НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Алгоритмы компьютерного зрения на основе сверточных нейронных сетей

Савченко А.В.
Д.т.н., проф. Каф. Информационные системы и технологии
avsavchenko@hse.ru

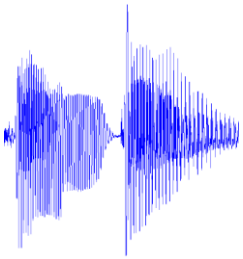
1. Deep learning
2. Сверточные нейронные сети
3. Современные архитектуры
4. Проблемы ConvNet
5. Детектирование объектов

Deep learning

Специально спроектированные признаки + обучаемый классификатор

Hand-crafted Feature Extractor → "Simple" Trainable Classifier →

Распознавание речи



MFCC → Mix of Gaussians → Classifier

fixed — unsupervised — supervised

Распознавание изображений

SIFT HoG → K-means Sparse Coding → Pooling → Classifier

fixed — unsupervised — supervised

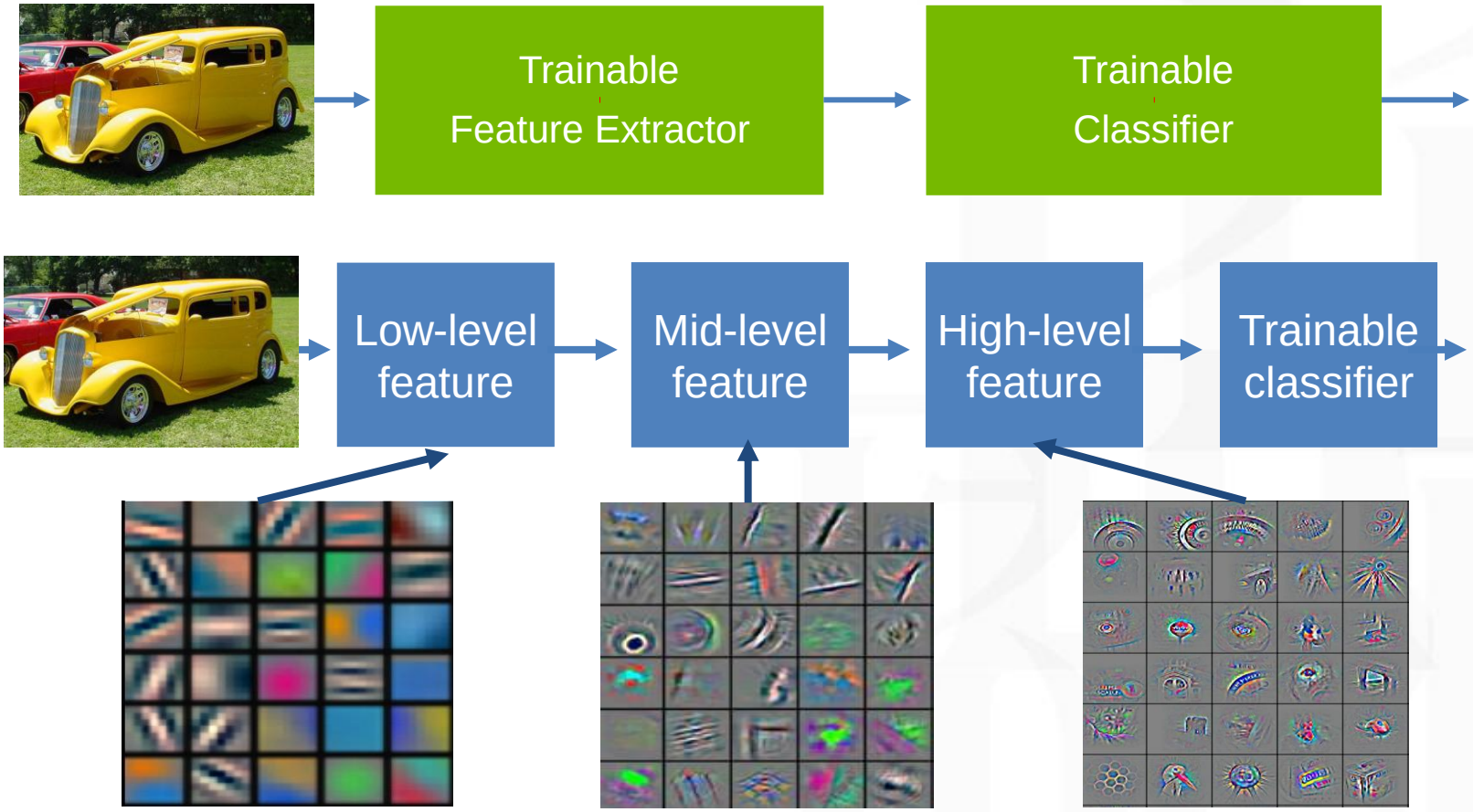
Low-level Features — Mid-level Features

DLI-Teaching Kit

– End-to-end learning / Feature learning / Deep learning

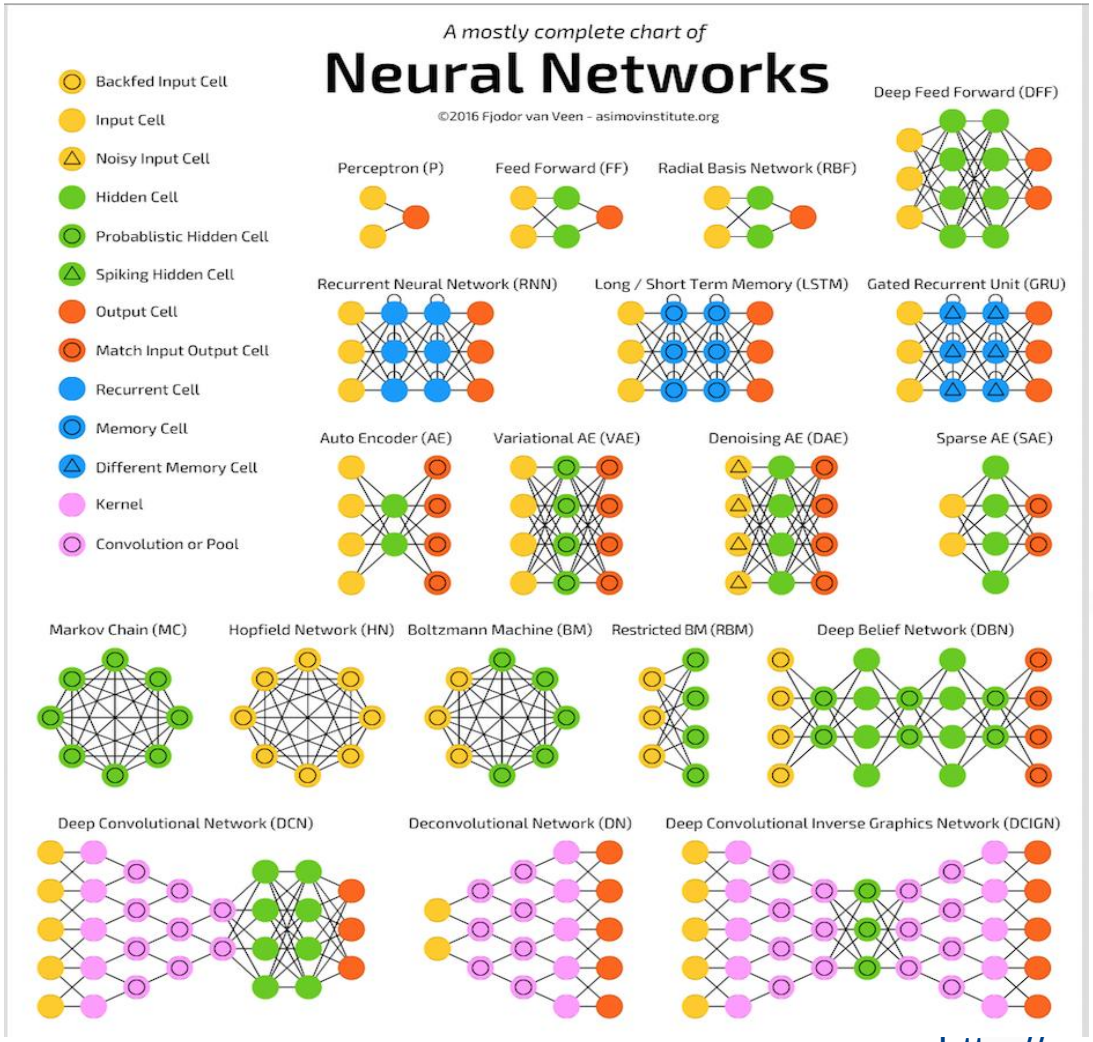
Характерные признаки являются обучаемыми:

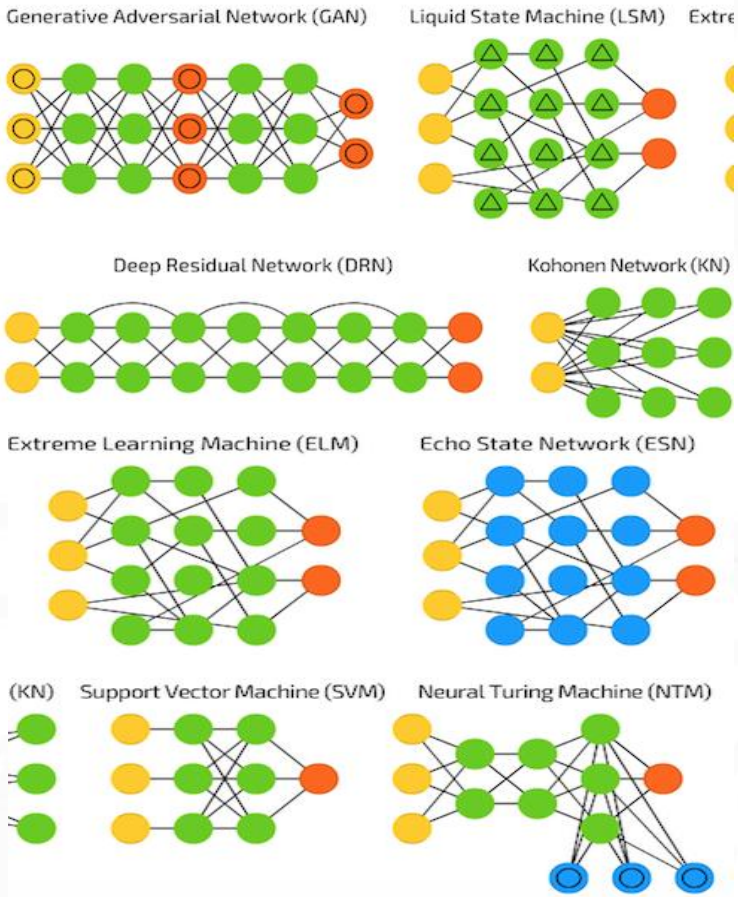
обучаемые признаки + обучаемый классификатор



