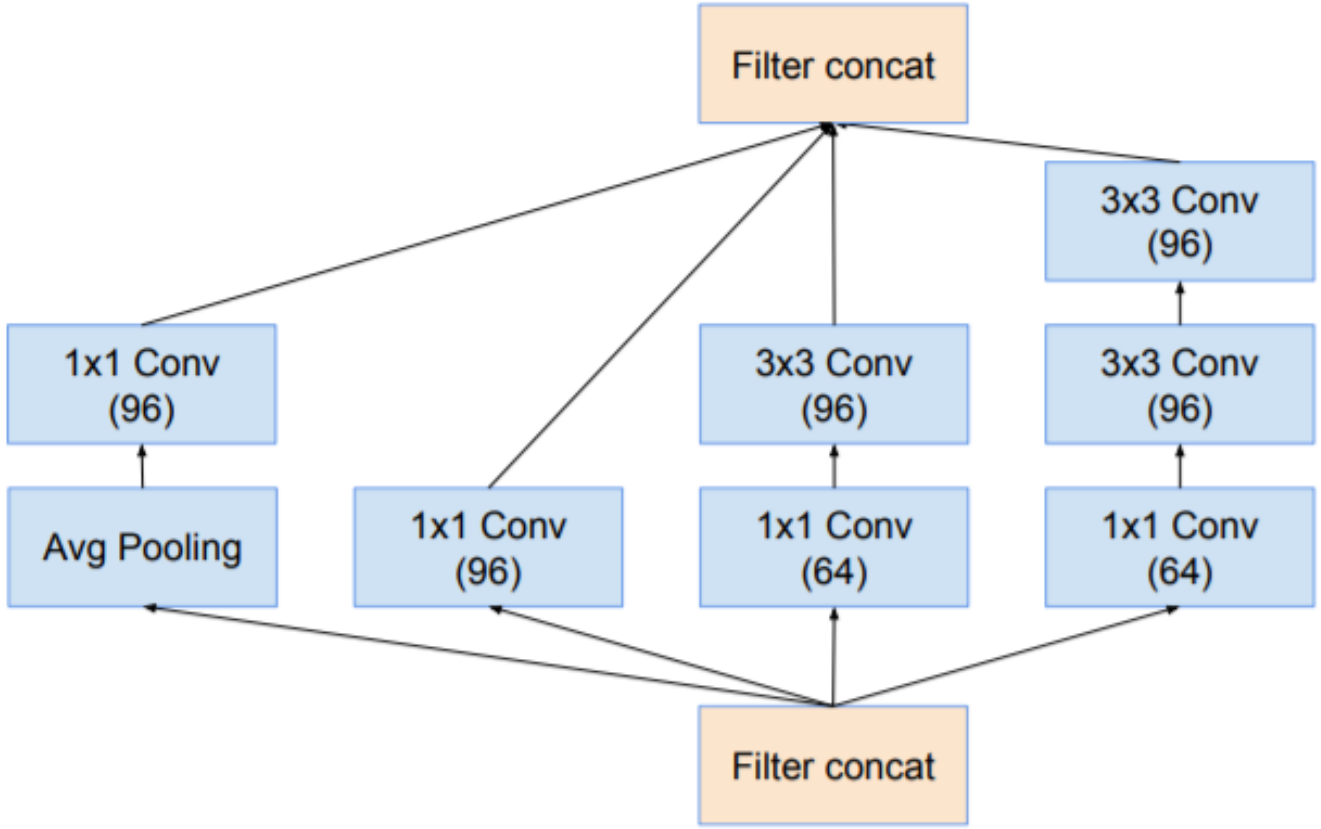


Figure 4. The schema for 35 × 35 grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 9.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

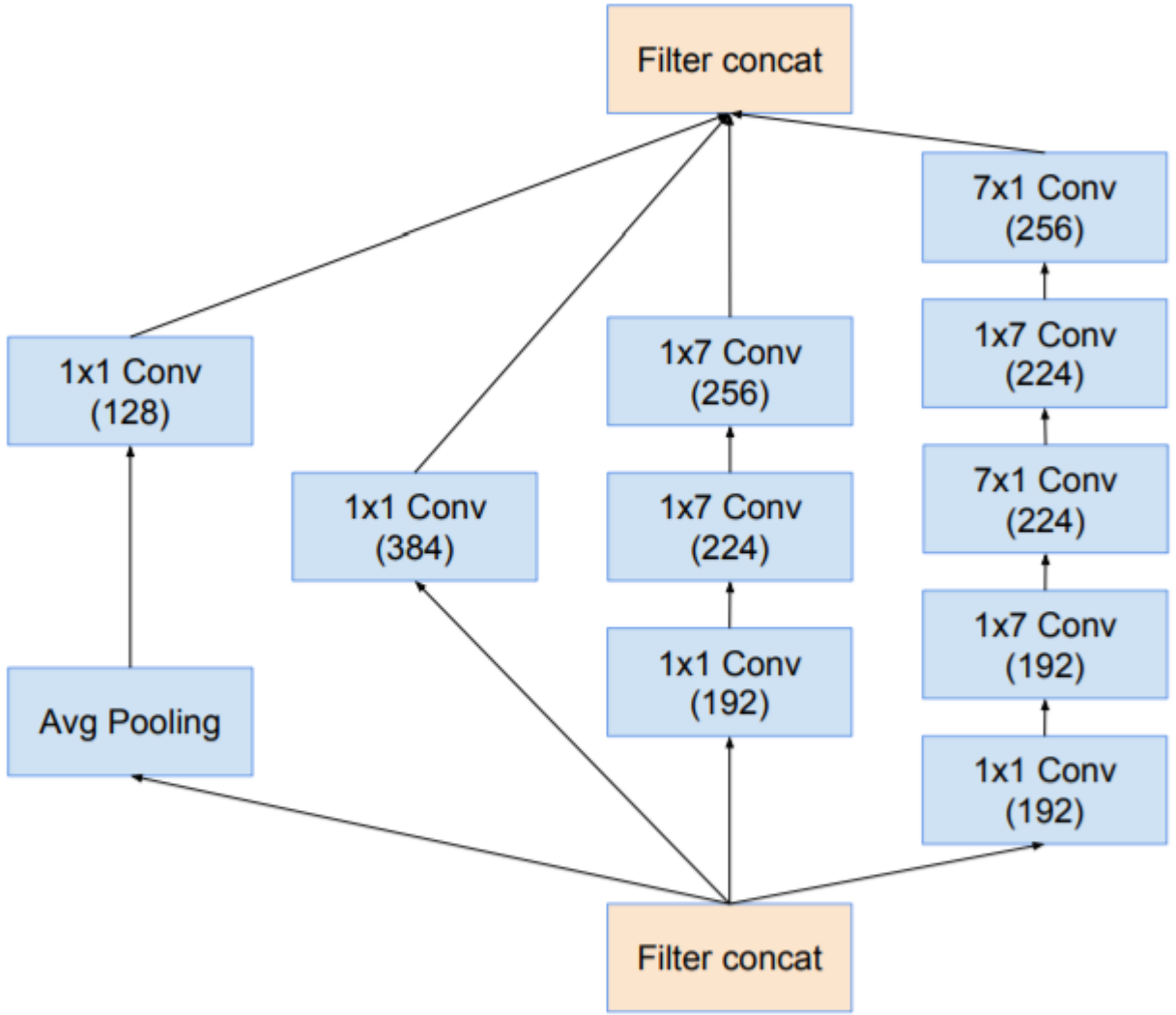


Figure 5. The schema for 17 × 17 grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 9.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

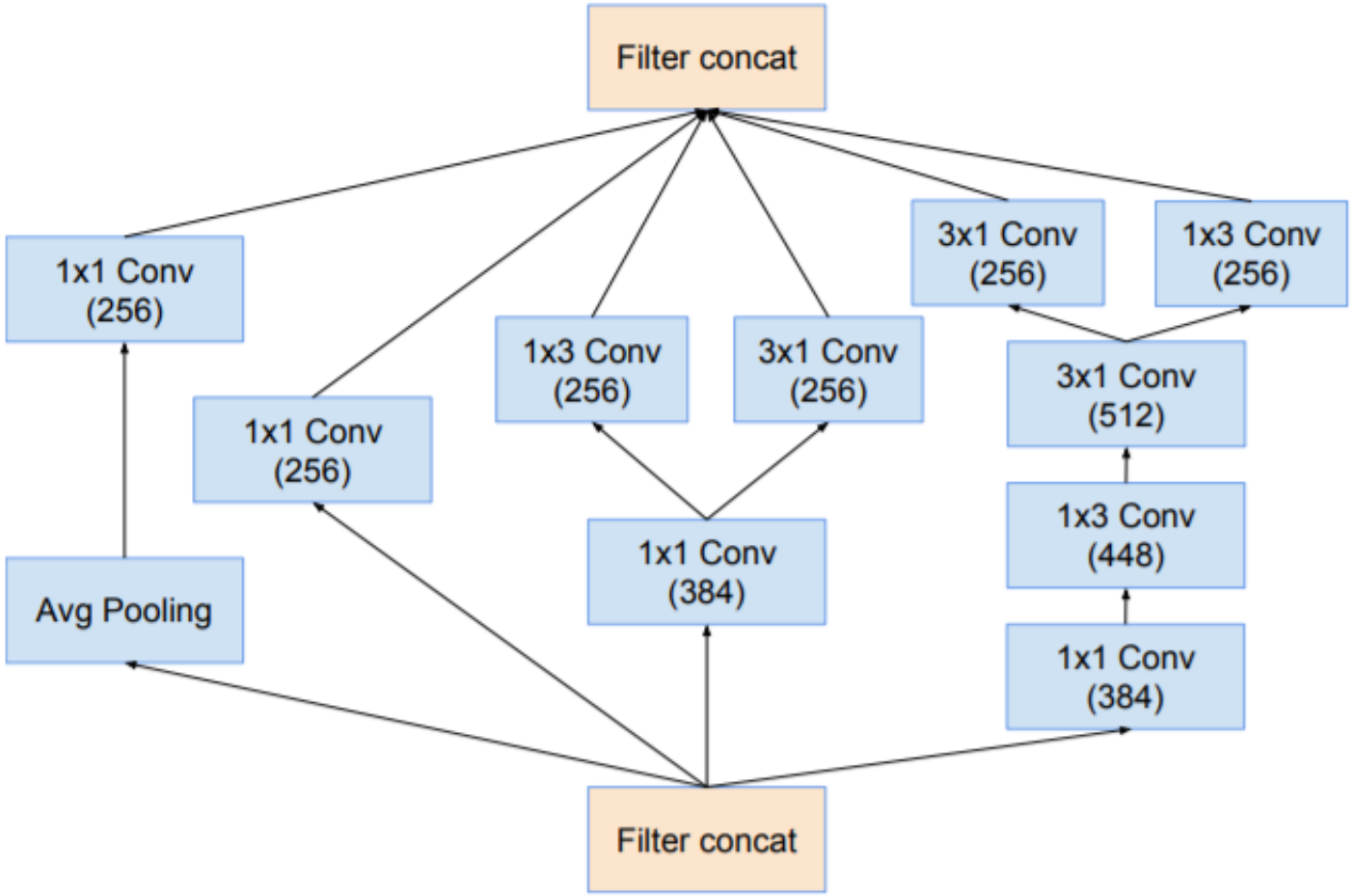
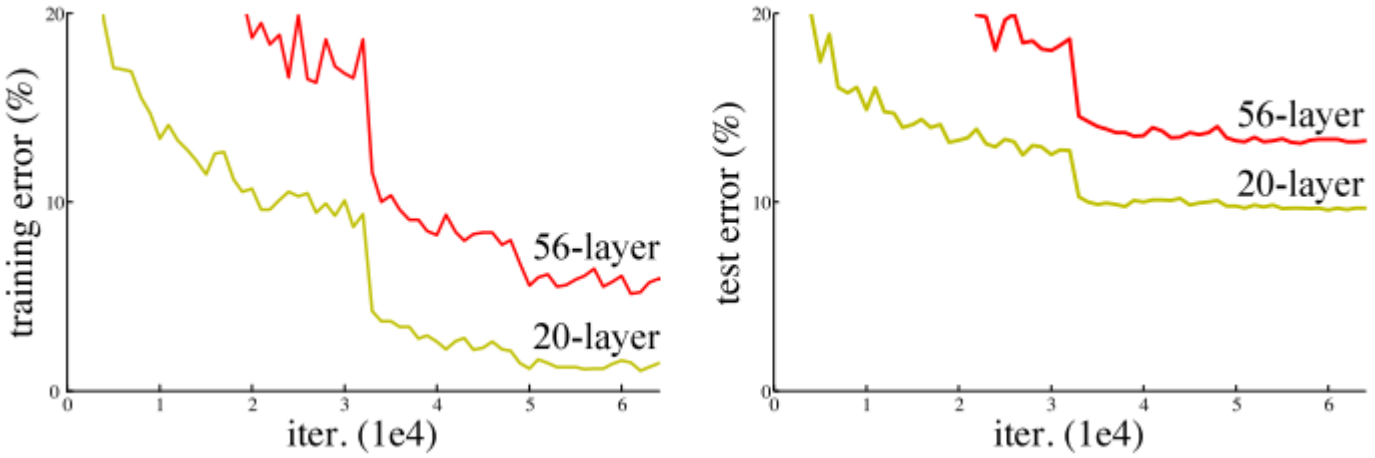


Figure 6. The schema for 8×8 grid modules of the pure Inception-v4 network. This is the Inception-C block of Figure 9.

Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI. Vol. 4. 2017.