DLI-Teaching Kit, Zeiler & Fergus 2013

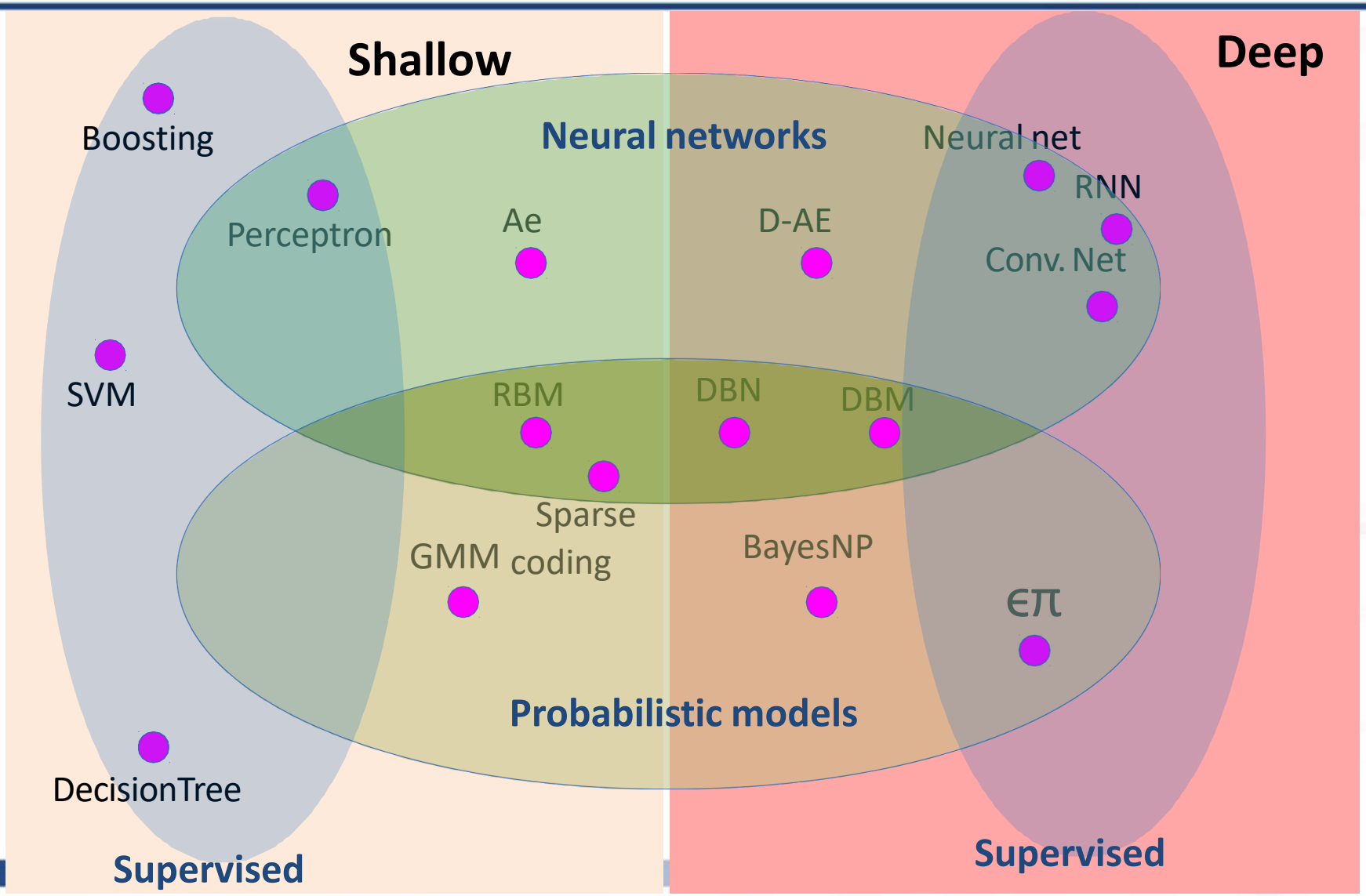
Многообразие архитектур (1)





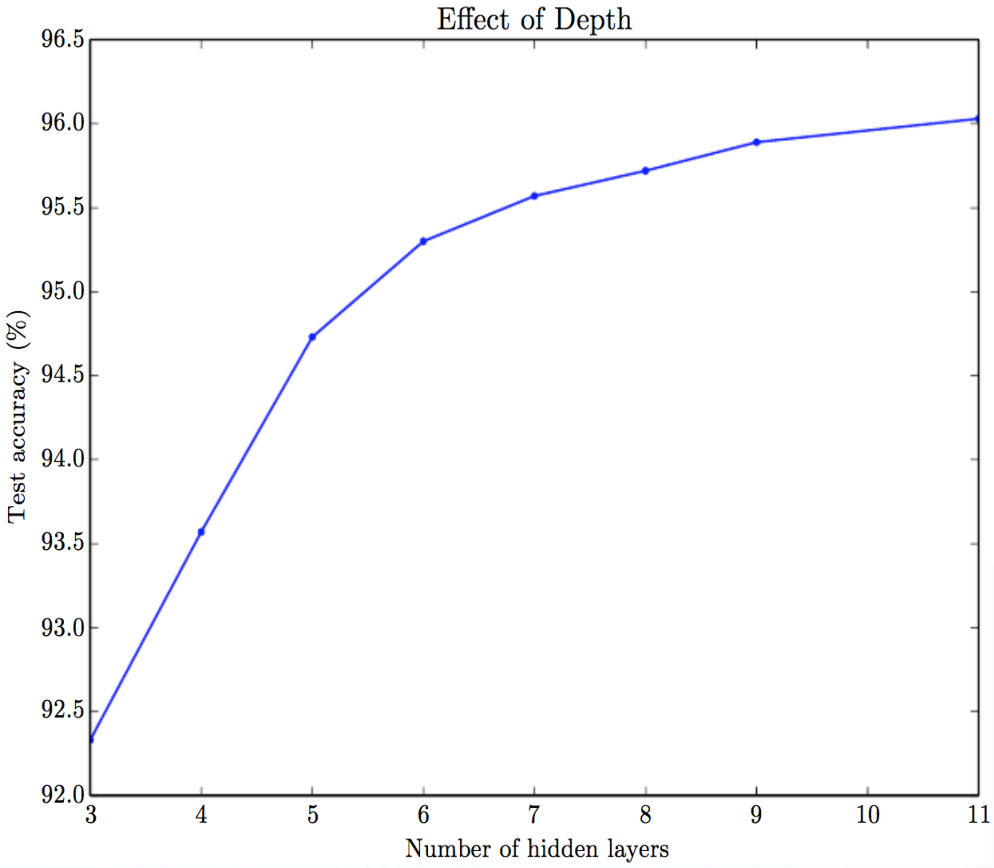
http://www.asimovinstitute.org/neural-network-zoo/

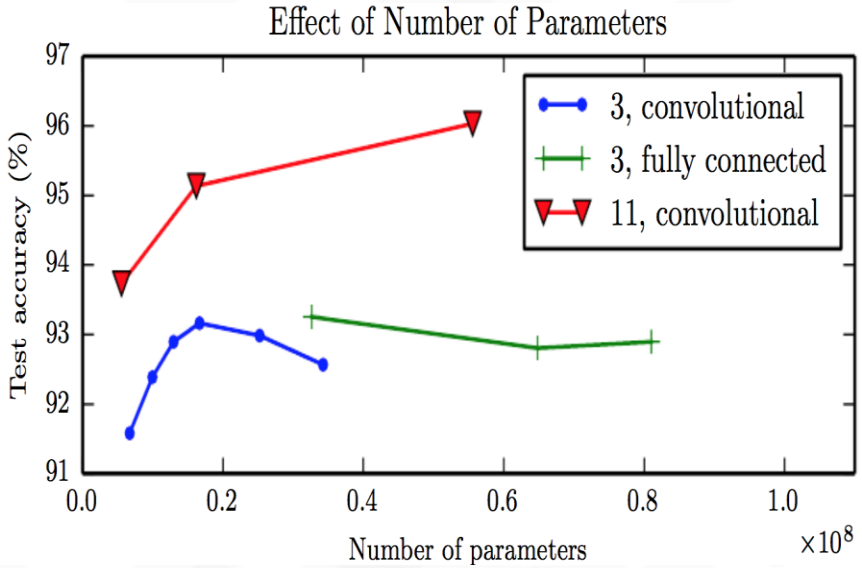
Многообразие архитектур (2)



Для теоремы универсальной аппроксимации может потребоваться экспоненциально большое число нейронов!

Пример. Классификация изображений номеров домов



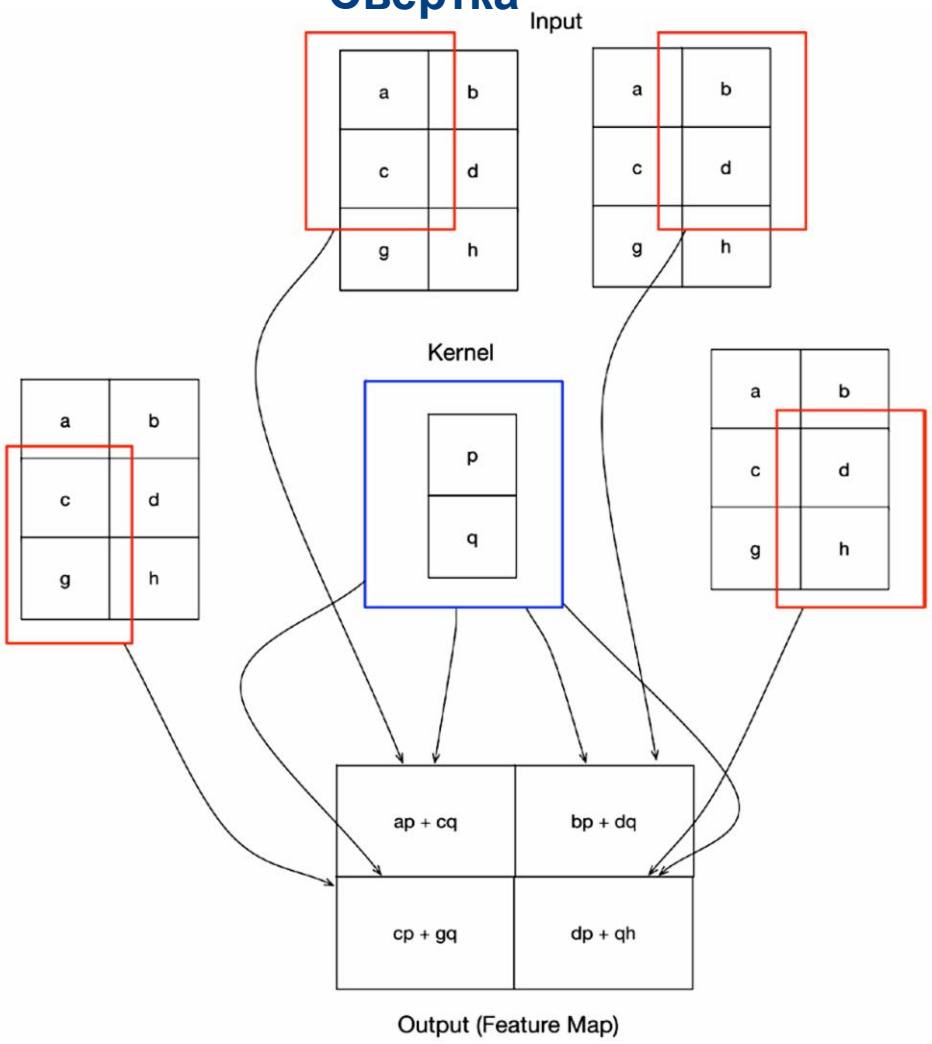


Goodfellow et al 2014,
Goodfellow I., Bengio Y., Courville A.,
Deep learning

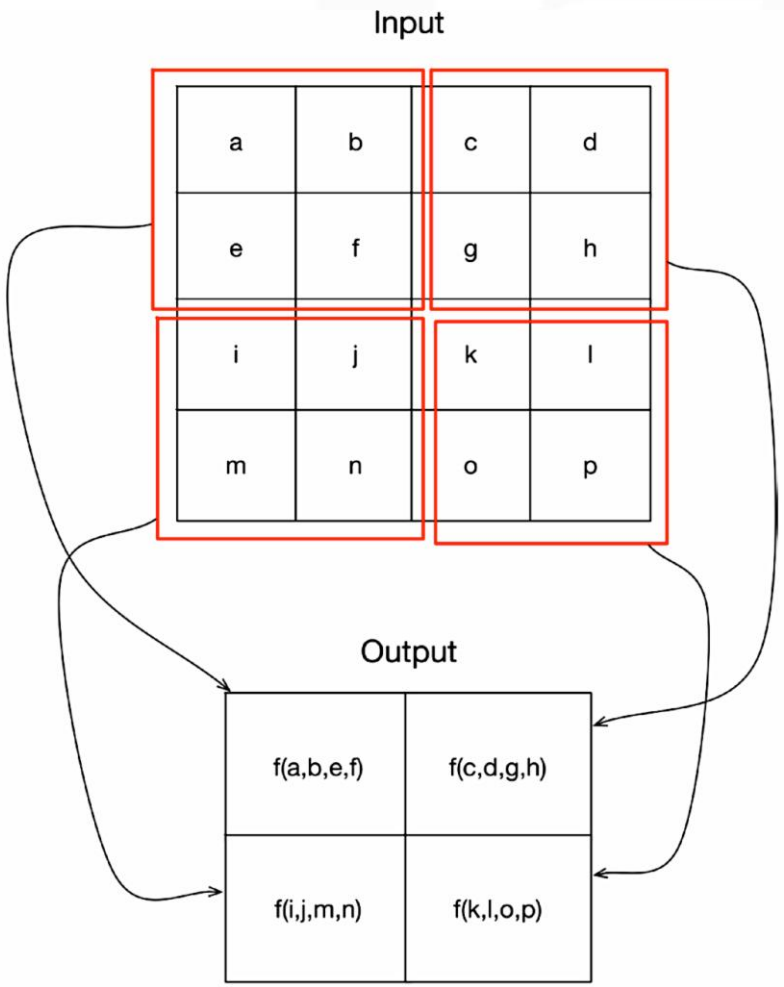
Инструмент	**Язык программирования**
Tensorflow	Python, C++
Keras	Python
PyTorch	Python
Torch	Lua, C, Python (PyTorch)
Caffe2	Python, C++
Caffe	C++, Python, Matlab
CNTK	Python, C++, C#, Java
Deeplearning4j	Java, Python
Misc (deeplearning.net/software_links/)	

Сверточные нейронные сети

Свертка

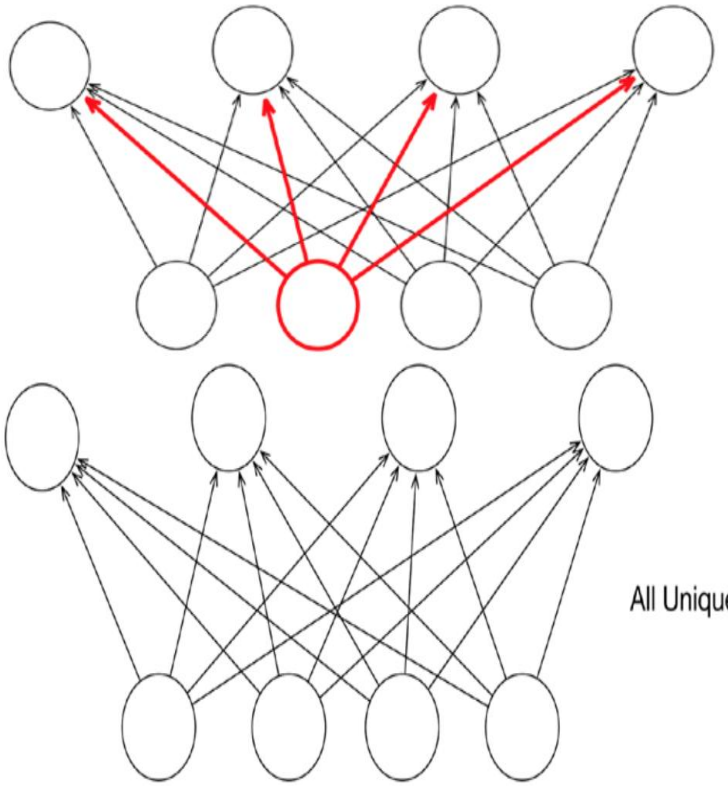


Pooling (max, avg,…)

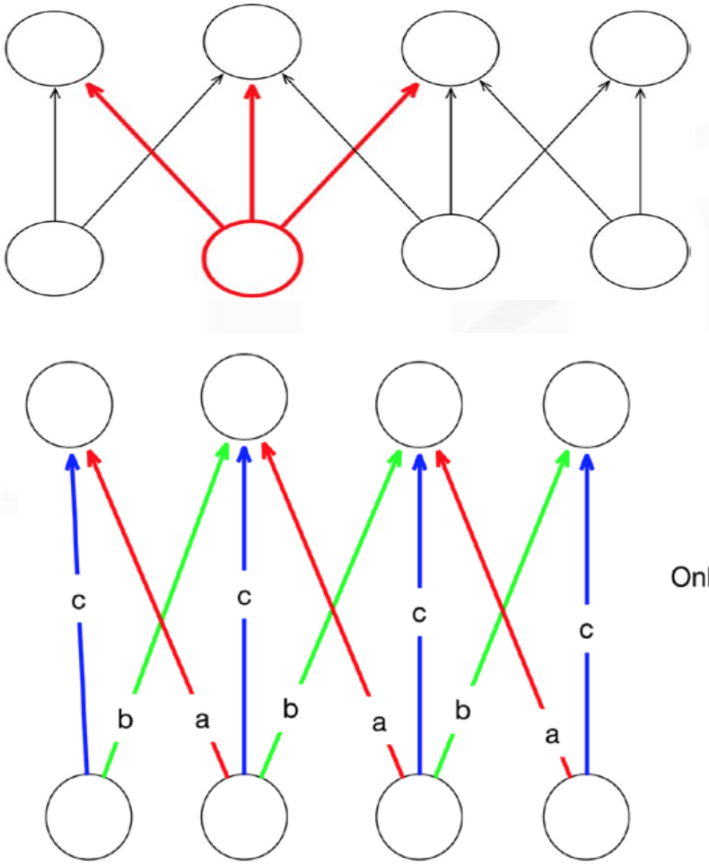


Ketkar “Deep Learning with Python”

Полносвязный слой

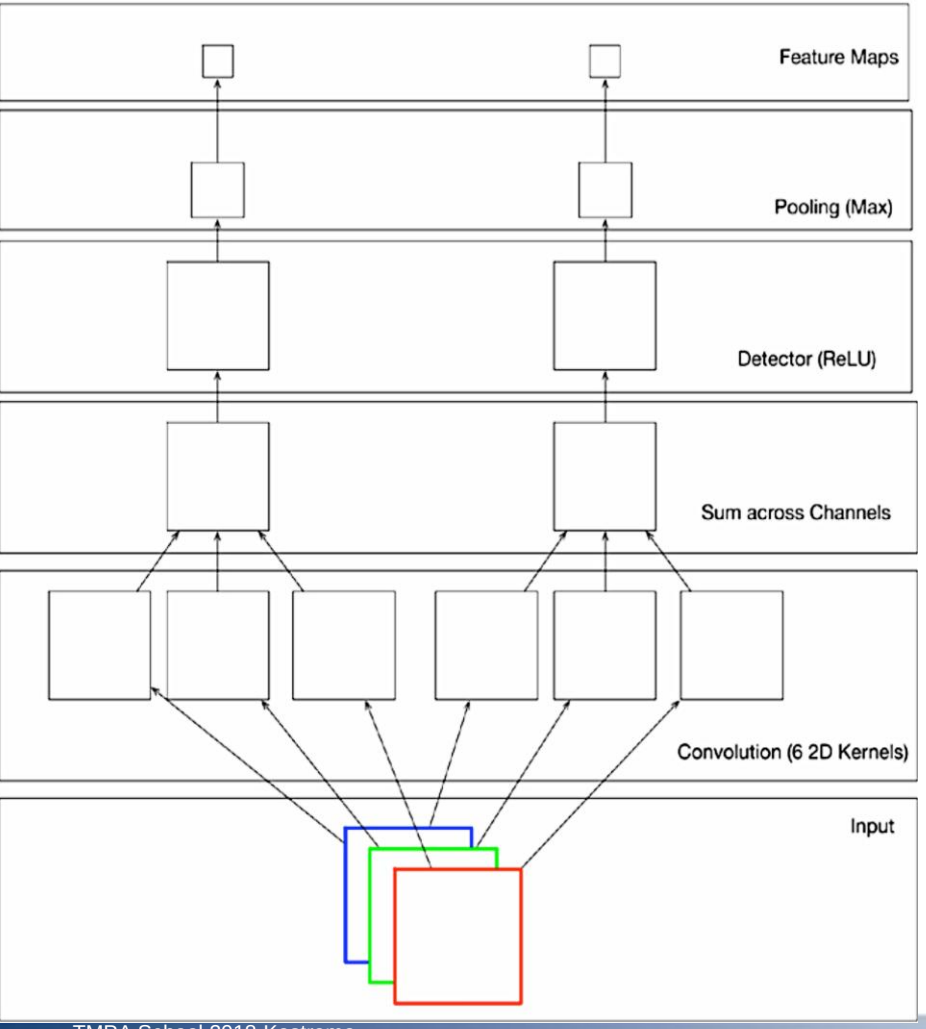


Сверточный слой

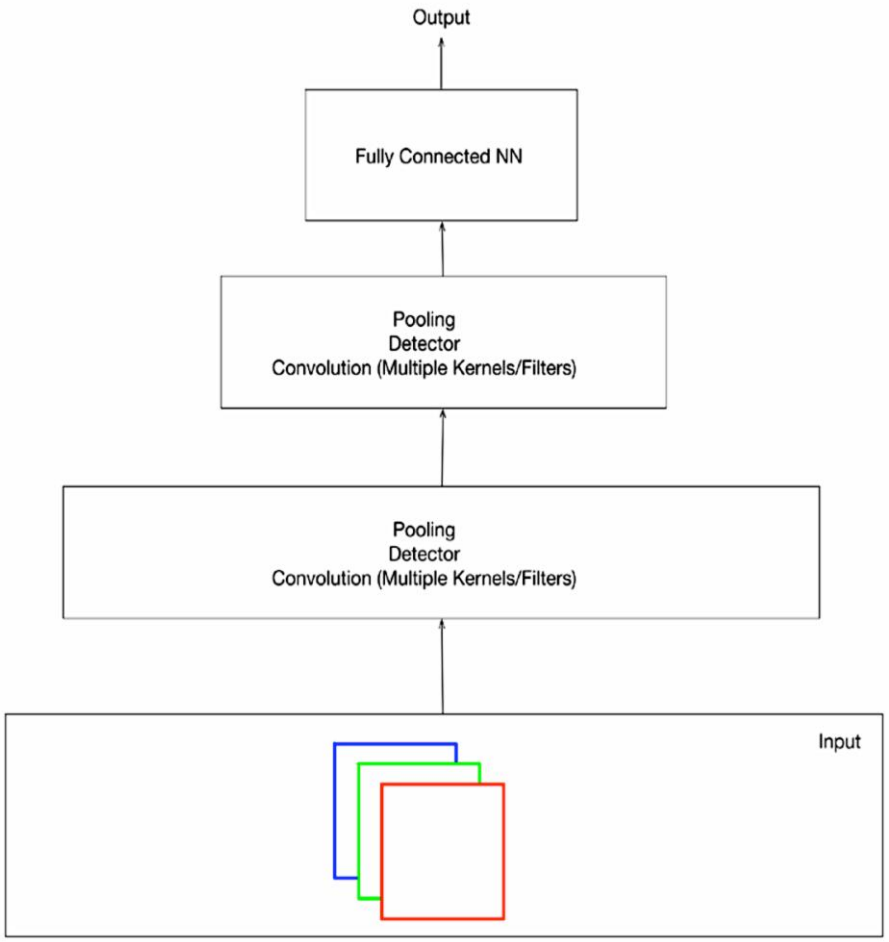


Ketkar “Deep Learning with Python”

Convolution-detector-pooling block



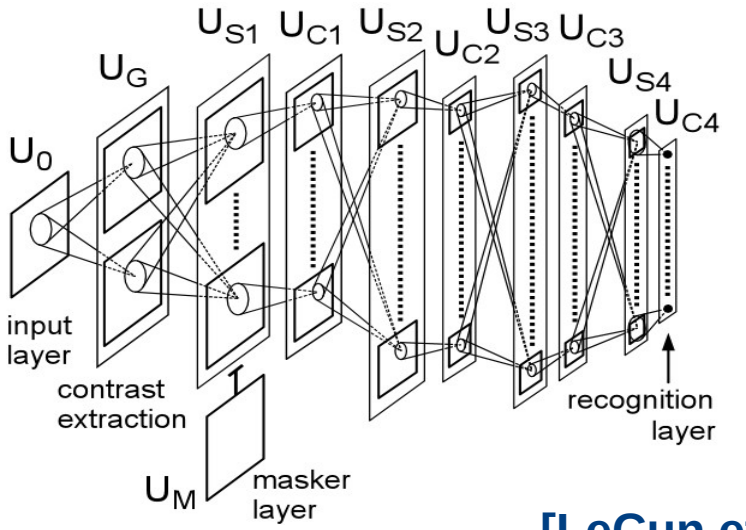
Complete ConvNet

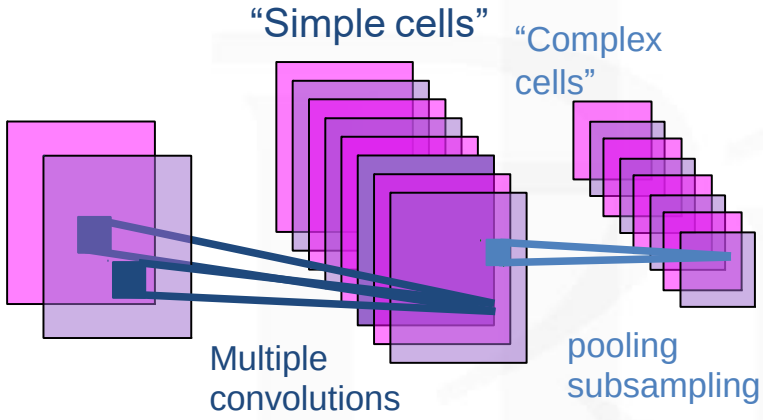


Ketkar “Deep Learning with Python”