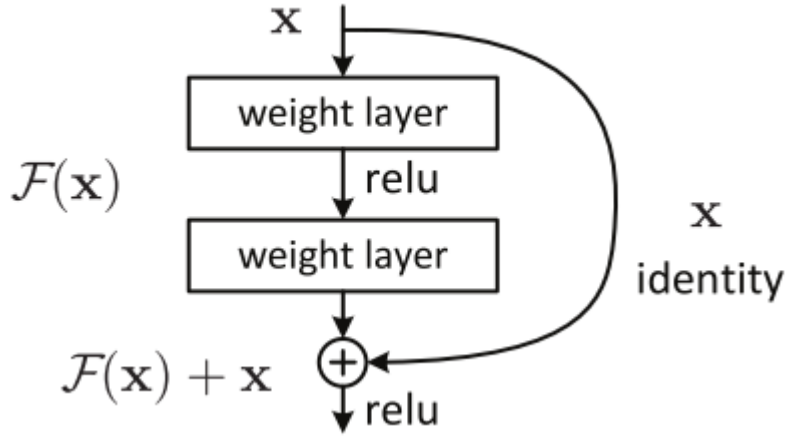
ILSVRC 2015's champion network



Network depth increases, resulting in performance degradation

He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

The core idea of ResNet is to introduce a "identity shortcut connection" and skip a layer or multi layer directly, as shown in the following figure:

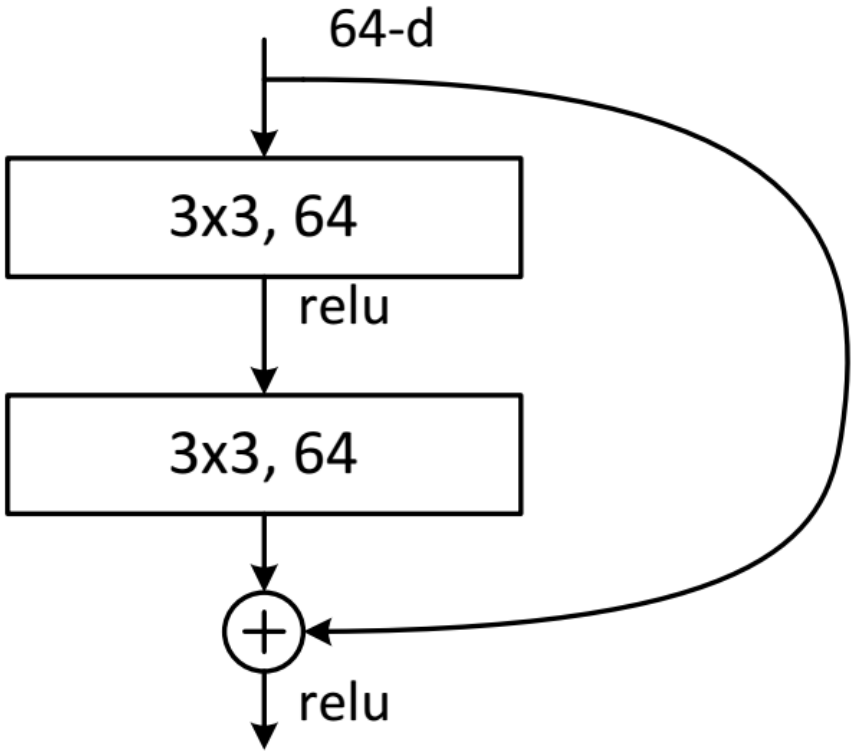


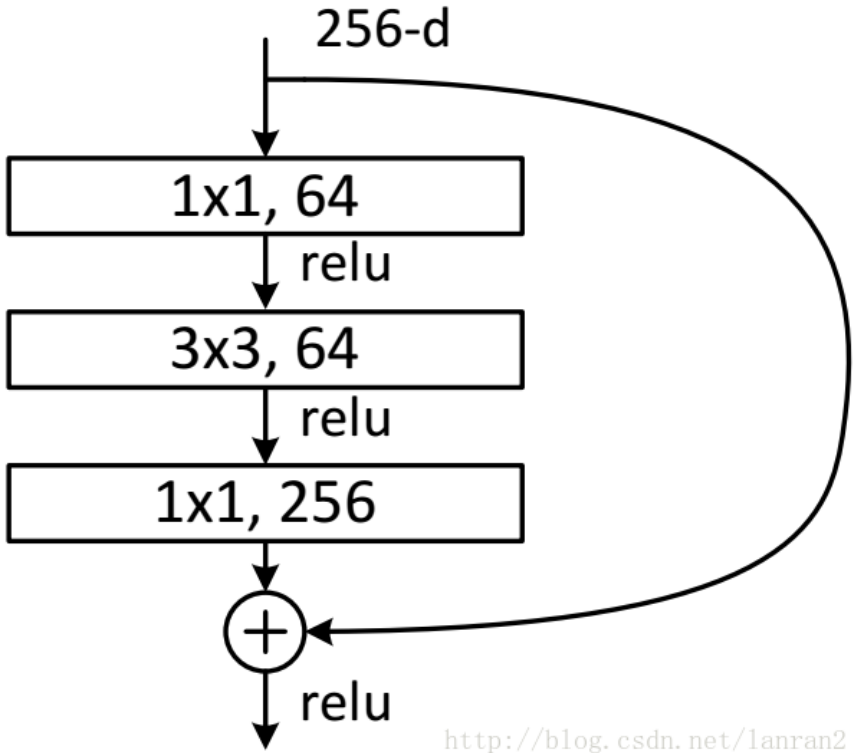
A residual block

He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

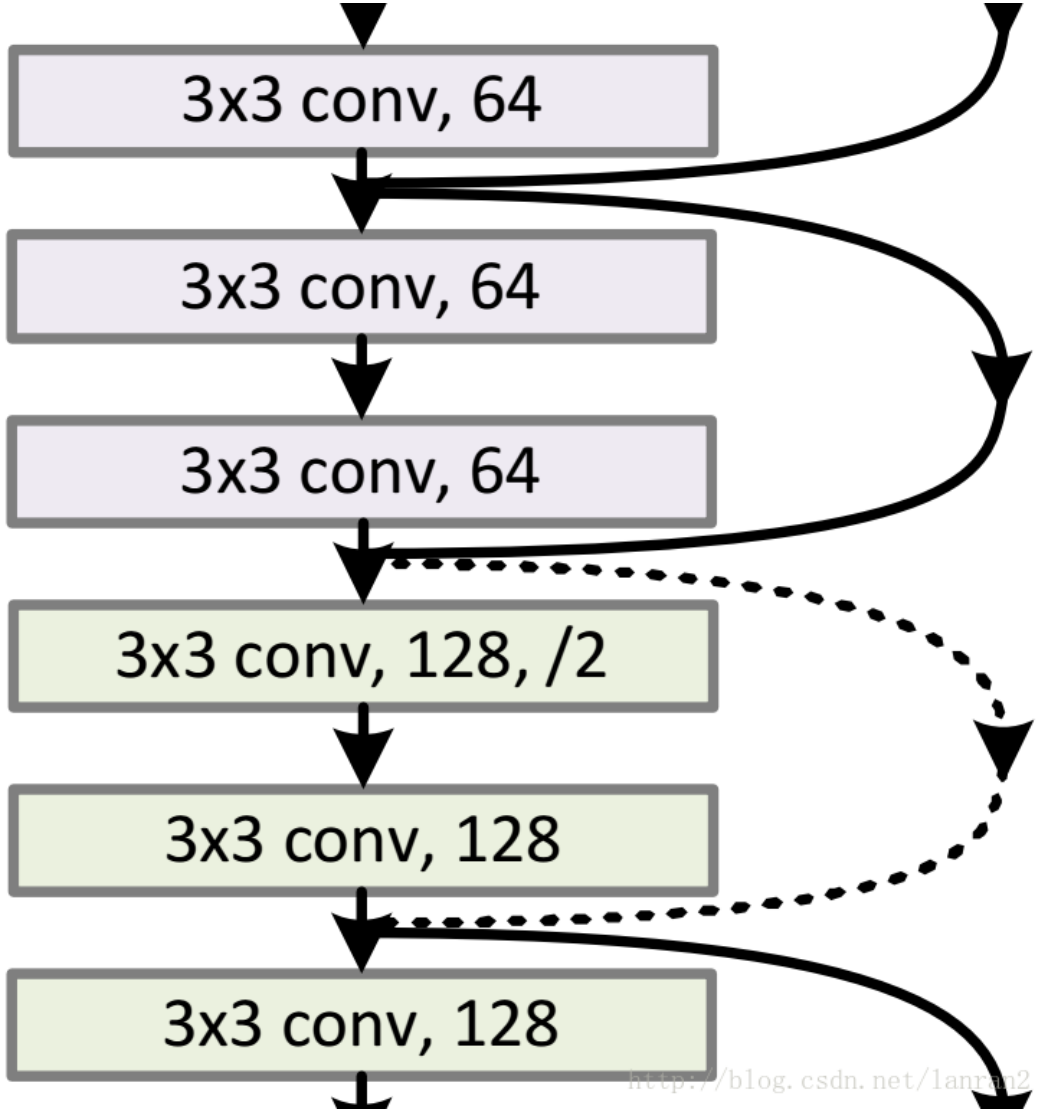
method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.





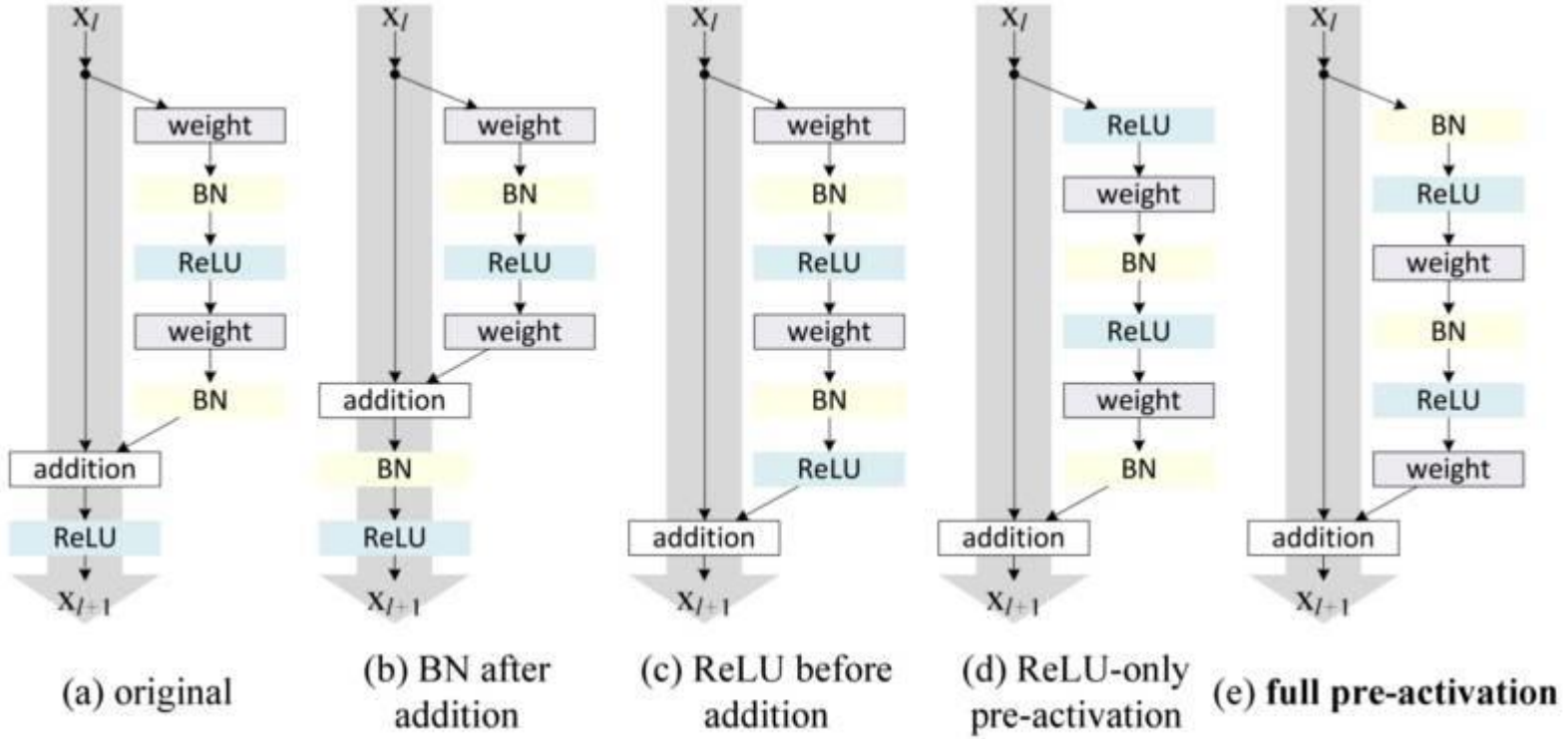
He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.



He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

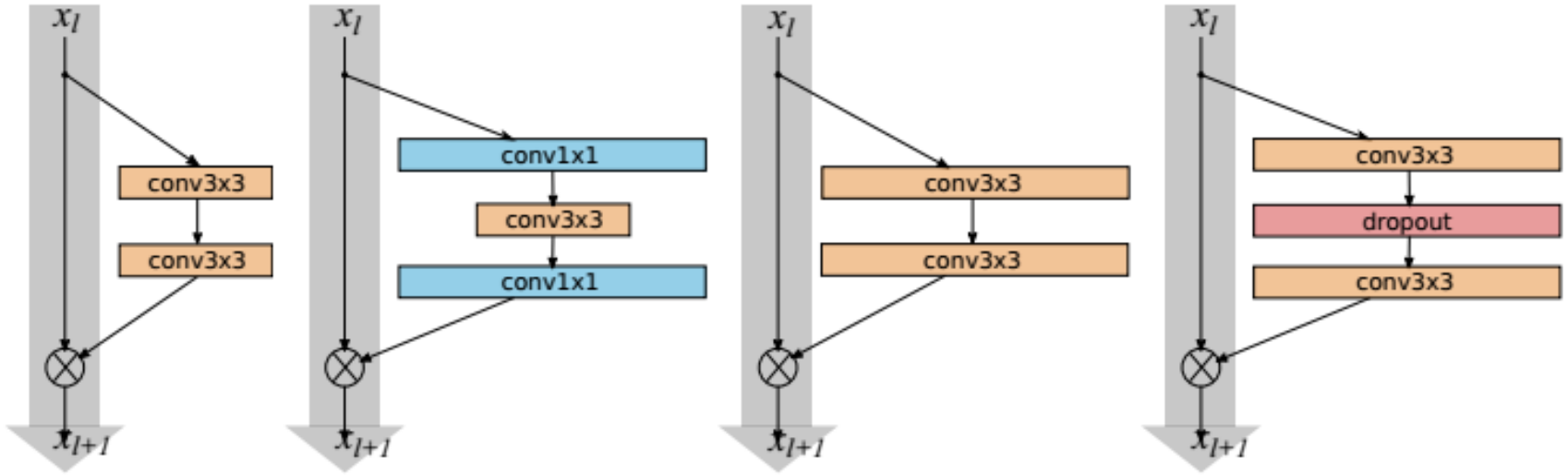
He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.



A variant of a residual block

He K, Zhang X, Ren S, et al. Identity Mappings in Deep Residual Networks[J]. arXiv preprint arXiv:1603.05027, 2016.