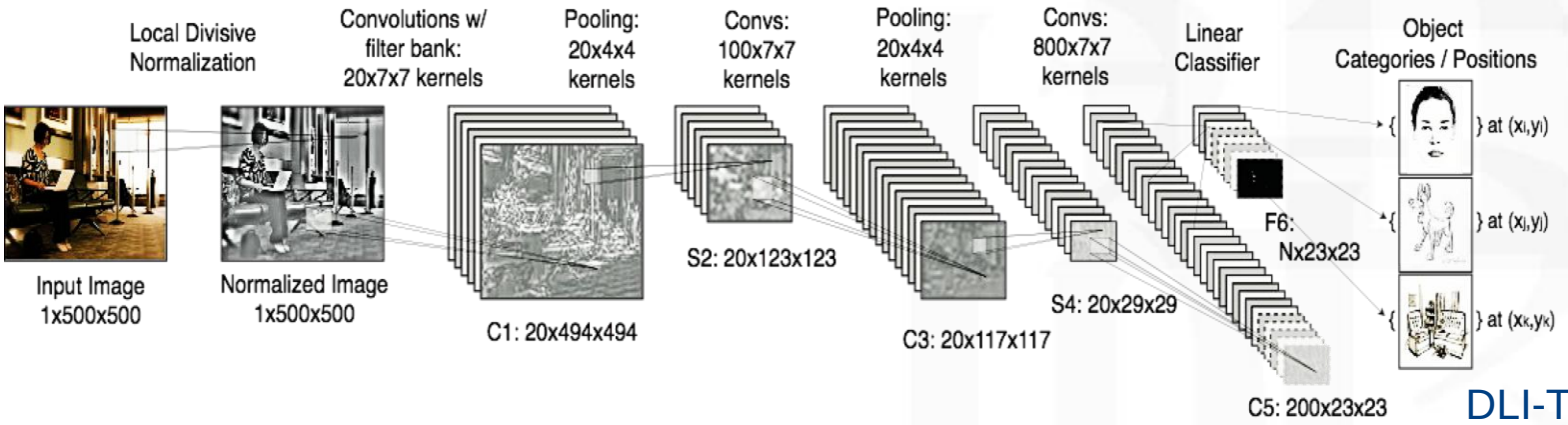
Современные архитектуры

[Hubel & Wiesel 1962], Cognitron & Neocognitron [Fukushima 1974-1982]





[LeCun et al. 89], [LeCun et al. 98]



DLI-Teaching-Kit

А потом появились:

- **ImageNet [Fei-Fei et al. 2012]**
  - **1.5 миллиона изображений**
  - **1000 категорий (классов)**
- **Быстрые NVIDIA Graphical Processing Units (GPU)**
  - **1 триллион операций/сек.**

Matchstick

Flute

Backpack

Sea lion

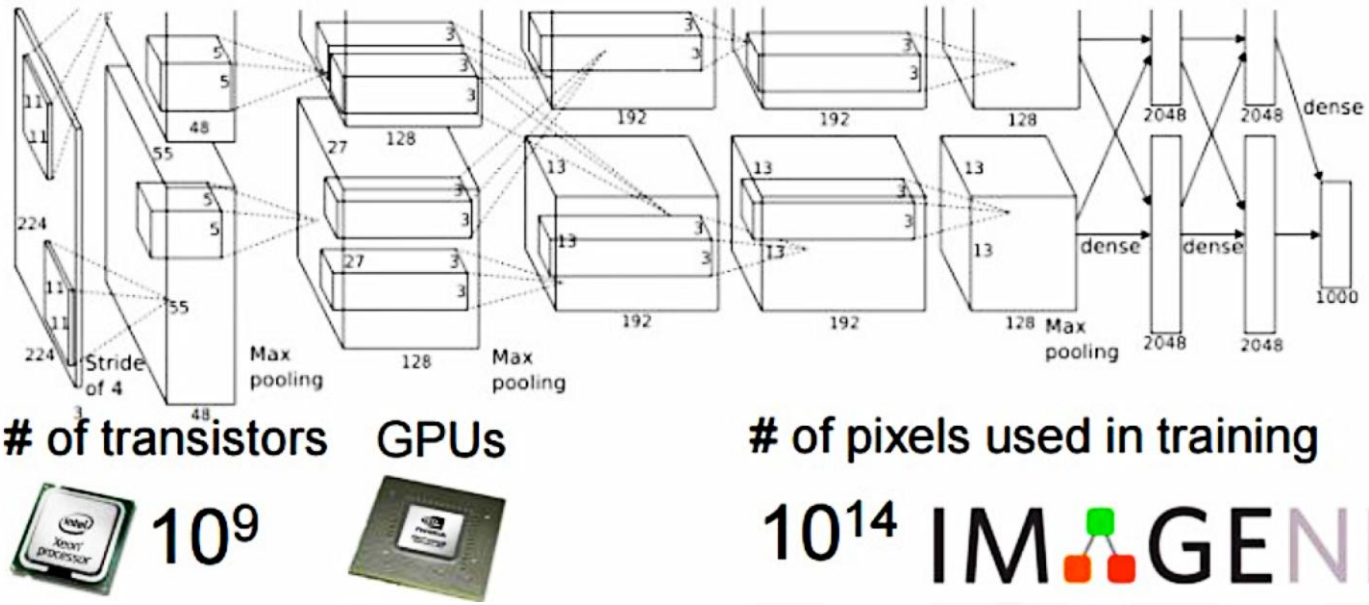
Strawberry

Bathing cap

Racket

DLI-Teaching-Kit

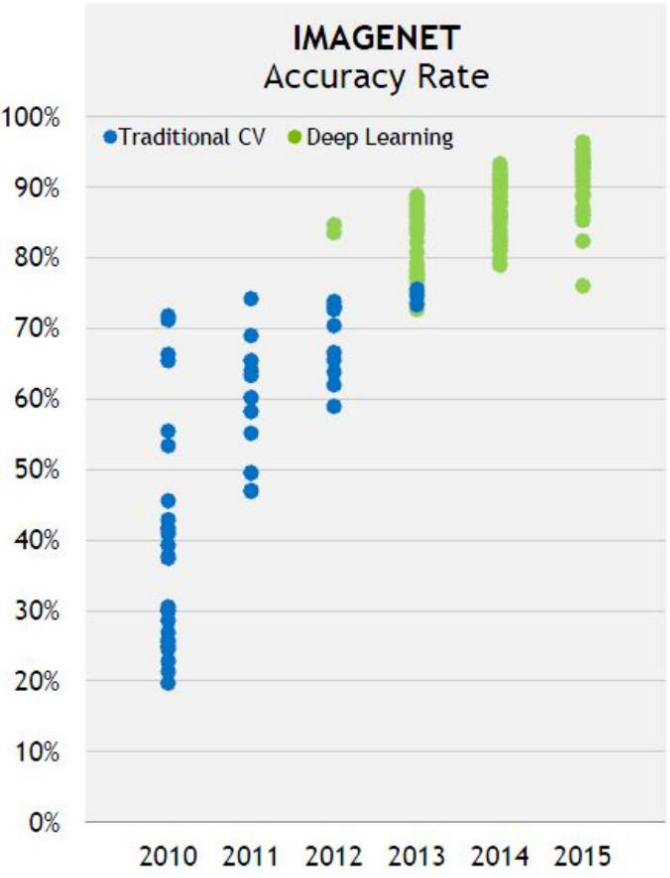
[Krizhevsky, Sutskever, Hinton 2012]

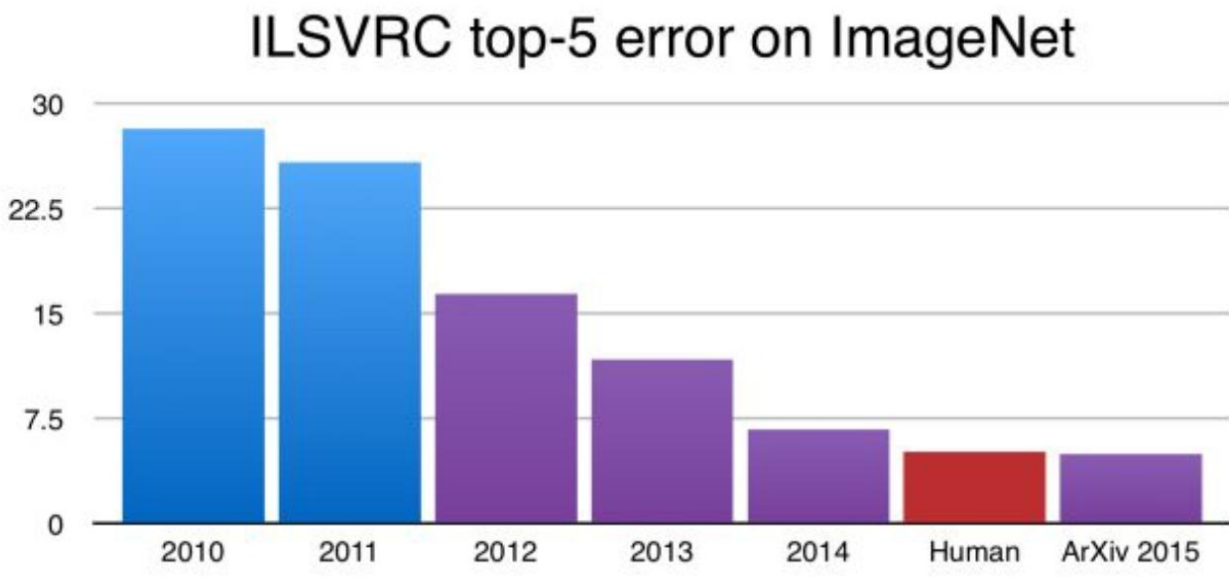


- **Глубокая нейронная сеть, обученная с помощью обратного распространения ошибки на NVIDIA GPU «with all the tricks Yann came up with in the last 20 years, plus dropout” (Hinton, NIPS 2012)»**
- **Top-5 Error rate: 15%. Предыдущий state of the art: 25%**

- **Куплен Google в январе 2013, появилась в Google+ Photo Tagging в мае 2013**

Результаты в конкурсе ImageNet

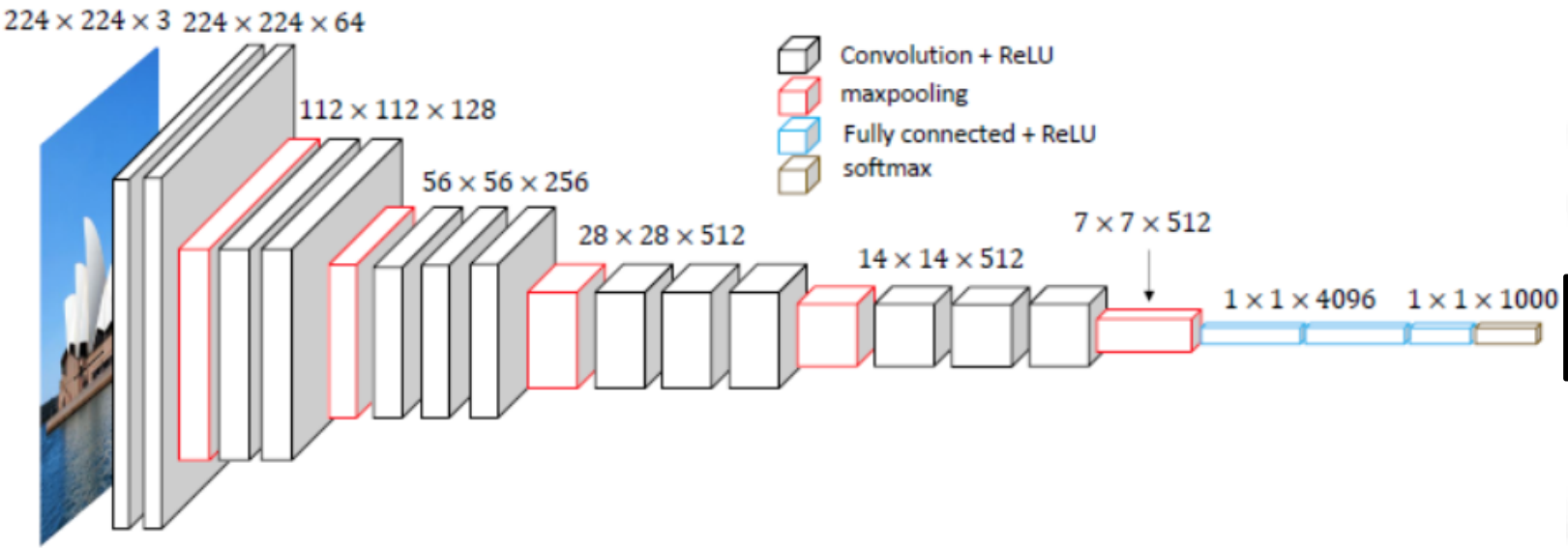




- Blue: Traditional CV
- Purple: Deep Learning
- Red: Human

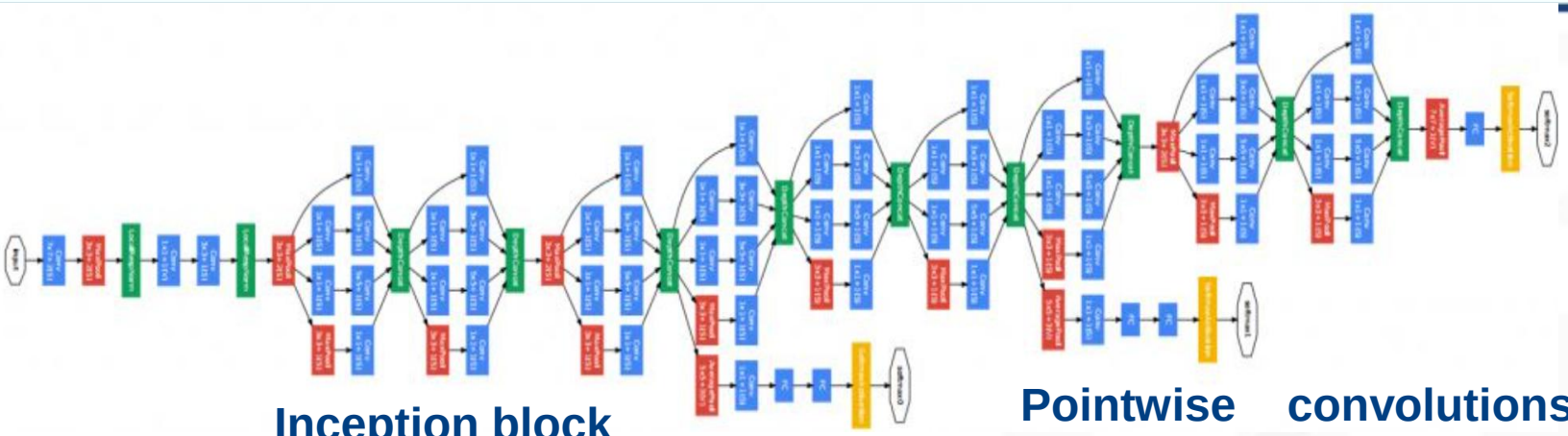
Sapunov
http://arxiv.org/abs/1502.01852

VGGNet (Oxford visual Geometry Group)

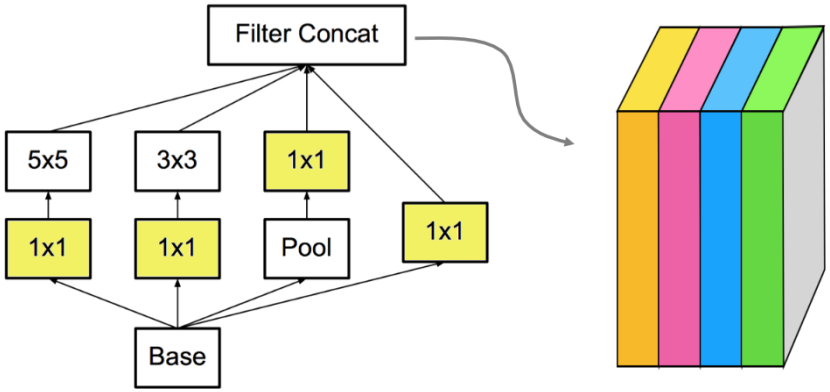


- **Только свертки 3x3, но много фильтров**
- **Вероятность ошибки (ImageNet Top-5): 8%**
- **138M параметров**

image
Conv-64
Conv-64
maxpool
Conv-128
Conv-128
maxpool
Conv-256
Conv-256
maxpool
Conv-512
Conv-512
Conv-512
maxpool
Conv-512
Conv-512
Conv-512
maxpool
fc-4096
fc-4096
fc-1000
Softmax

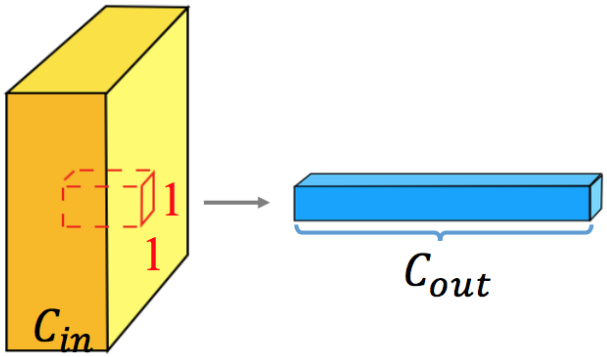


Inception block



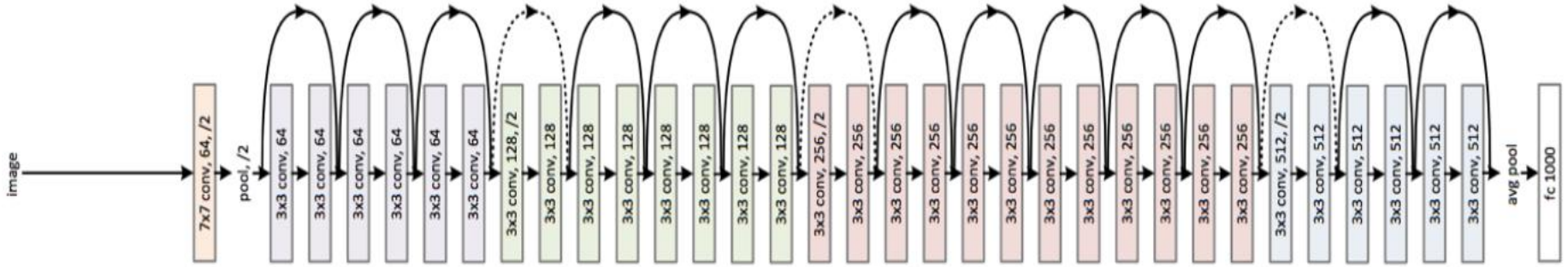
Christian Szegedy, https://arxiv.org/pdf/1512.00567.pdf

Pointwise convolutions: 1x1 convolution to reduce the number of channels



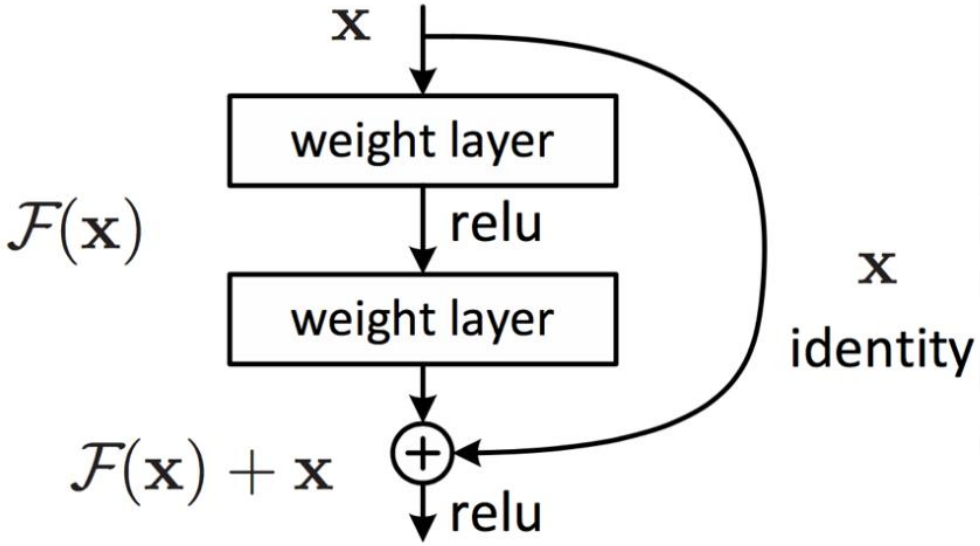
- **Вероятность ошибки v3 (ImageNet Top-5): 5.6% (одна модель), 3.6% (ансамбль)**
- **25M параметров**

Residual Network (ResNet)

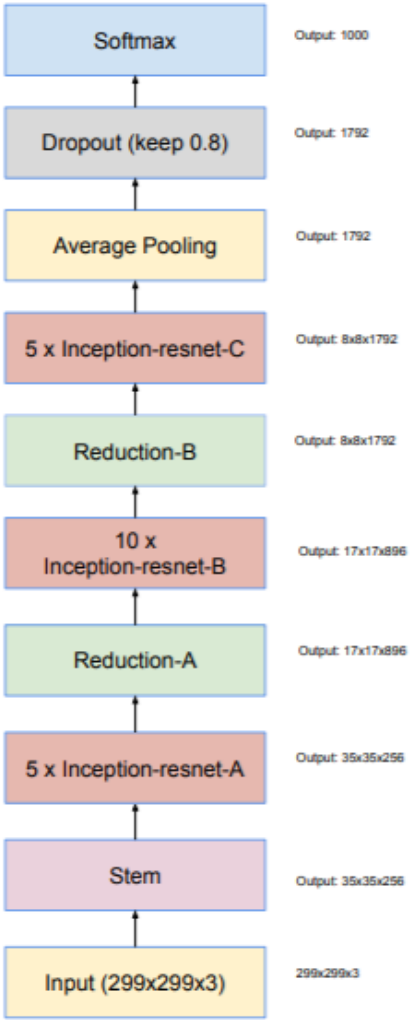


- **50/101/152 слоев, несколько сверток 7x7, много сверток 3x3, batch norm, max/average pooling**
- **Вероятность ошибки (ImageNet Top-5): 4.5% (одна модель), 3.5% (ансамбль)**
- **60M параметров у ResNet-152**

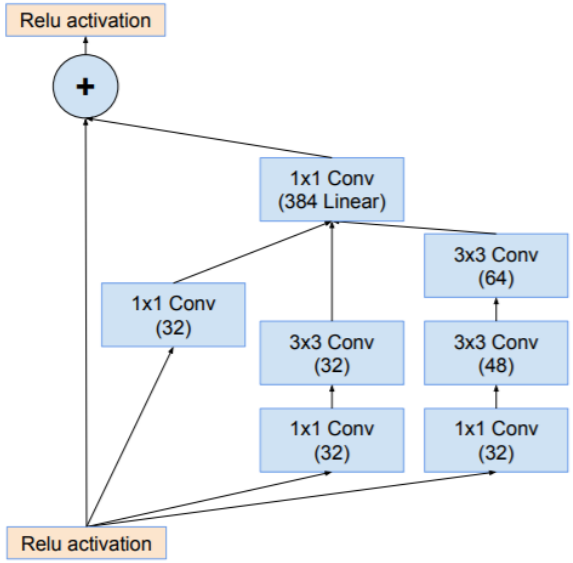
Residual block



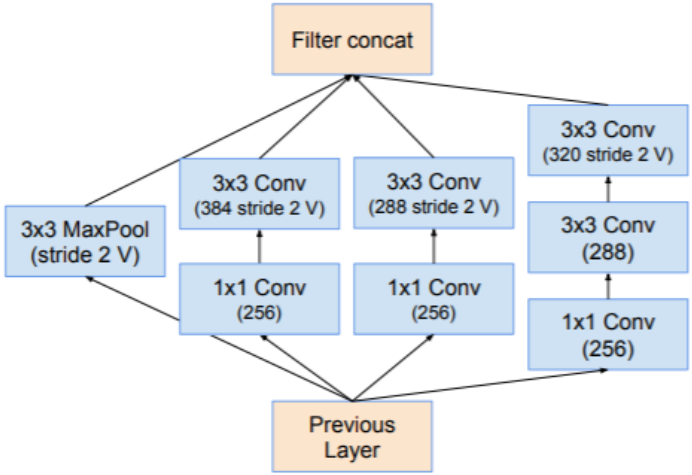
Inception-ResNet v2



Inception-ResNet block



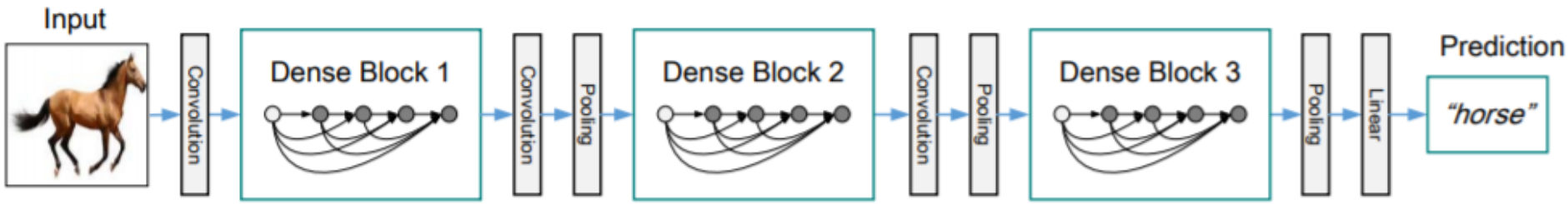
Grid-reduction block



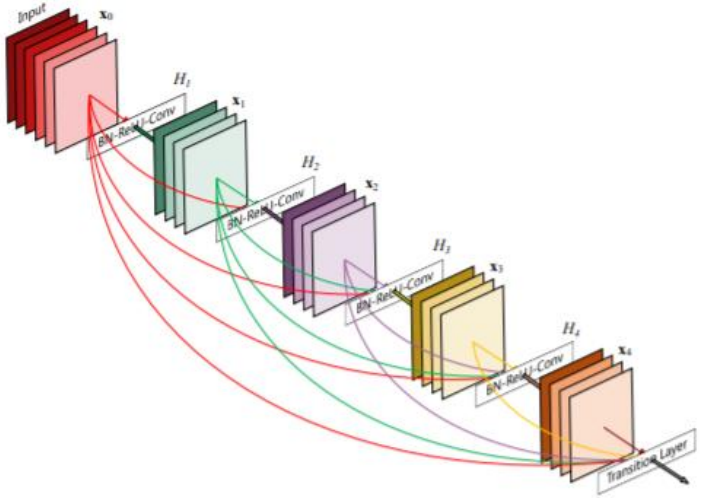
- **Более 550 слоев**
- **Вероятность ошибки (ImageNet Top-5): 4.5% (одна модель), 3.7% (ансамбль)**
- **55M параметров**