Wide Residual Network（WRN）:Start with the "width"

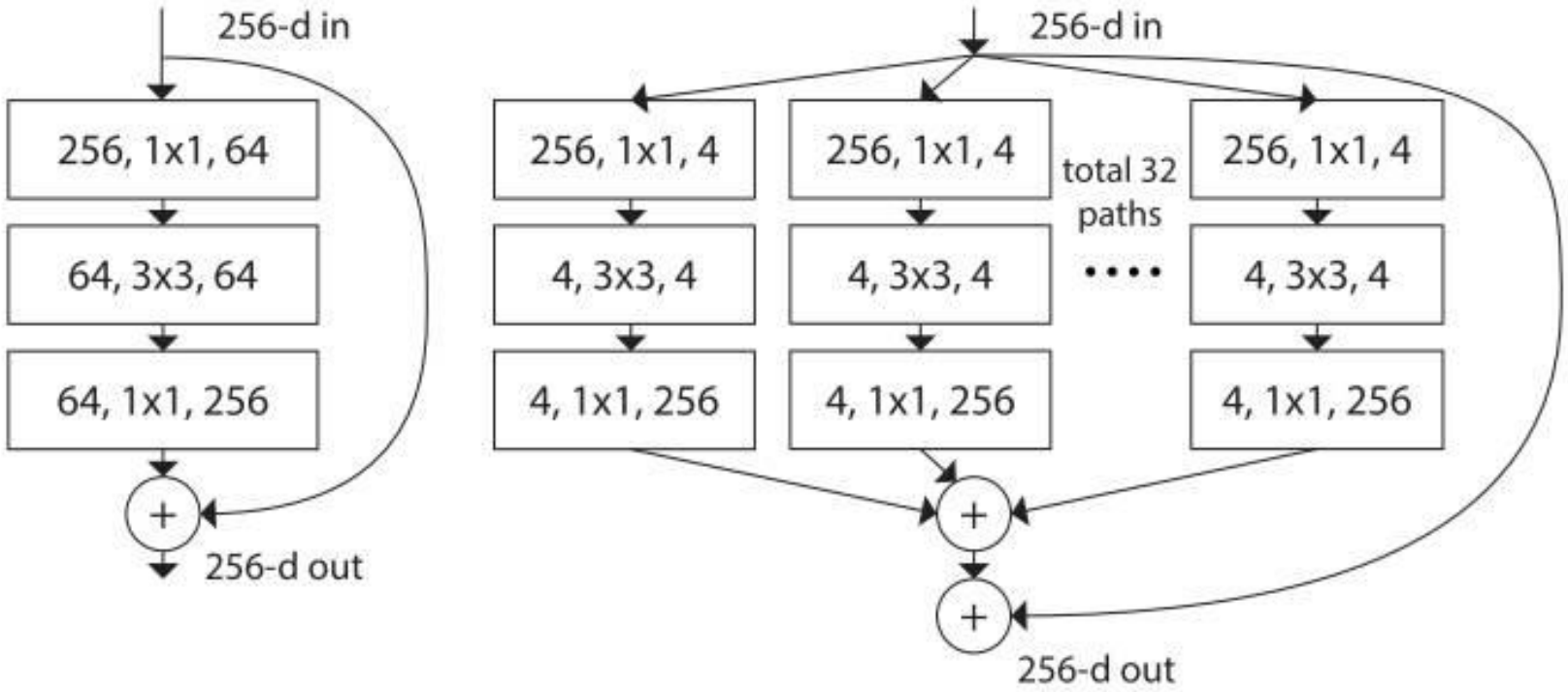


(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

Figure 1: Various residual blocks used in the paper. Batch normalization and ReLU precede each convolution (omitted for clarity)

Zagoruyko S, Komodakis N. Wide residual networks[J]. arXiv preprint arXiv:1605.07146, 2016.

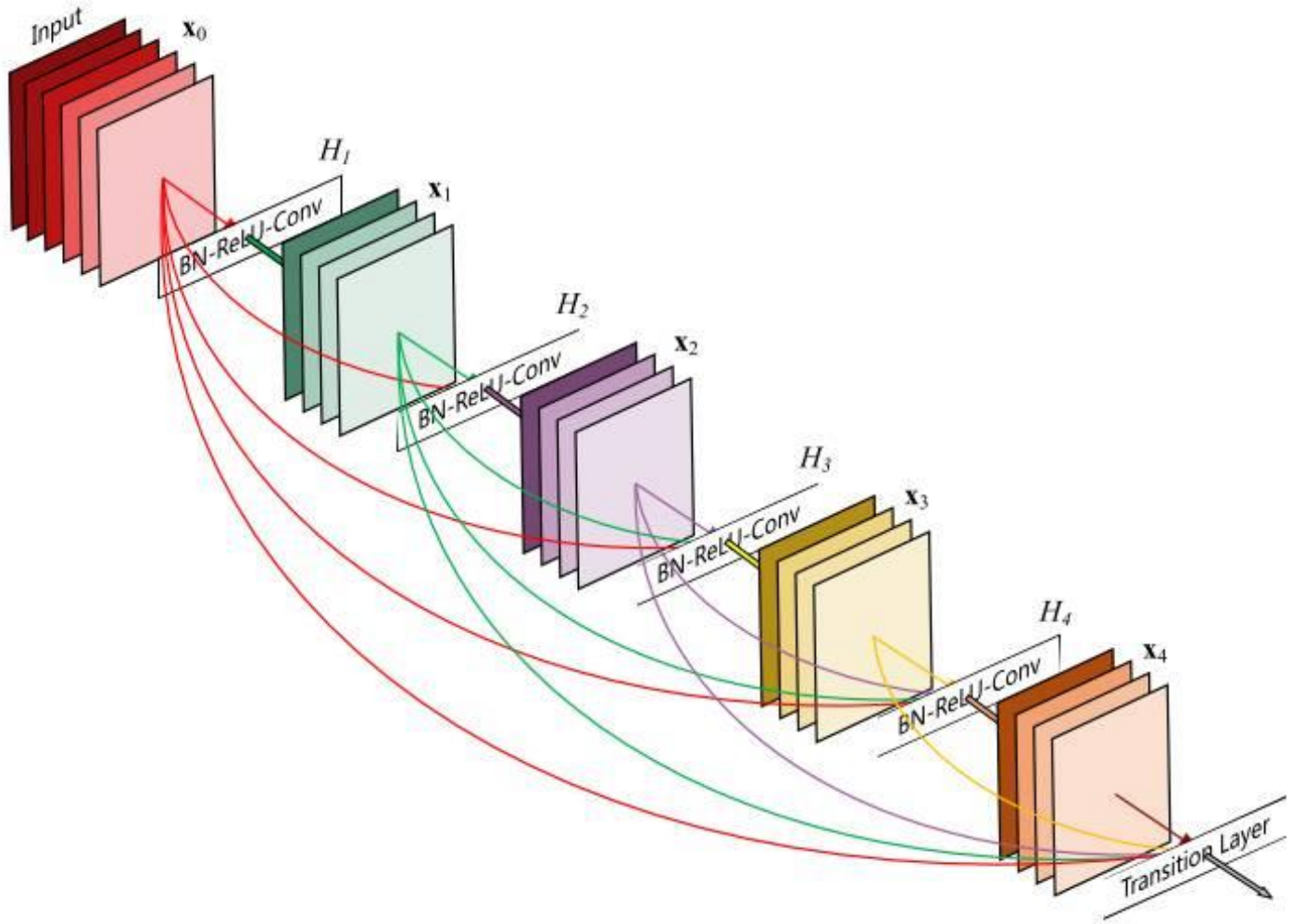
ResNeXt：



Left: building block of ResNet; right: a building block for ResNeXt, base number =32

Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.

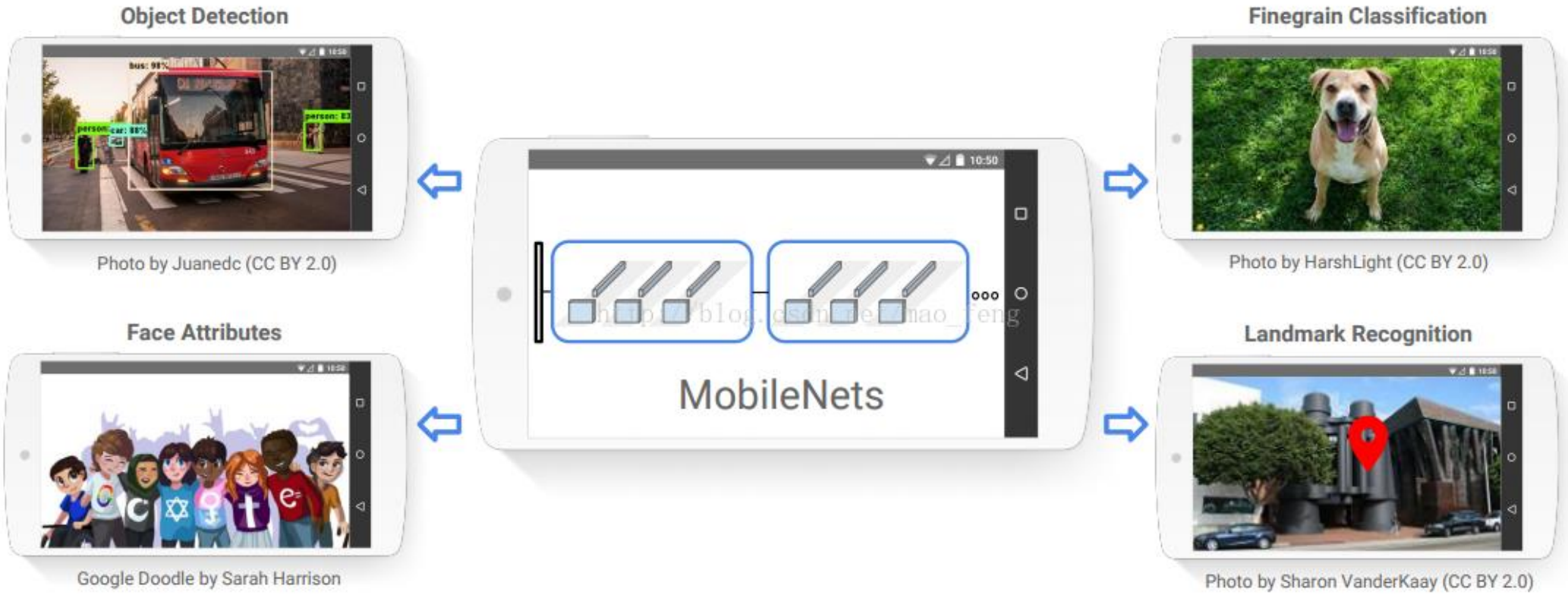
DenseNet: change the output from the addition to "phase parallel"



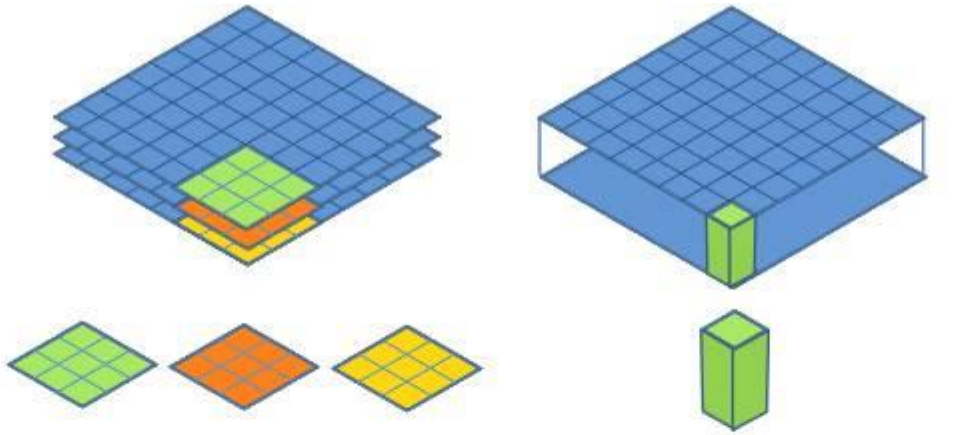
ResNet adds output to input to form a residual structure, while DenseNet outputs output parallel to input, so that each layer can directly get output of all previous layers.

Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. Vol. 1. No. 2. 2017.

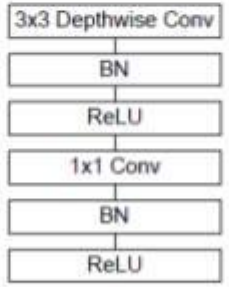
Google MobileNet: a lightweight development of the visual model to the mobile end



Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).



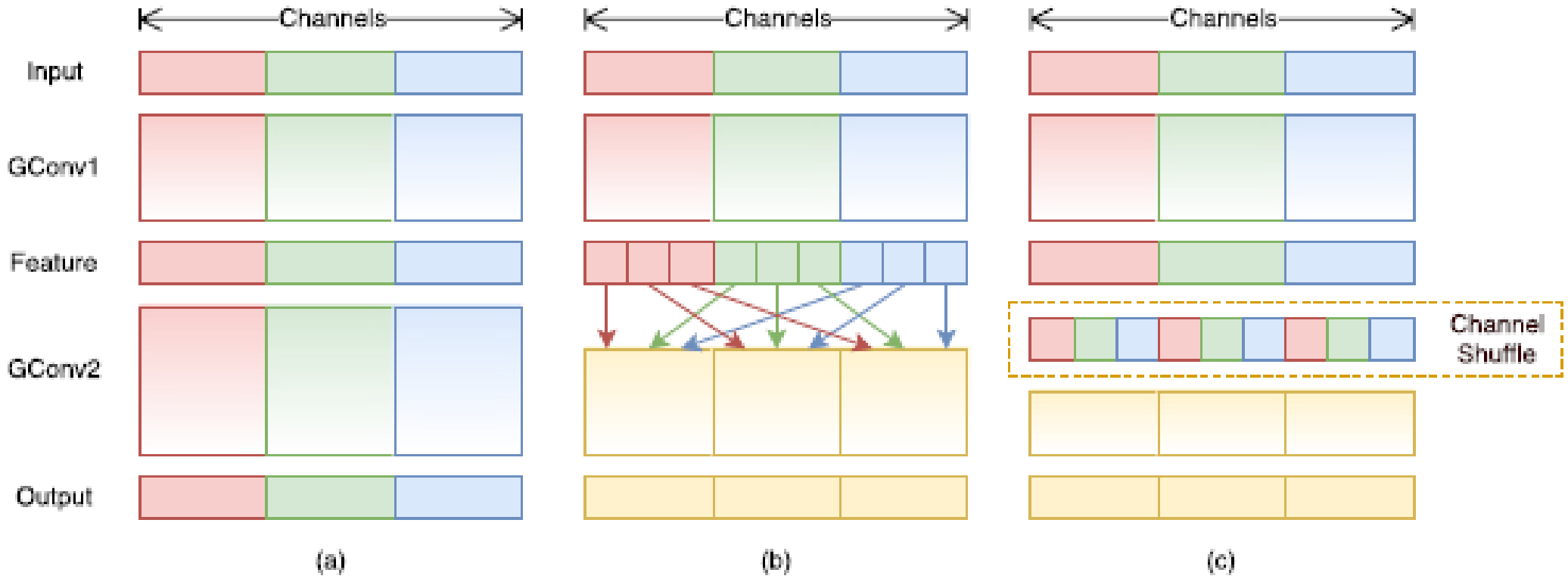
Depthwise Convolutional Filters　　Pointwise Convolutional Filters