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

DenseNet



Dense convolution block

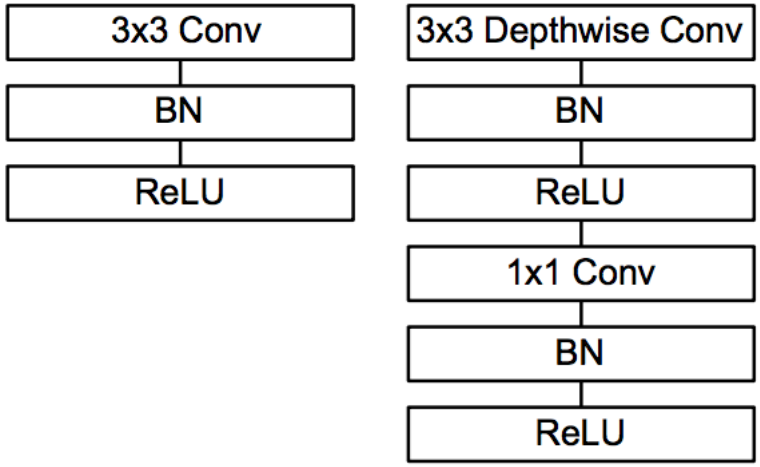


Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48	[1 × 1 conv, 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

https://arxiv.org/pdf/1608.06993.pdf

- **Вероятность ошибки (ImageNet Top-5): 6.7% (одна модель), 5.3% (ансамбль)**
- **80M параметров (DenseNet-200)**

Depthwise convolution

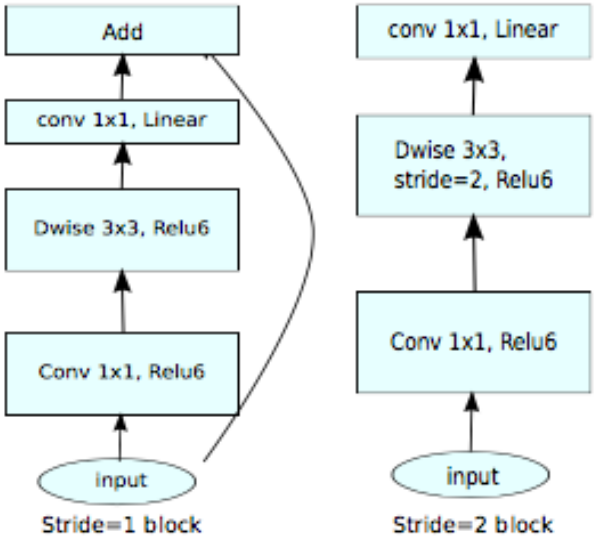


- **28 слоев**
- **Вероятность ошибки (ImageNet Top-5): 12.81%**
- **4.2M параметров**

https://arxiv.org/abs/1704.04861

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Bottleneck residual block



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

Каждая строка в таблице – описание слоя, который повторяется n раз.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$28^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times k$	conv2d 1x1	-	k	-	

- **ImageNet точность (Top-1): 74.7% (ср. с 70.6% для MobileNet v1)**
- **6..9M параметров**

http://arxiv.org/abs/1801.04381

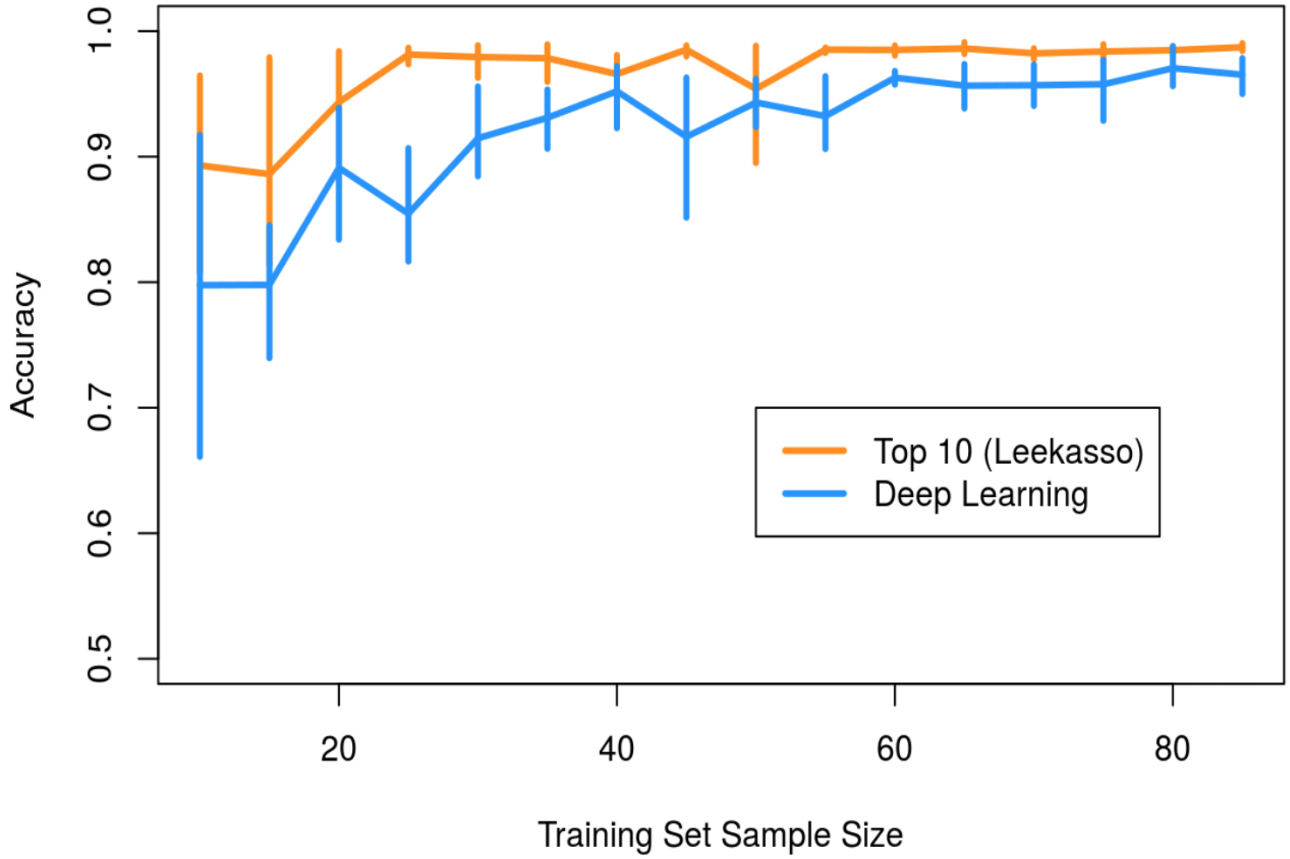
Как выбрать архитектуру сети?

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

https://keras.io/applications/

Проблемы ConvNet

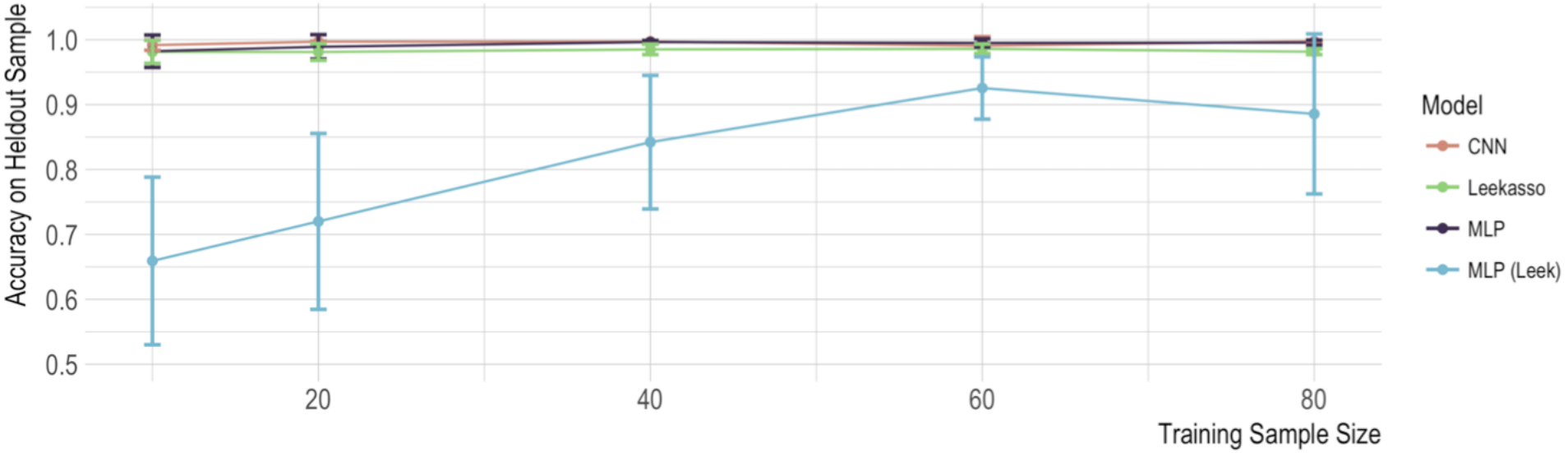
Пример: Распознавание цифр «0» и «1» (MNIST)



Bias/variance tradeoff

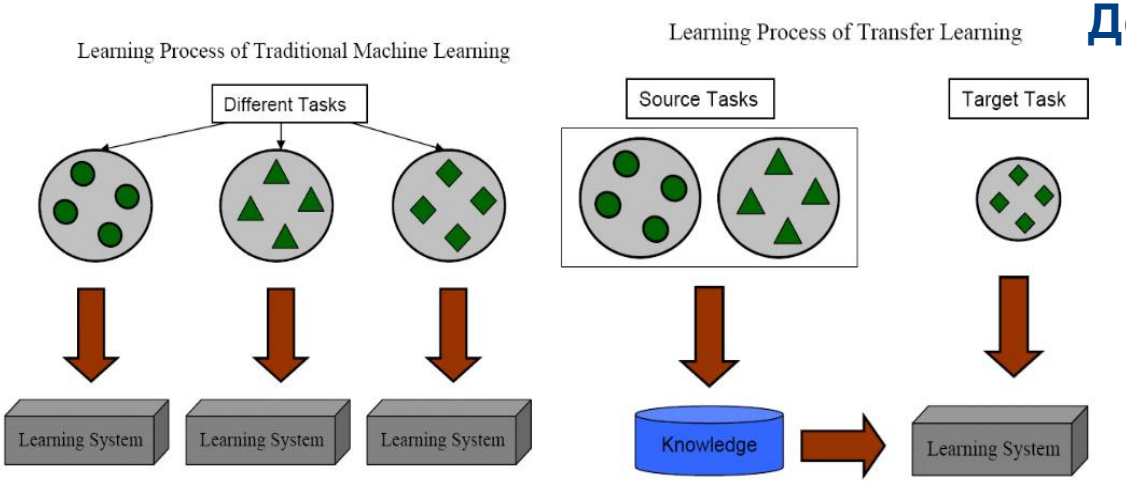
Leek “Don’t use deep learning your data isn’t that big”

Обучать сеть нужно правильно

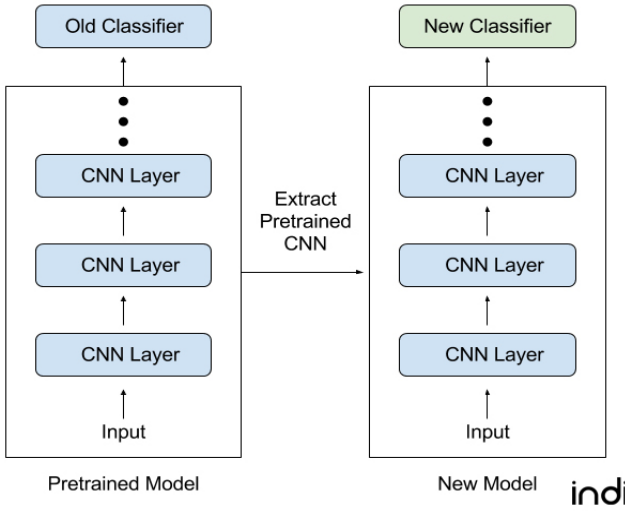


1. **tanh->relu**
2. **SGD должен сойтись (20-50 эпох недостаточно)**
3. **Использовать dropout**
4. **Важен подбор параметров (но параметры по умолчанию в Keras – ok)**

http://beamandrew.github.io/deeplearning/2017/06/04/deep_learning_works.html



Дообучение сети (fine-tuning)