Depthwise Separable Convolution

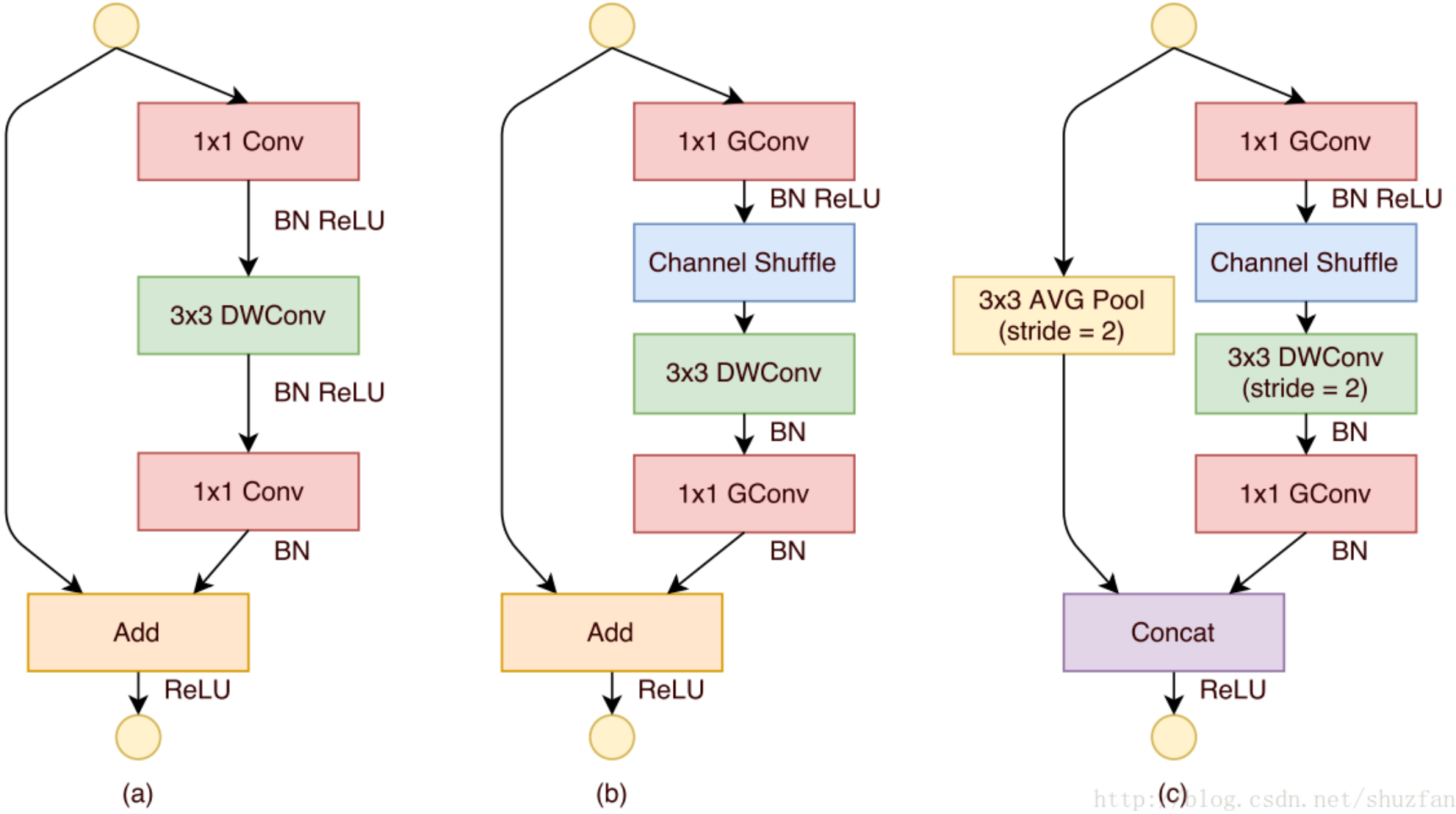
Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

Sun Jian team ShuffleNet: Surpassing Google MobileNet



Point by point convolution and channel rearrangement operation

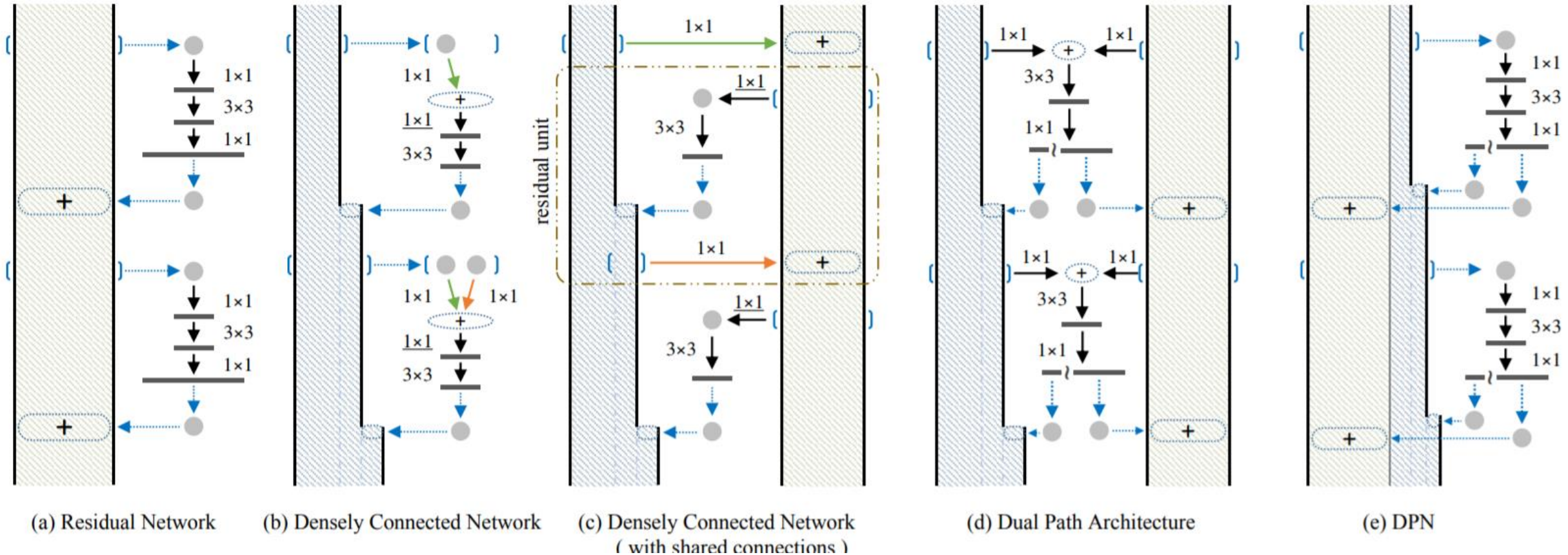
Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." arXiv preprint arXiv:1707.01083 (2017).



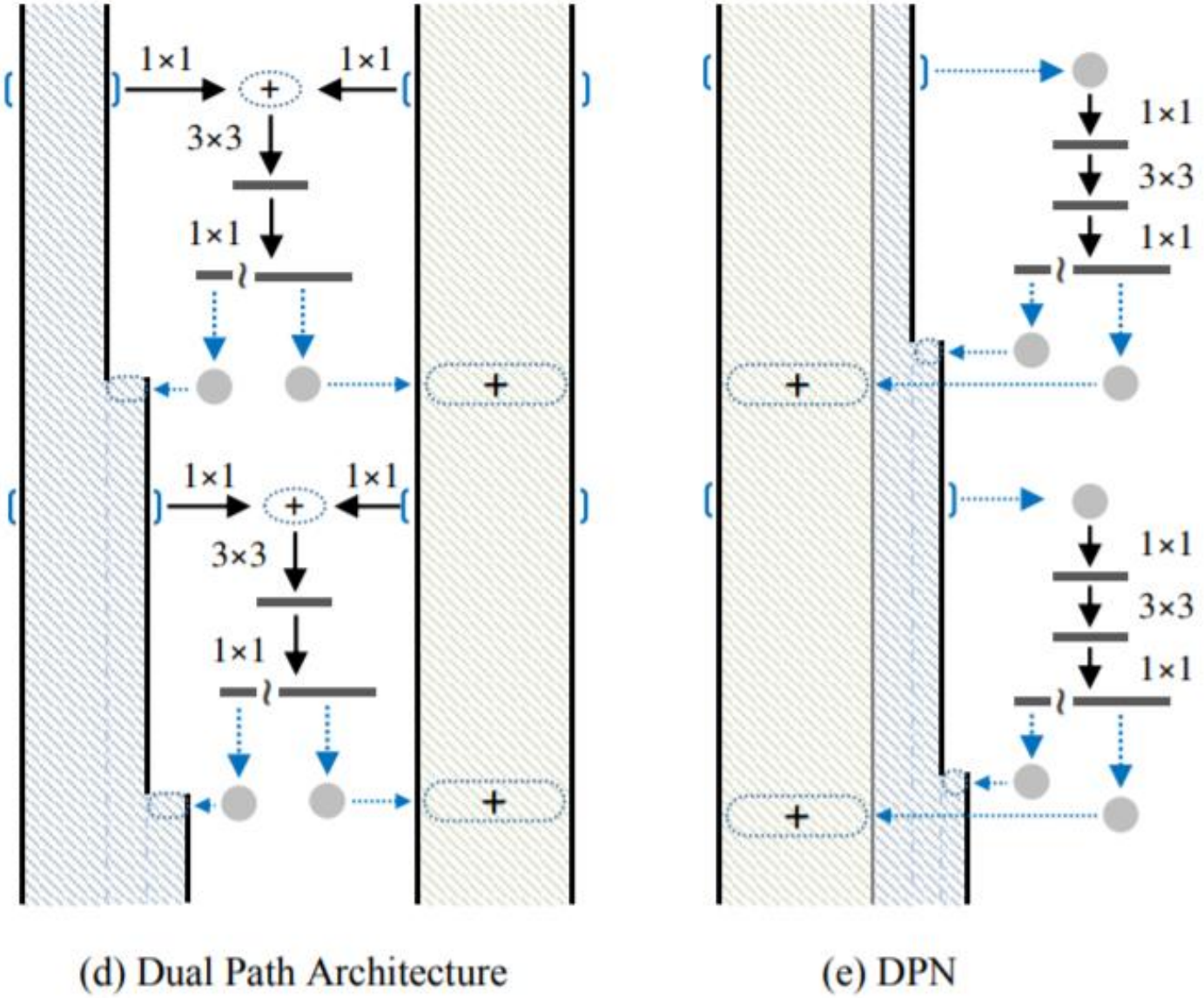
Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." arXiv preprint arXiv:1707.01083 (2017).

DPN

Architecture comparison of different networks



Chen, Yunpeng, et al. "Dual path networks." Advances in Neural Information Processing Systems. 2017.



(d) Dual Path Architecture (e) DPN

Chen, Yunpeng, et al. "Dual path networks." Advances in Neural Information Processing Systems. 2017.

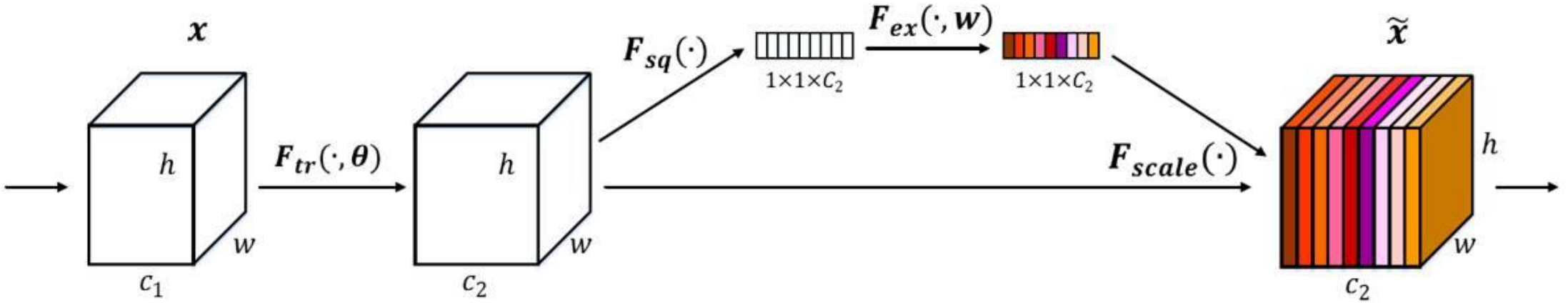
Squeeze-and-Excitation (SE) Networks

- If a network can be enhanced from the aspect of **channel relationship**?

- **Motivation:**
  - Explicitly model channel-interdependencies within modules
  - Feature recalibration
    - Selectively enhance useful features and suppress less useful ones

Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." arXiv preprint arXiv:1709.01507 (2017).

Squeeze-and-Excitation Module



Squeeze

- Shrinking feature maps $\in \mathbb{R}^{w\times h\times c_2}$ through spatial dimensions $(w \times h)$
- Global distribution of channel-wise responses

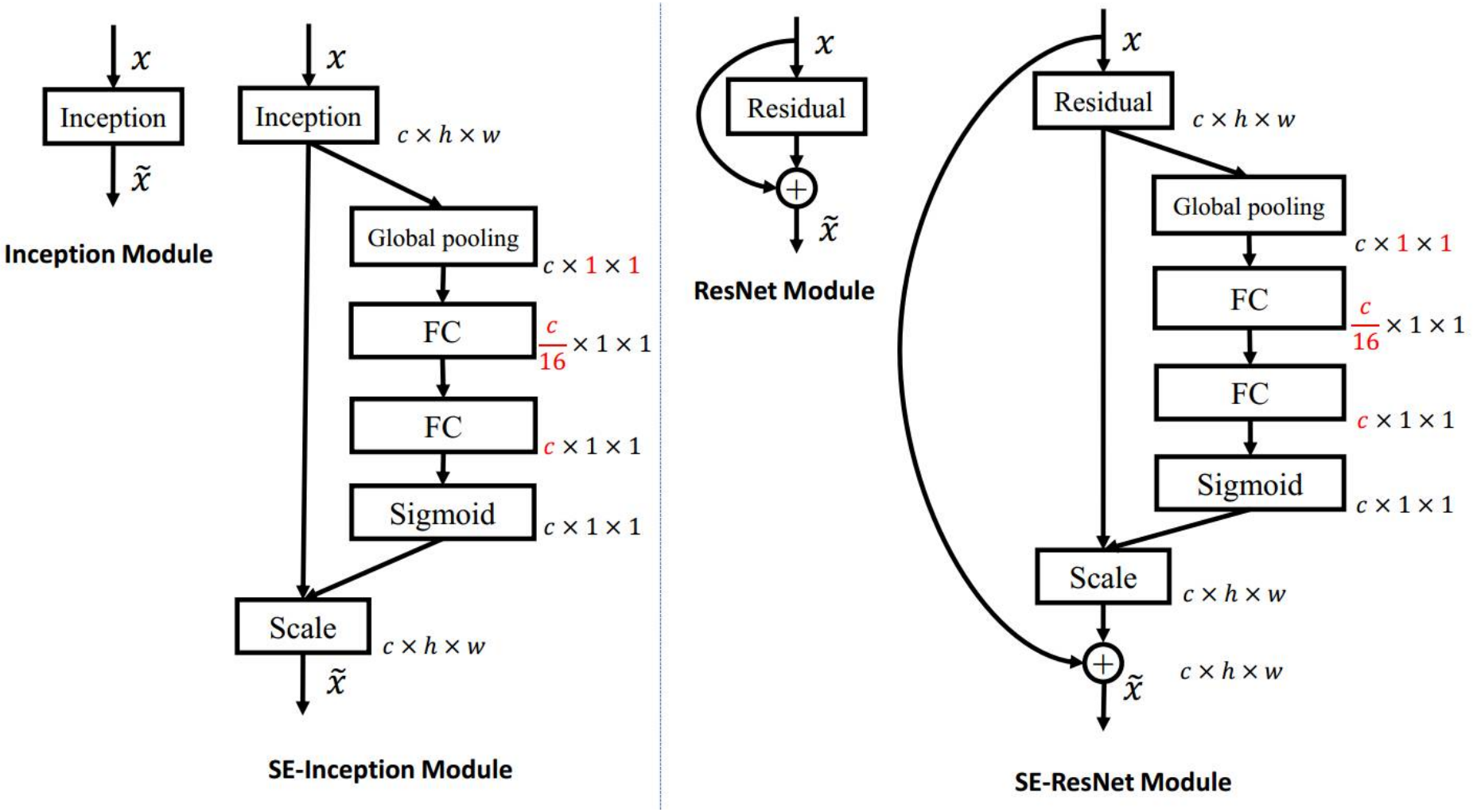
Excitation

- Learning $W \in \mathbb{R}^{c_2\times c_2}$ to explicitly model channel-association
- Gating mechanism to produce channel-wise weights

Scale

- Reweighting the feature maps $\in \mathbb{R}^{w\times h\times c_2}$

Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." arXiv preprint arXiv:1709.01507 (2017).