OverFeat features → Trained classifier on other data sets [Razavian, Azizpour, Sullivan, Carlsson "CNN features off-the-shelf: An astounding baseline for recogniton", CVPR 2014], http://www.csc.kth.se/cvap/cvg/DL/ots/

• image classification	ImageNet LSVRC 2013	competitive	13.6 % error
	Dogs vs Cats Kaggle challenge 2014	state of the art	98.9%
• object localization	ImageNet LSVRC 2013	state of the art	29.9% error
• object detection	ImageNet LSVRC 2013	state of the art	24.3% mAP

[Han et al. 2016] Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding – **ICLR16 Best paper**

- **Pruning**
  [Han et al. 2016], [Molchanov et al. 2016]
- **Distillation The Knowledge (FitNet)**
  [Hinton et al. 2014], [Romero et al. 2014]
- **Weights Hashing / Quantization**
  [Chen et al. 2015], [Han et al. 2016]
- **Tensor Decompositions**
  [Lebedev et al. 2015], [Kim et al. 2015], [Novikov et al. 2015], [Garipov et al. 2016]
- **Binarization**
  [Courbariaux / Hubara et al. 2016], [Rastegari et al. 2016], [Merolla et al. 2016], [Hou et al. 2016]
- **Architectural tricks**
  [Hong et al. 2016], [Iandola et al. 2016], [Teerapittayanon et al. 2016]

[Rassadin, Savchenko, 2017]

Сжатие сверточных сетей (2).
Распознавание эмоций (Radboud Faces Database, RaFD)

	Training time per epoch, ms	Inference time, ms	Model size, MB	Accuracy, %
VGG-S (baseline)	43.7	33.4	372.2	97.13
SqueezeNet-1.1 (baseline)	**22.94**	**4.94**	2.8	89.14
SqueezeNet-1.1, CP-Decomposition	**22.94**	7.74	**2.1**	87.5
HashedNets	294.8	158.2	68.3	96.31
Binary-Weight-Network (BWN)	83.8	33.5	11.6	**98.57**
XNOR-Net	84.3	34.2	11.6	58.81
XNOR-Net w/o weights activation	43.4	34.1	11.6	88.32

[Rassadin, Savchenko, 2017]

Уязвимости сверточных сетей (1). Adversarial examples

x

"panda"
57.7% confidence

$+ .007 \times$

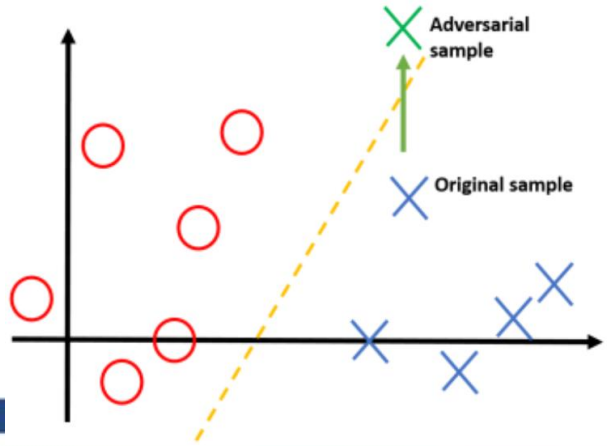
$\text{sign}(\nabla_x J(\theta, x, y))$

"nematode"
8.2% confidence

$=$

$x + \epsilon\text{sign}(\nabla_x J(\theta, x, y))$

"gibbon"
99.3 % confidence



Goodfellow I., Bengio Y., Courville A., Deep learning, [Szegedy et al., 2014]

ImageNet

Attack Method	VGG-16	AlexNet	ResNet-50	Inception v3
FGS (Fast gradient sign)	0.85/0.56/0.61	0.56/0.22/0.33	0.88/0.53/0.62	**0.92**/0.01/0.07
Black-Box	0.85/0.61	0.56/0.43	0.88/0.69	**0.92**/0.19

CIFAR-10

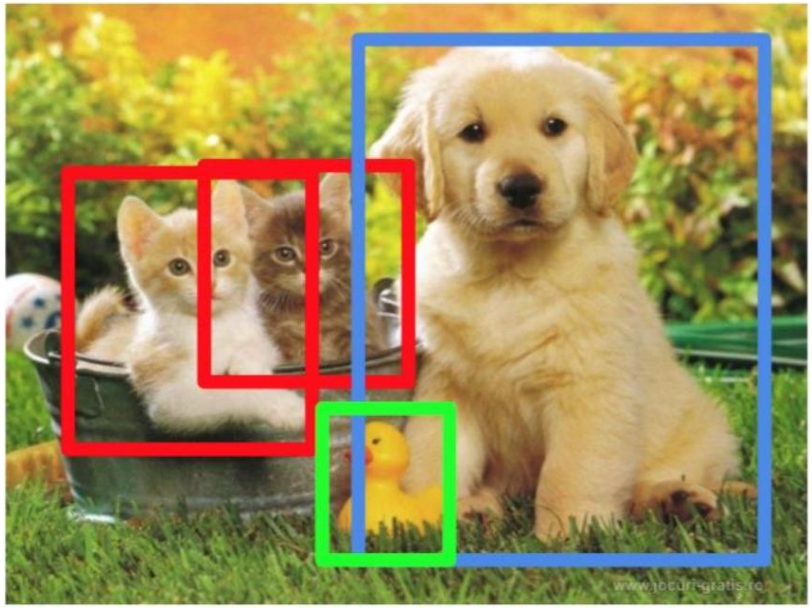
Attack Method	VGG-16	ResNet-50
FGS	0.79/0.03/0.36	**0.81**/0.05/0.43
Black-Box	0.79/0.44	**0.81**/0.51

CIFAR-100

Attack Method	VGG-16	ResNet-50
FGS	0.64/0.15/0.36	**0.67**/0.29/0.43
Black-Box	0.64/0.41	**0.67**/0.38

[Груздев, 2017]

Детектирование объектов



Кошка

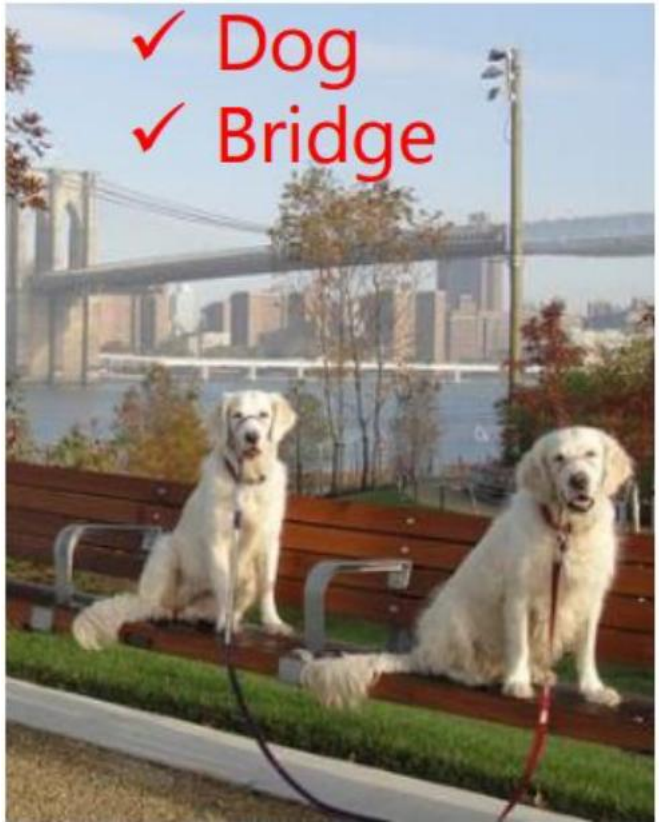
Собака

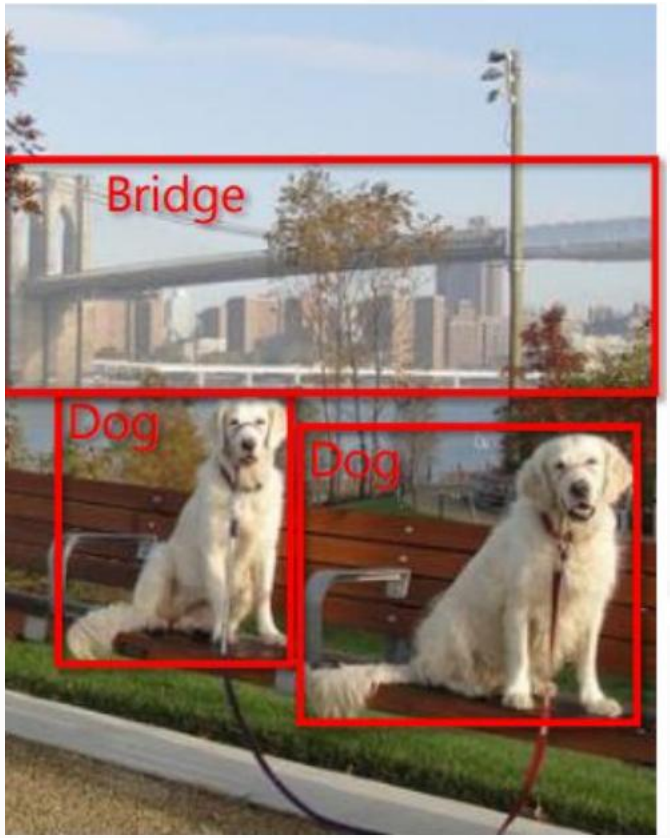
Утка

Определить положение (прямоугольник) и метку для каждого объекта (из определенного множества классов) на изображении

https://telecombcn-dl.github.io/2017-dlcv/slides/D3L4-objects.pdf

Распознавание (классификация) и детектирование





http://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-detection.pdf

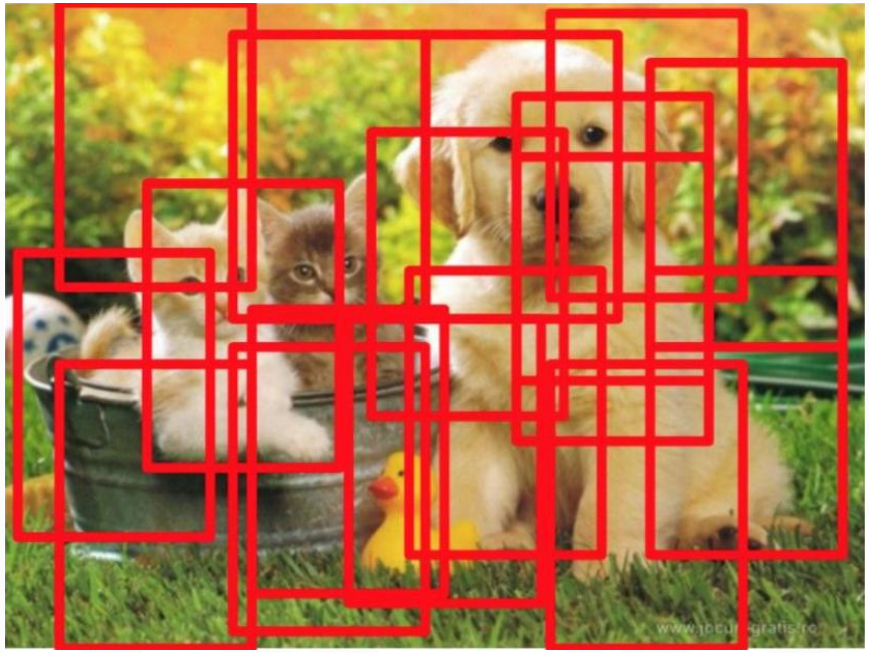
Детектирование объектов как задача классификации

Классы = [Кошка, Собака, Утка]

Кошка (окно(0,0,w,h))? **Нет**

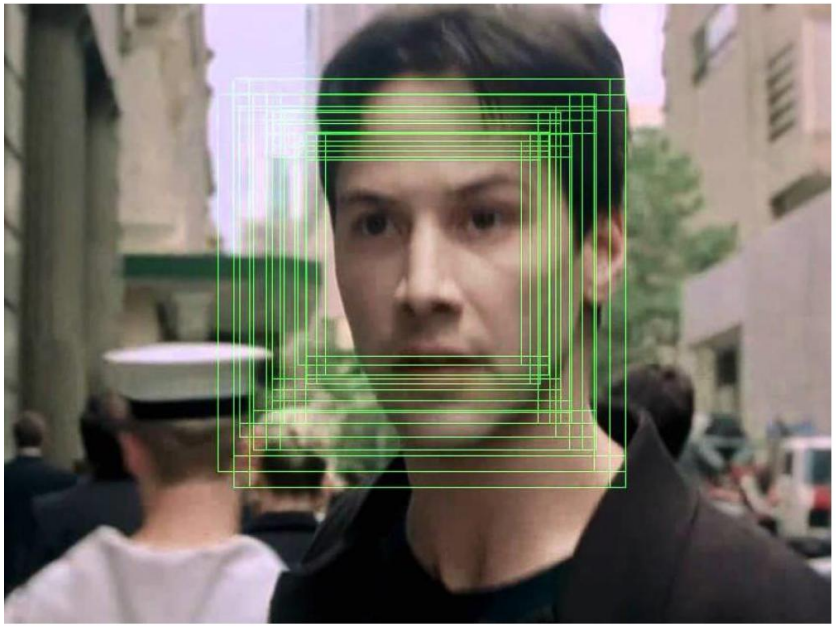
Собака (окно(0,0,w,h))? **Нет**

Утка(окно(0,0,w,h))? **Нет**

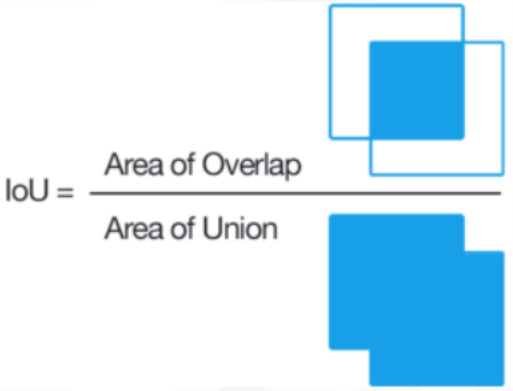


Перебираем разные положения и размеры окон

Классификатор должен быть очень быстрым!