On the left is an example of a SE module embedded in the Inception structure. The dimension information next to the box represents the output of the layer.

Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." arXiv preprint arXiv:1709.01507 (2017).

Model and Computational Complexity

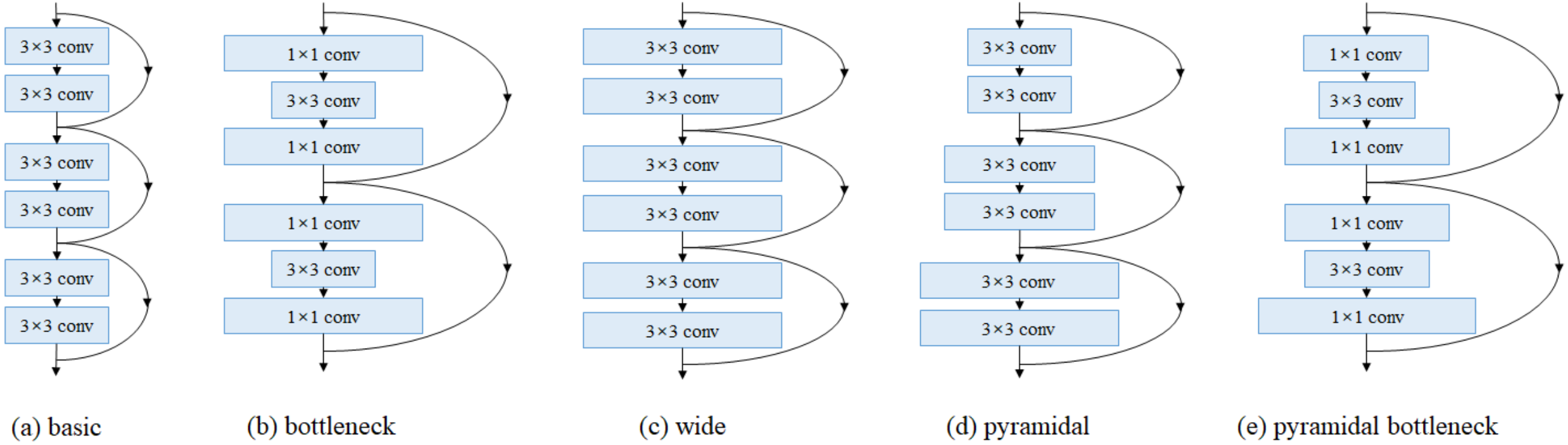
SE-ResNet-50 vs. ResNet-50

- Parameters: 2%~10% additional parameters
- Computation cost: <1% additional computation (theoretical)
- GPU inference time: 10% additional time
- CPU inference time: <2% additional time

Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." arXiv preprint arXiv:1709.01507 (2017).

PyramidNet

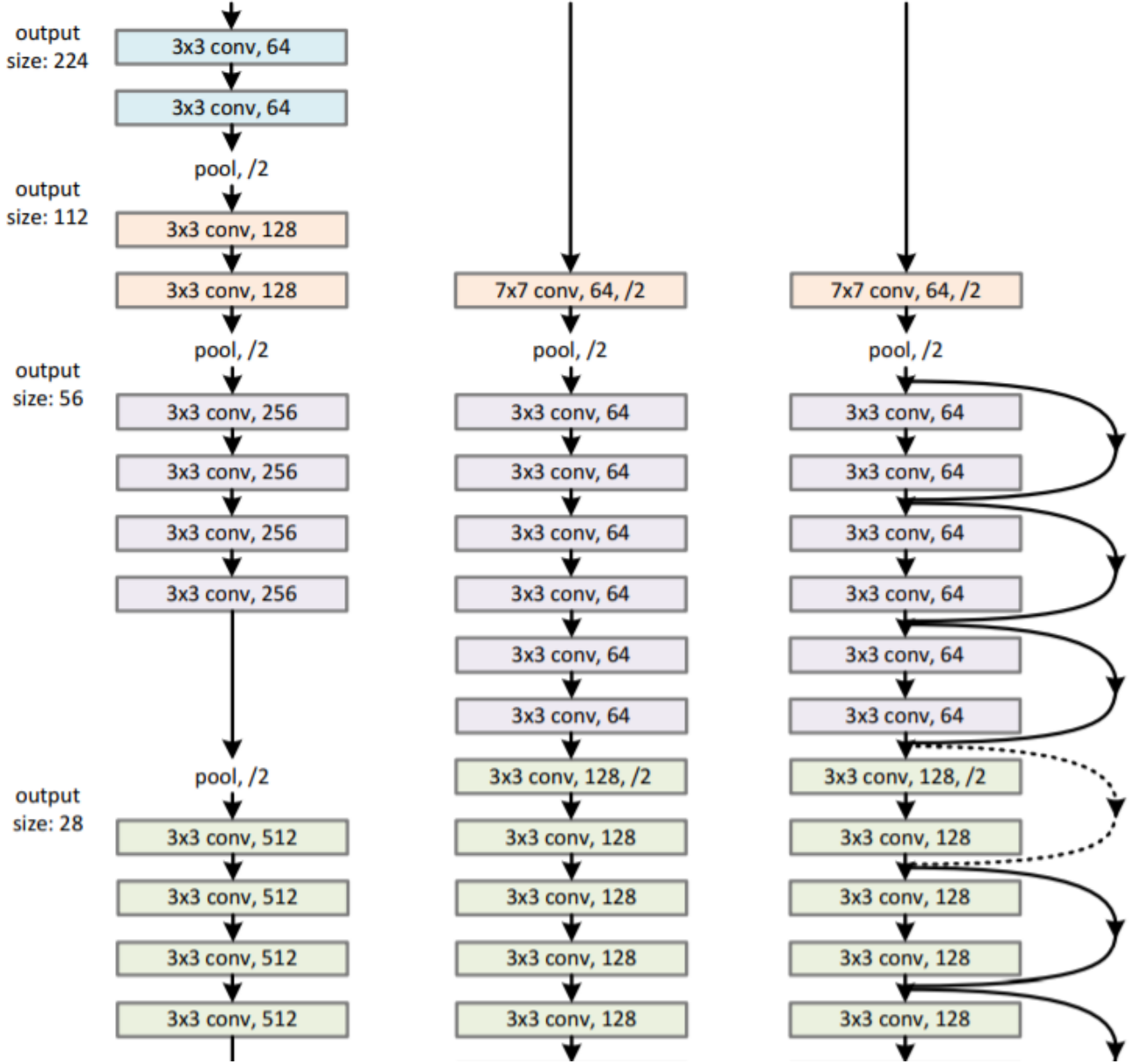
Network architecture details



(a) basic (b) bottleneck (c) wide (d) pyramidal (e) pyramidal bottleneck

Schematic illustration of comparision of several units: (a) basic residual units, (b) bottleneck, (c) wide residual units, and (d) our pyramidal residual units, and (e) our pyramidal bottleneck residual units:

Han, Dongyoon, Jiwhan Kim, and Junmo Kim. "Deep pyramidal residual networks." Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.



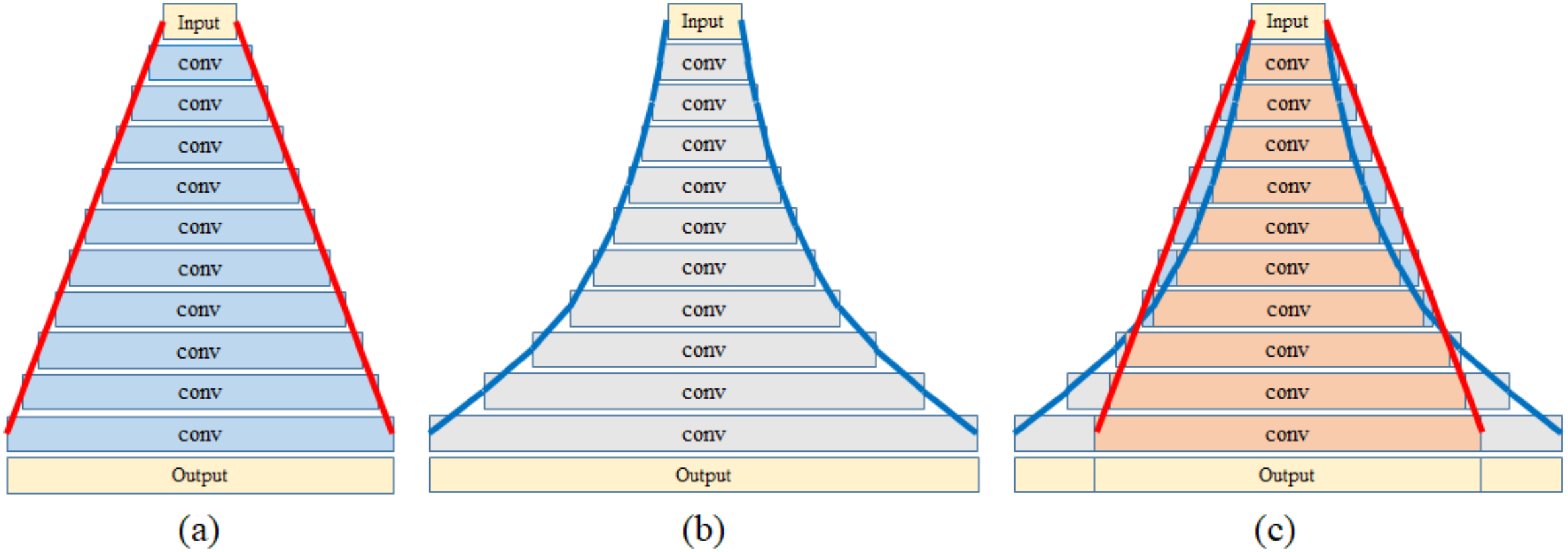
He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

$$D_k = \begin{cases} 16, & \text{if } n(k) = 1, \\ 16 \cdot 2^{n(k)-2}, & \text{if } n(k) \geq 2, \end{cases} \quad (1)$$

$$D_k = \begin{cases} 16, & \text{if } k = 1, \\ \lfloor D_{k-1} + \alpha/N \rfloor, & \text{if } 2 \leq k \leq N+1, \end{cases} \quad (2)$$

$$D_k = \begin{cases} 16, & \text{if } k = 1, \\ \lfloor D_{k-1} \cdot \alpha^{\frac{1}{N}} \rfloor, & \text{if } 2 \leq k \leq N+1. \end{cases} \quad (3)$$

Han, Dongyoon, Jiwhan Kim, and Junmo Kim. "Deep pyramidal residual networks." Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.



Visual illustration of (a) additive PyramidNet (the feature map dimension of each unit increases linearly), (b) multiplicative PyramidNet (the feature map dimension of each unit increases geometrically), and (c) comparison of (a) and (b):

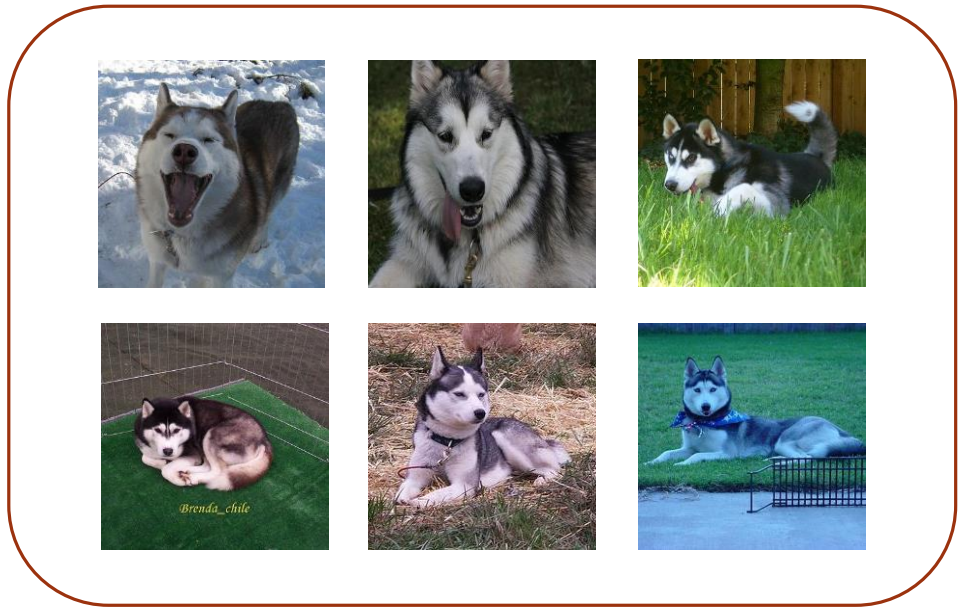
Han, Dongyoon, Jiwhan Kim, and Junmo Kim. "Deep pyramidal residual networks." Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.

The main research direction of image classification

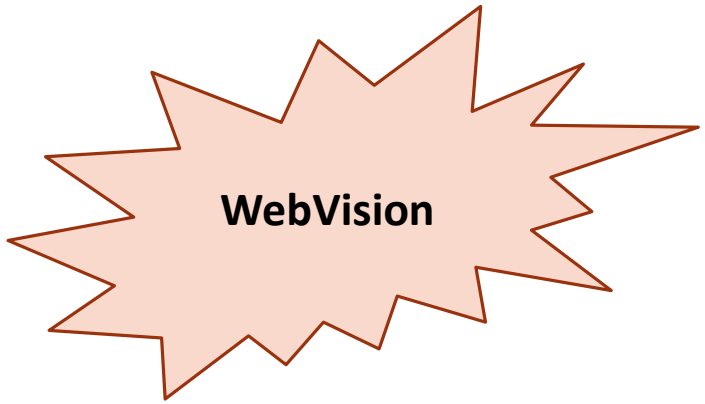
Fine-grained image classification

Fine-grained image classification

aims to distinguish subordinate-level categories which are very similar.



Imbalanced Image Classification



1. Noisy
2. Large
3. imbalanced

How to deal with the imbalance problem?

Table of Content

1. Classical method
   Data level approach
   Algorithm level approach
2. Methods about convolutional neural network

Evaluation Indexes

Machine learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced datasets. The conventional model evaluation methods do not accurately measure model performance when faced with imbalanced datasets.

Actual	Predicted	
	Positive Class	Negative Class
Positive Class	True Positive(TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

• True Positive (TP) refers to the number of positive examples which are correctly predicted as positives by a classifier

• True Negative (TN) denotes as the number of negative examples correctly classified as negatives by a classifier

• False Positive (FP), often referred to as false alarm; defines as the number of negative examples incorrectly classified as positives by a classifier

• False Negative (FN), sometimes known as miss; is determined as the number of positive examples incorrectly assigned as negatives by a classifier

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F\text{-}Measure = \frac{(1 + \beta)^2 \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision}$$

$$G\text{-}mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}.$$

$$Accuracy = \frac{TP + TN}{P_C + N_C}; \quad ErrorRate = 1 - accuracy.$$

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Approach to handling imbalanced datasets:

Data level approach: Resampling Techniques

Dealing with imbalanced datasets entails strategies such as improving classification algorithms or balancing classes in the training data (data preprocessing) before providing the data as input to the machine learning algorithm.

The main objective of balancing classes is to either **increasing the number of the minority class samples** or **decreasing the number of the majority class samples**. This is done in order to obtain approximately the same number of samples for both the classes.

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Random Under-Sampling

Advantages:

It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge.

Disadvantages:

It can discard potentially useful information which could be important for building rule classifiers.
The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set.

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Random Over-Sampling

Advantages:

Unlike under sampling this method leads to no information loss.
Outperforms under sampling.

Disadvantages:

It increase the likelihood of overfitting since it replicates
the minority class samples.

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Cluster-Based Over Sampling

In this case, the K-means clustering algorithms is independently applied to minority and majority class samples. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of samples and all classes have the same size.

After oversampling of each cluster, all clusters of the same class contain the same number of samples.

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Advantages:

This clustering technique helps overcome the challenge between class imbalance. Where the number of examples representing positive class differs from the number of examples representing a negative class.
Also, overcome challenges within class imbalance, where a class is composed of different sub clusters. And each sub cluster does not contain the same number of examples.

Disadvantages:

The main drawback of this algorithm, like most oversampling techniques is the possibility of over-fitting the training data.

Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

Informed Over Sampling: Synthetic Minority Over-sampling Technique

Advantages:

Mitigates the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances

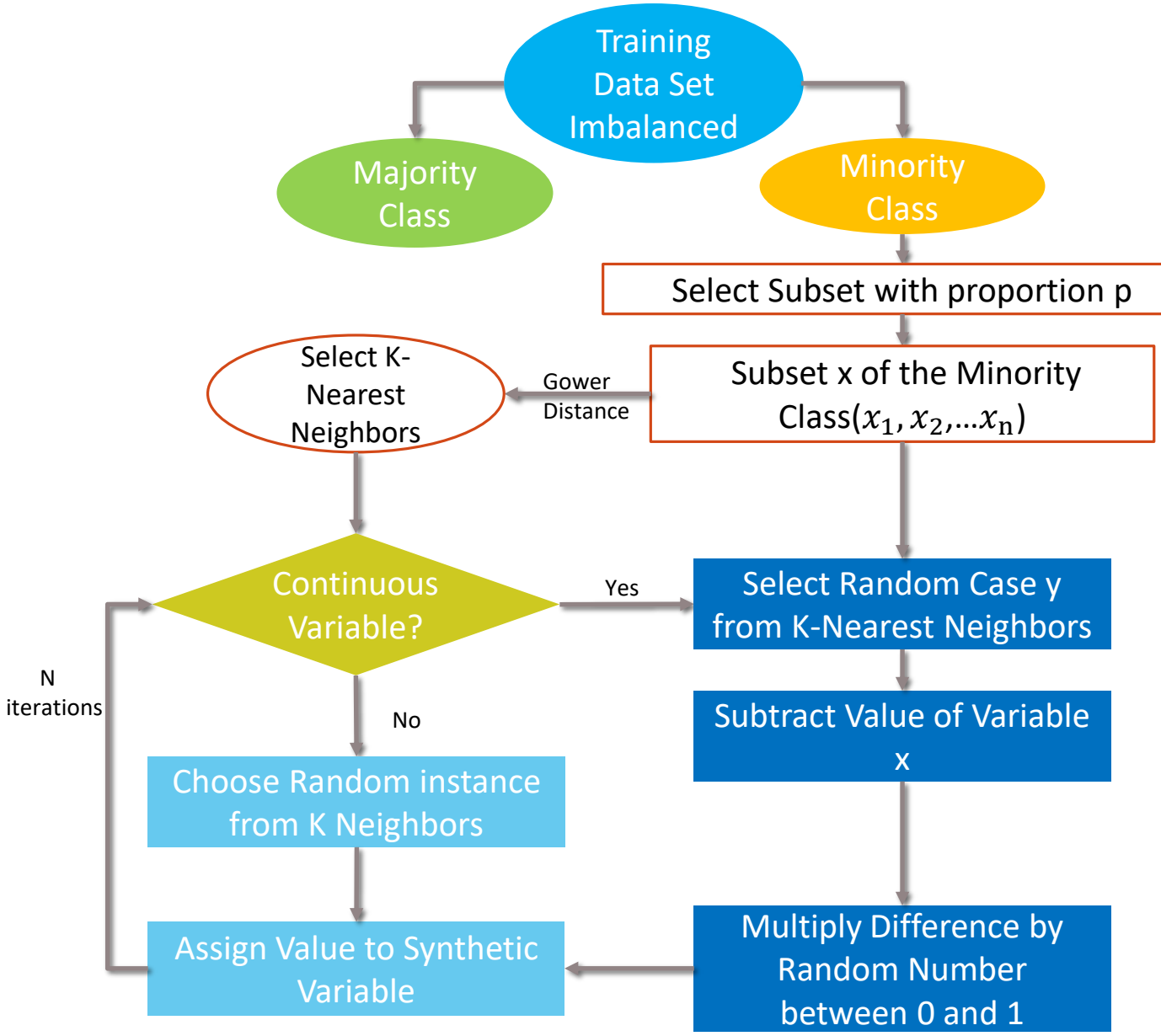
No loss of useful information

Disadvantages:

1、 While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise.

2、 SMOTE is not very effective for high dimensional data

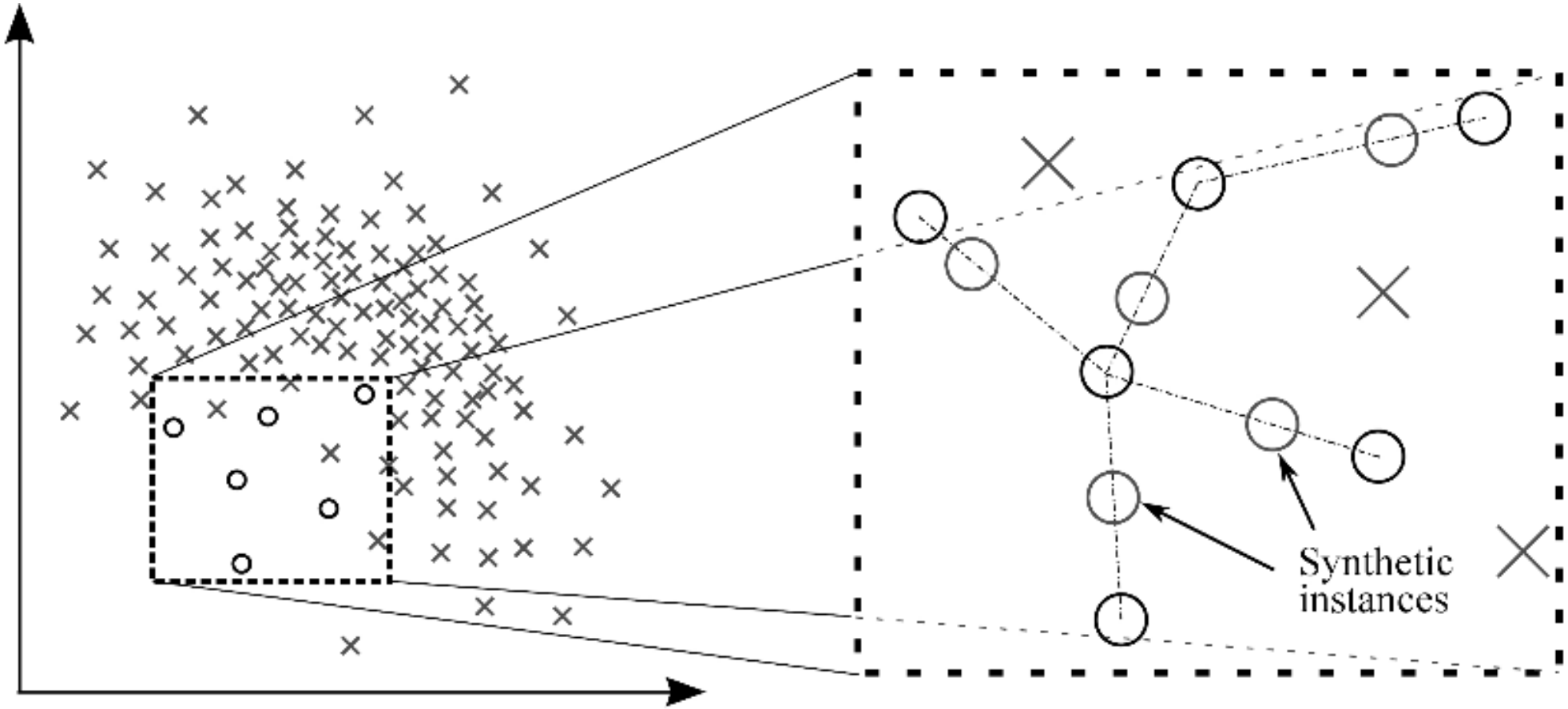
Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research16 (2002): 321-357.



N is the number of attributes

Synthetic Minority Oversampling Algorithm

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research16 (2002): 321-357.



Generation of Synthetic Instances with the help of SMOTE

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research16 (2002): 321-357.

Modified synthetic minority oversampling technique(MSMOTE)

1、It is a modified version of SMOTE.

2、This algorithm classifies the samples of minority classes into 3 distinct groups – Security/Safe samples, Border samples, and latent nose samples.

3、Security samples are those data points which can improve the performance of a classifier.

4、While the basic flow of MSOMTE is the same as that of SMOTE(discussed in the previous section).

Hu, Shengguo, et al. "MSMOTE: improving classification performance when training data is imbalanced." Computer Science and Engineering, 2009. WCSE'09. Second International Workshop on. Vol. 2. IEEE, 2009.

Algorithm level approach

1 Improved Algorithm

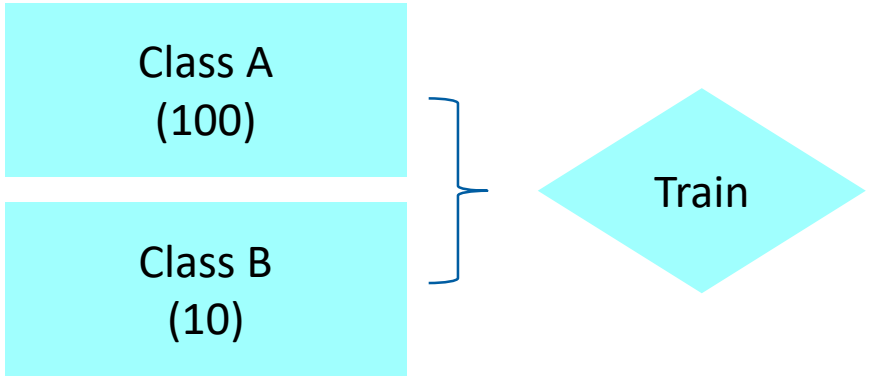
Improved SVM: z-SVM； GSVM-RU

Improved K-NN: k Examplar-based Nearest Neighbour(KENN)； A Class Conditional Nearest Neighbour Distribution(CCNND)

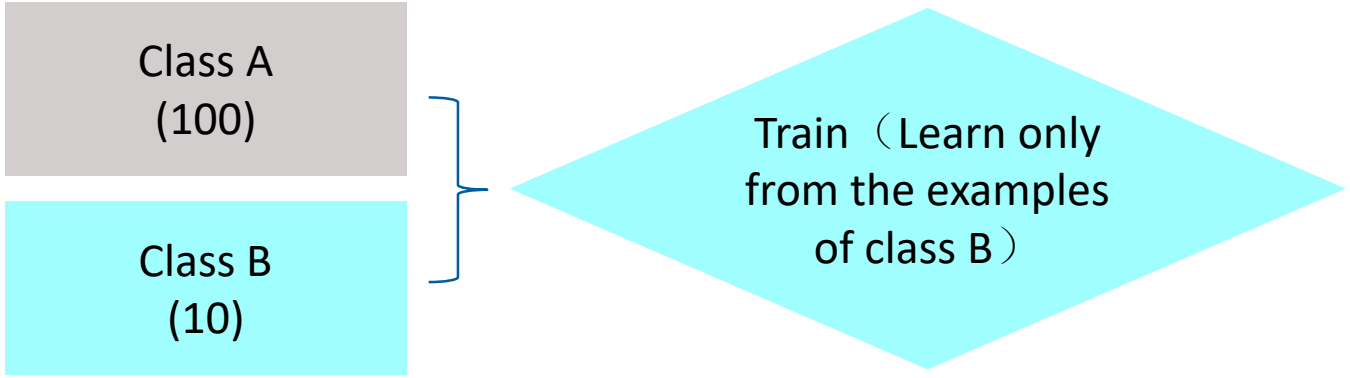
Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

2 One-class learning

How to make the classifier better learn the feature information of B?



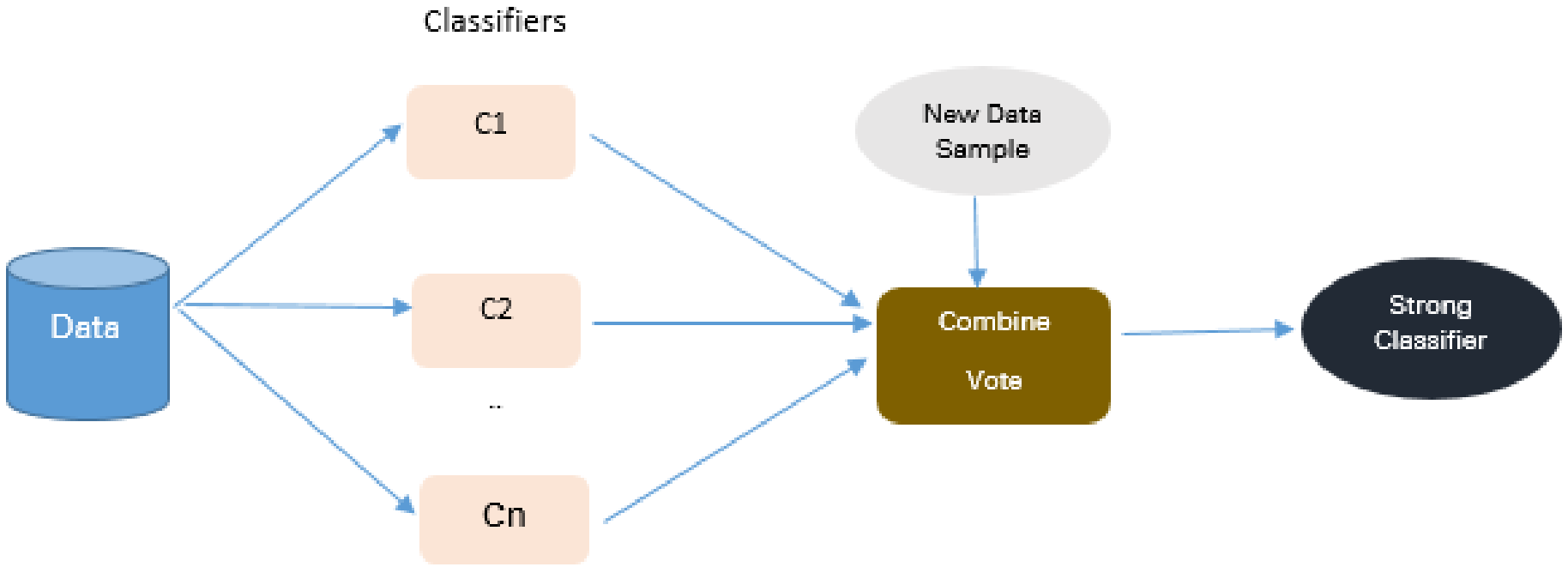
It tends to learn the features of A more likely and treat B as a noisy sample



Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem: a review." Int. J. Advance Soft Compu. Appl 7.3 (2015).

3 Cost-sensitive Learning

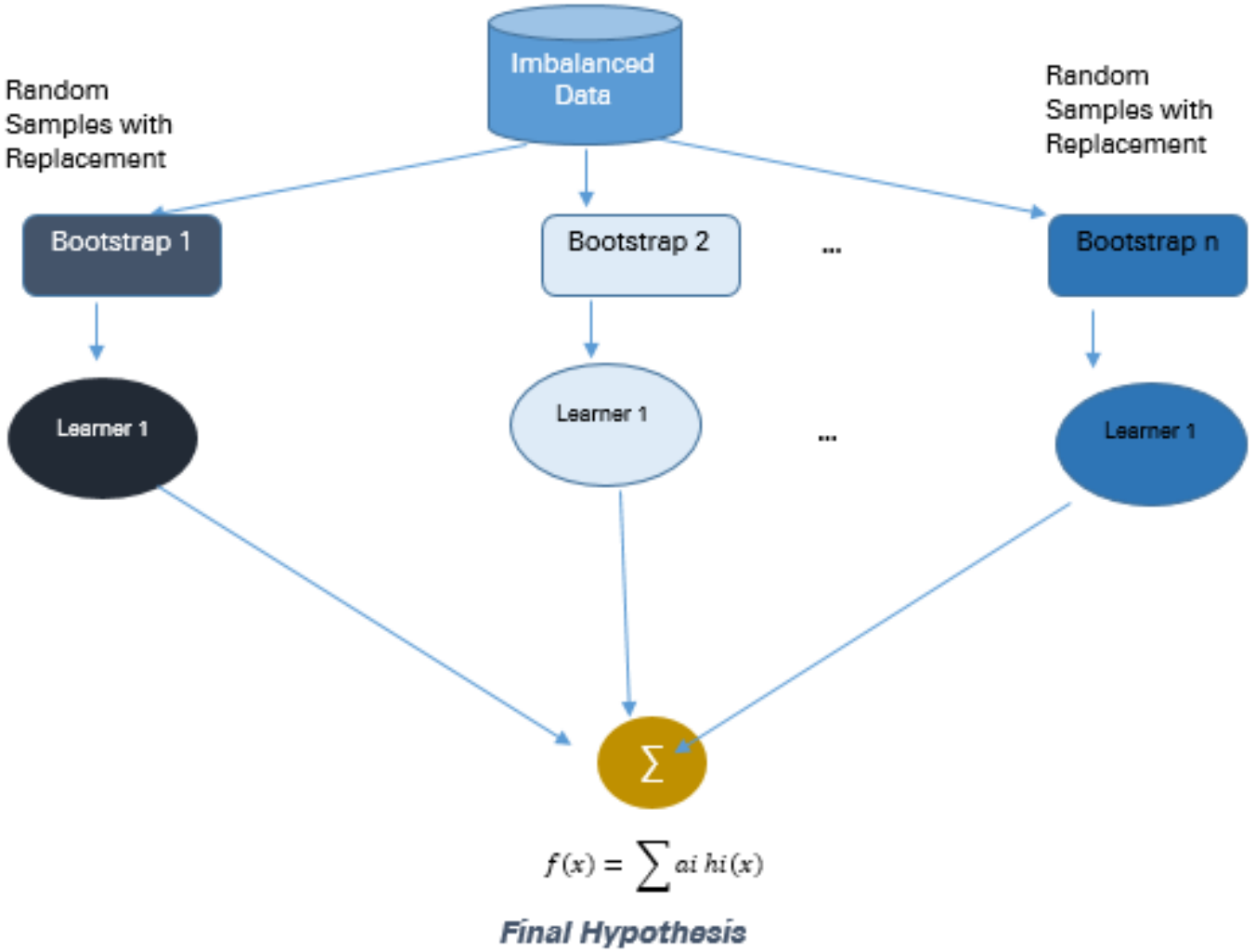
4 Ensemble Techniques



Approach to Ensemble based Methodologies

Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

Bagging Based



Approach to Bagging Methodology

Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

Advantages:

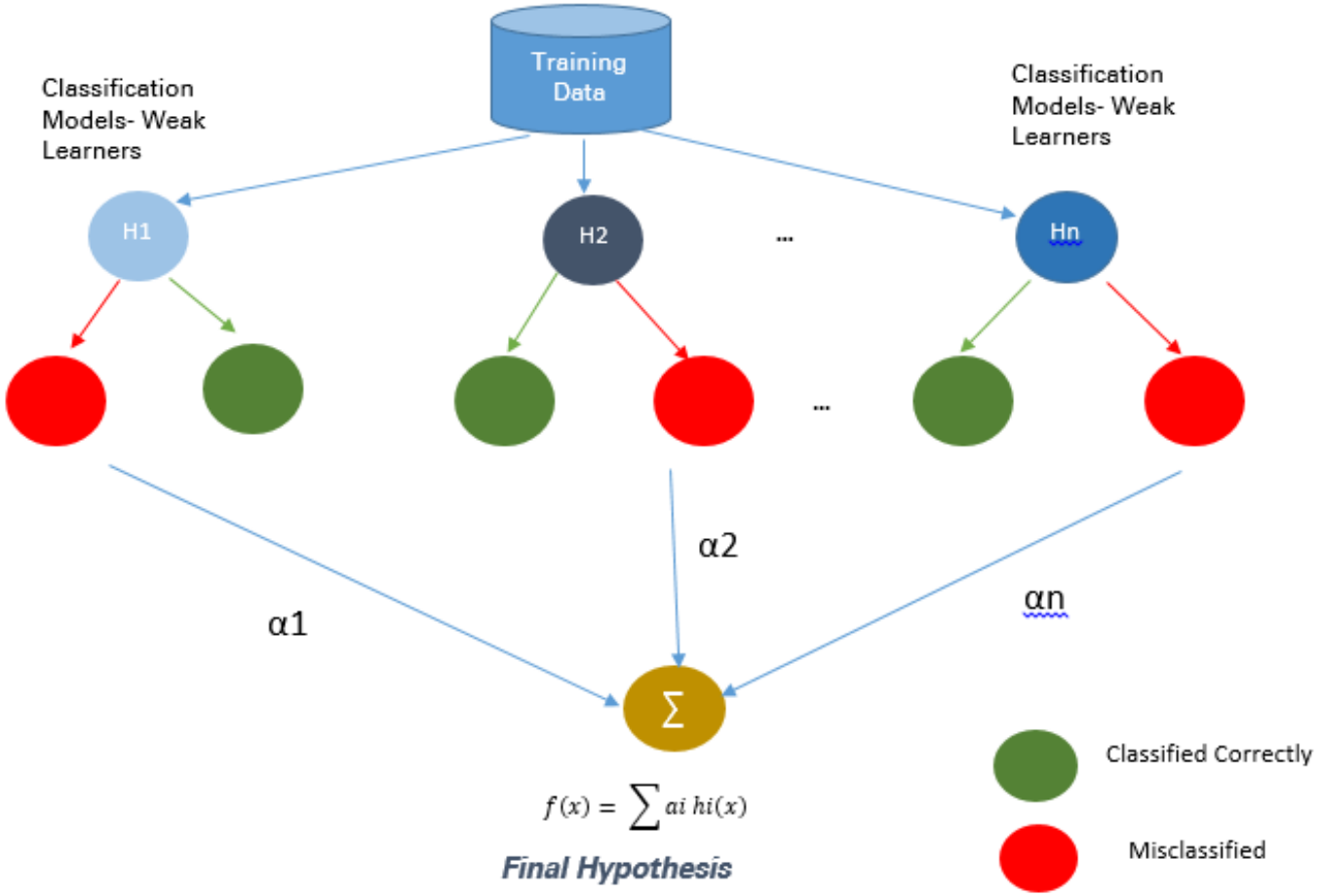
Improves stability & accuracy of machine learning algorithms
Reduces variance
Overcomes overfitting
Improved misclassification rate of the bagged classifier
In noisy data environments bagging outperform boosting

Disadvantages

Bagging works only if the base classifiers are not bad to begin with. Bagging bad classifiers can further degrade performance

Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

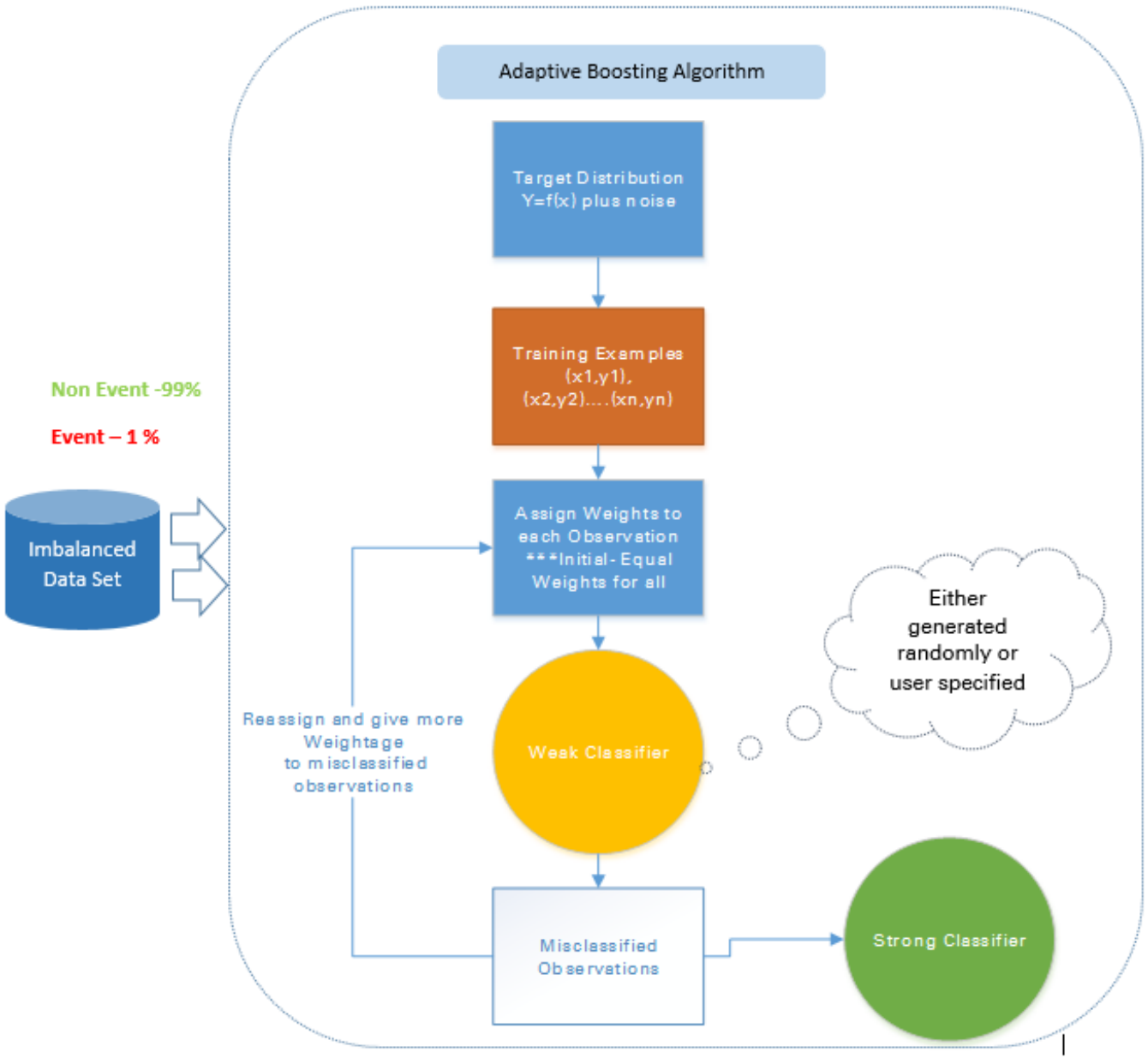
Boosting-Based



Approach to Boosting Methodologies

Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

Adaptive Boosting-Ada Boost



Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

Advantages:

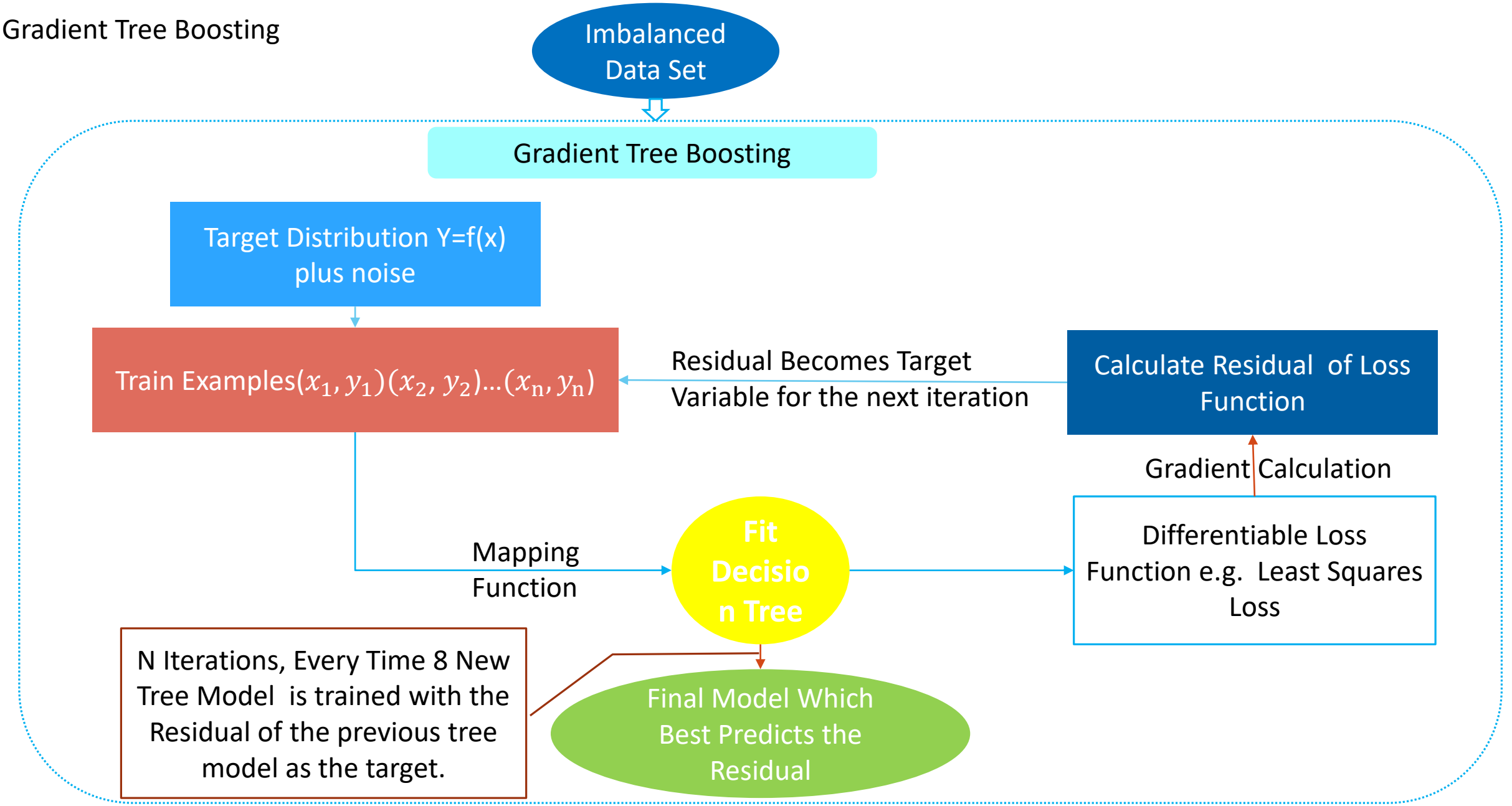
1. Very Simple to implement
2. Good generalization-suited for any kind of classification problem, not prone to overfitting

Disadvantages:

1. Sensitive to noisy data and outliers

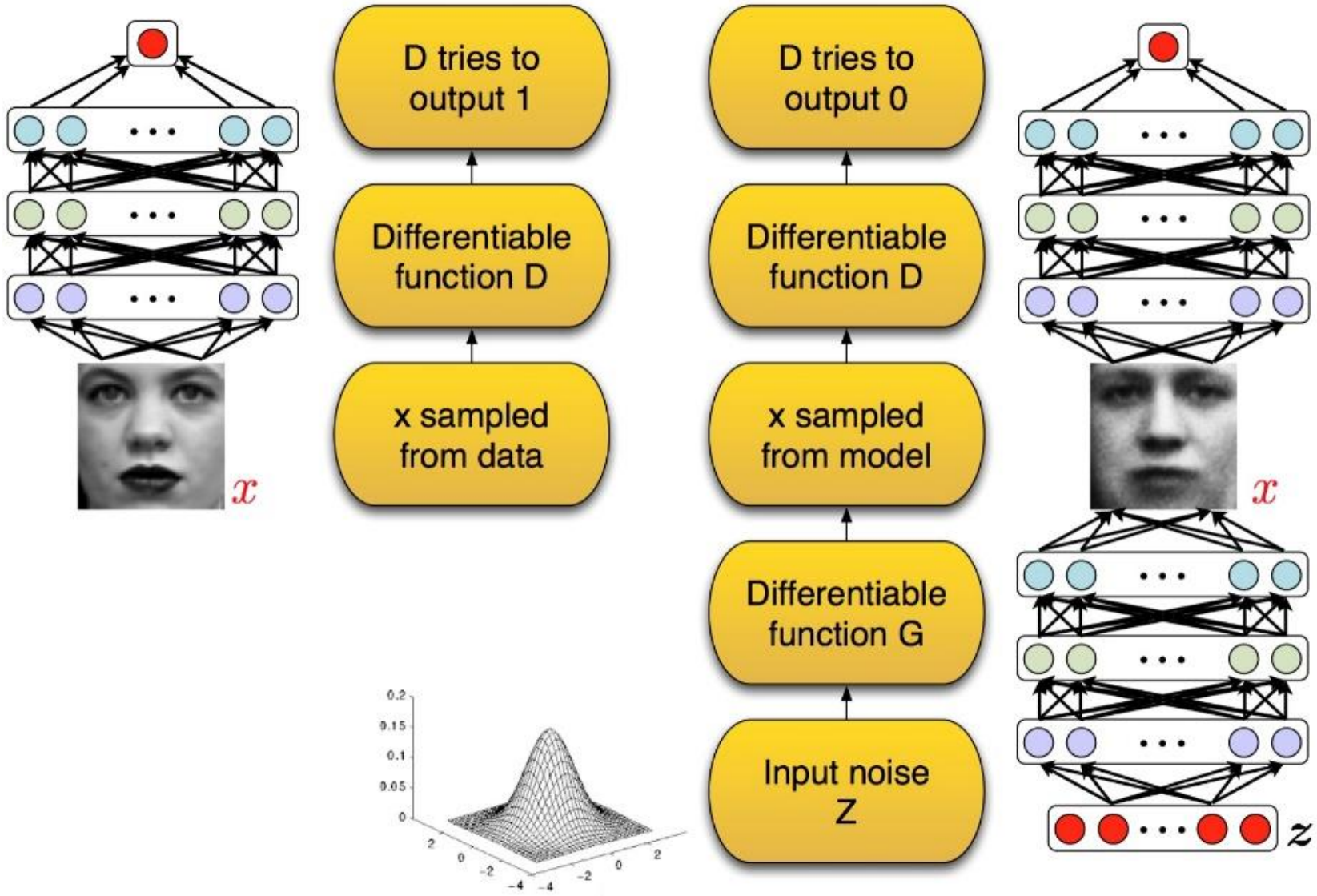
Galar, Mikel, et al. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.4 (2012): 463-484.

Gradient Tree Boosting

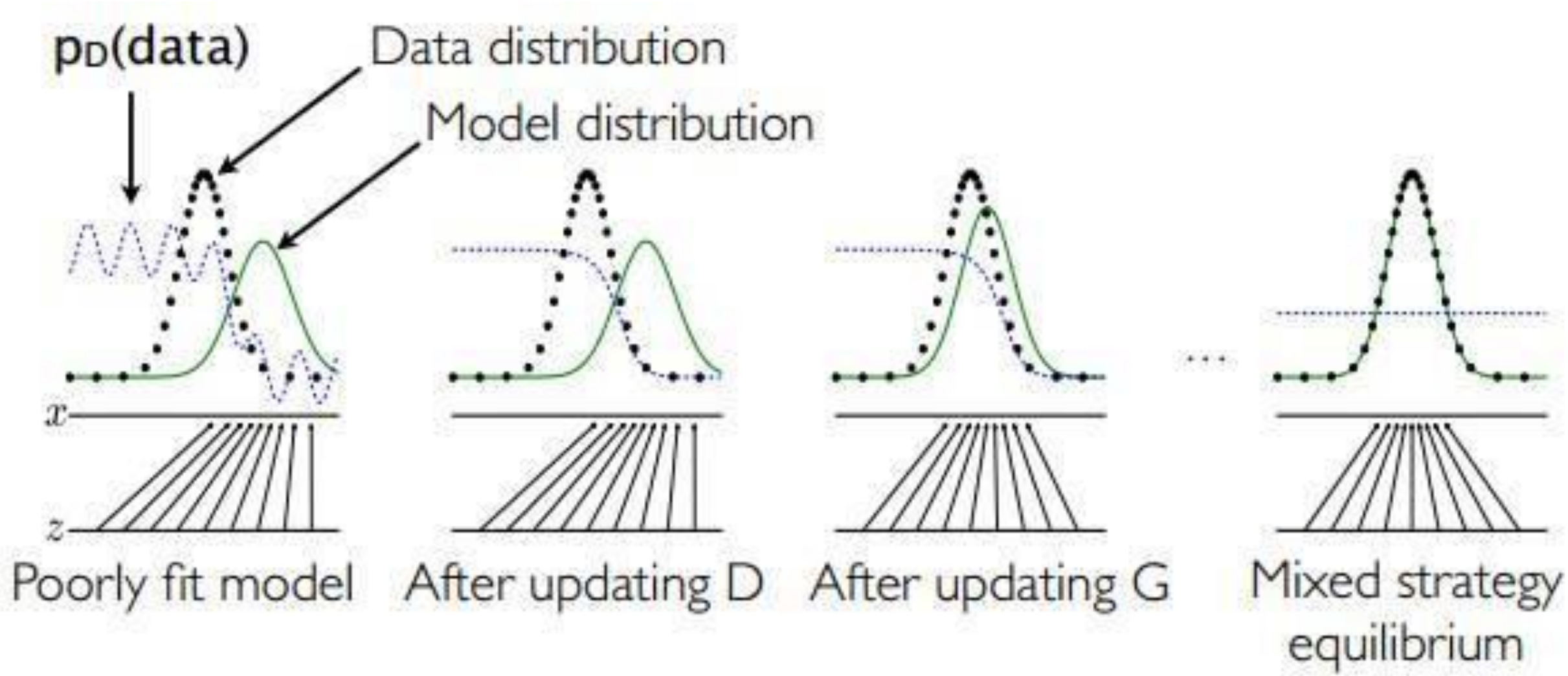


Methods about convolutional neural network

Use GAN to Generate Minority Samples



Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.



Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

Optimize the Loss of CNN Algorithms

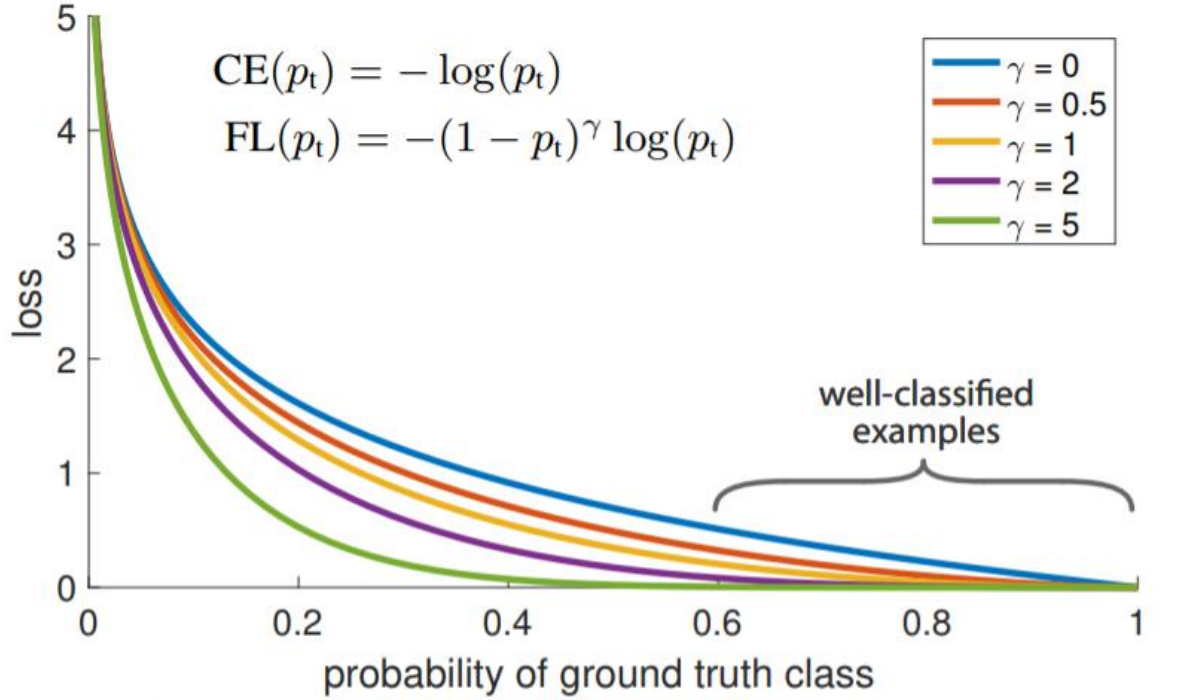


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." arXiv preprint arXiv:1708.02002 (2017).

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \tag{1}$$

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

and rewrite $\mathrm{CE}(p, y) = \mathrm{CE}(p_{\mathrm{t}}) = -\log(p_{\mathrm{t}})$.

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." arXiv preprint arXiv:1708.02002 (2017).

$$\mathrm{CE}(p_\mathrm{t}) = -\alpha_\mathrm{t} \log(p_\mathrm{t}). \tag{3}$$

Focal Loss Definition

$$\mathrm{FL}(p_\mathrm{t}) = -(1-p_\mathrm{t})^\gamma \log(p_\mathrm{t}). \tag{4}$$

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." arXiv preprint arXiv:1708.02002 (2017).

$$\mathrm{FL}(p_\mathrm{t}) = -\alpha_\mathrm{t}(1-p_\mathrm{t})^\gamma \log(p_\mathrm{t}). \quad (5)$$

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." arXiv preprint arXiv:1708.02002 (2017).

Thank you !