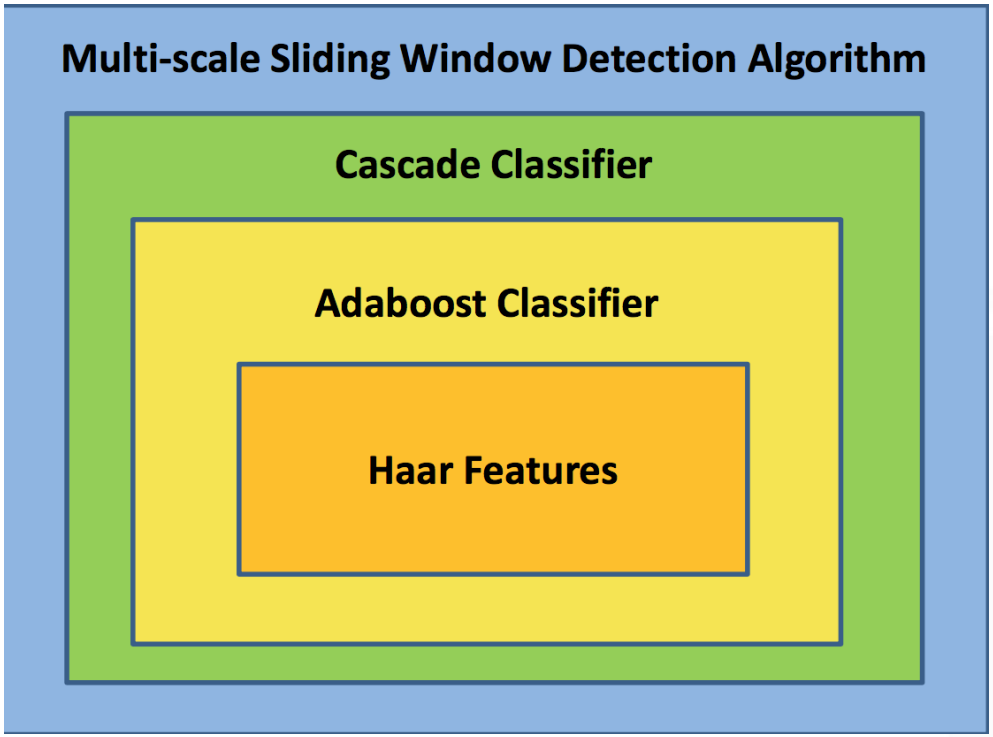
Коэффициент сходства Жаккара (Jaccard)



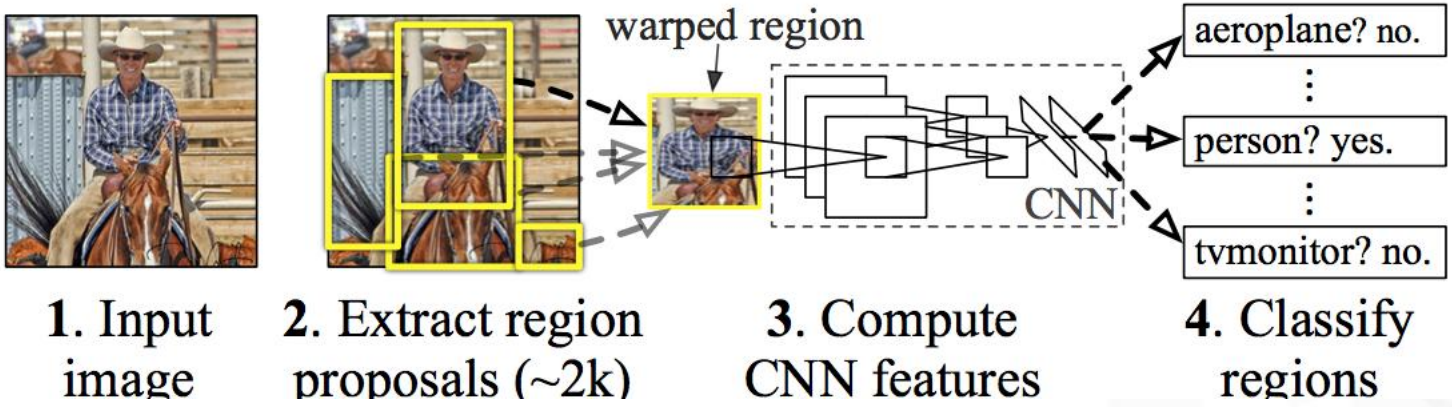
- **A и B считаются близкими если JD(A, B) ≤ eps (eps ≈ 0.6..0.9)**
- **Можно построить классы эквивалентности (связные компоненты в графе близких кандидатов). Из каждого класса сформировать один прямоугольник (median, avg, ...)**
- **Можно использовать non-maximum suppression: рассмотреть все пары близких прямоугольников (A,B), выбрать прямоугольник с большим значением confidence**

Один из первых детекторов лиц в режиме реального времени – 15 FPS на Pentium III

P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

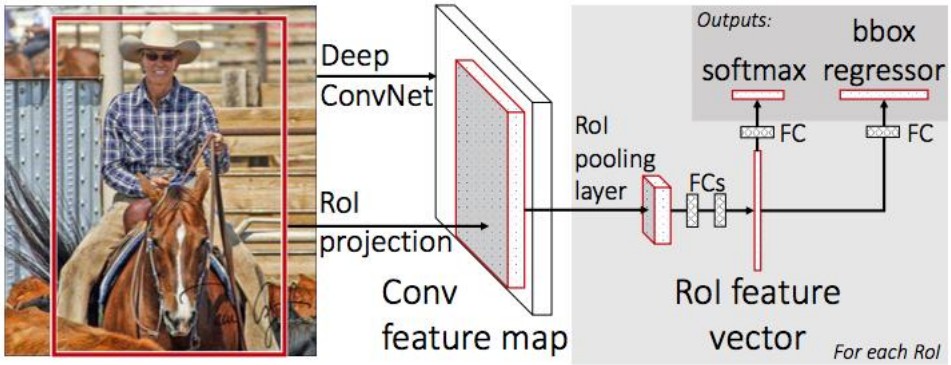


R-CNN

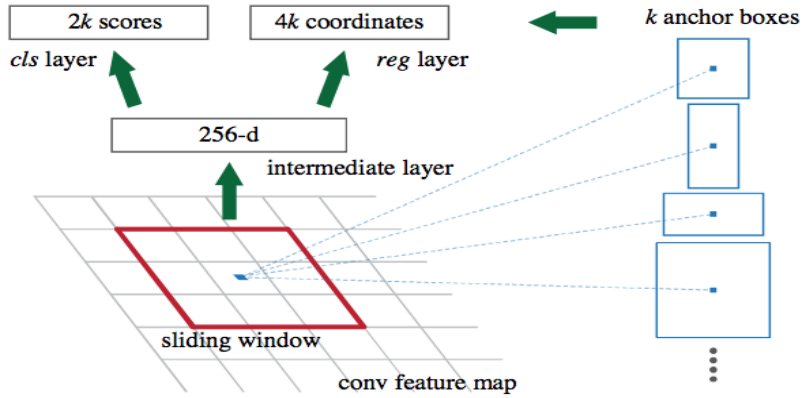


3 – признаки из AlexNet
4 – SVM
В конце – non-maximum suppression (для каждого класса)

Fast R-CNN

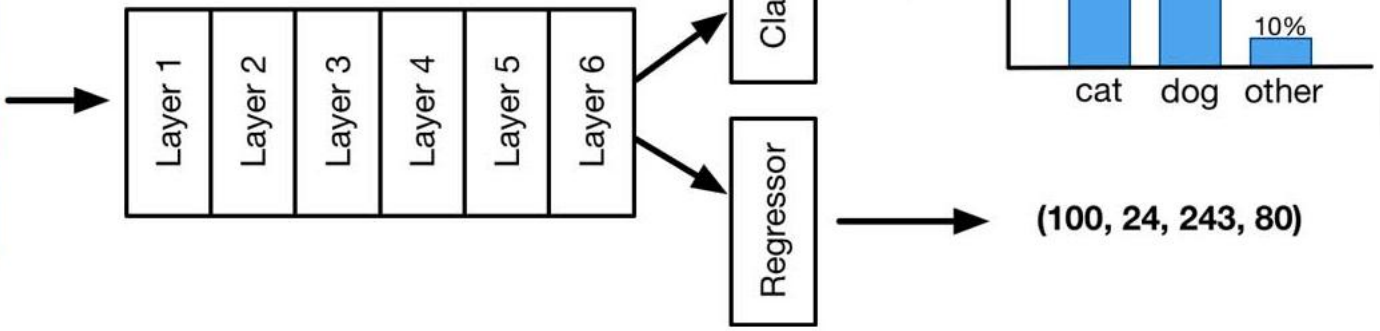


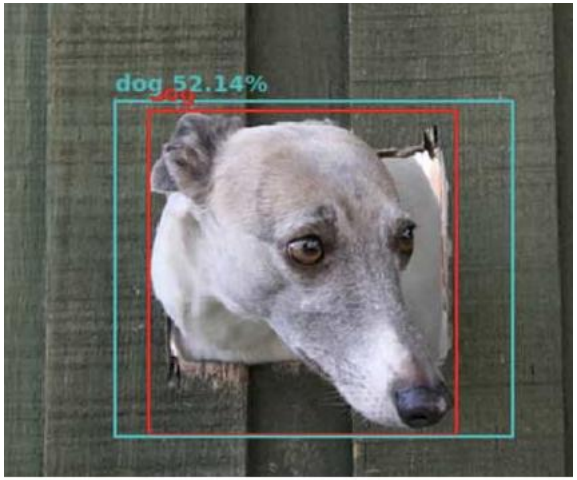
Faster R-CNN with region proposal network

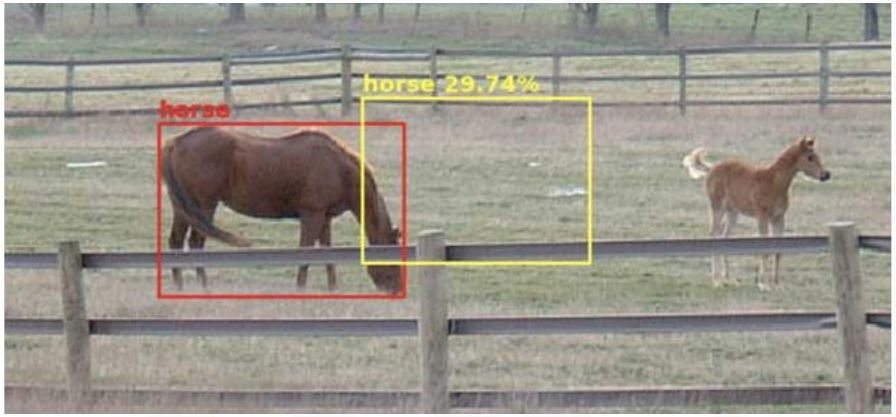


[Girshick et al, 2013], [Girshick et al, 2015], [Ren et al, 2016]

One-shot detection (1). Наивный подход

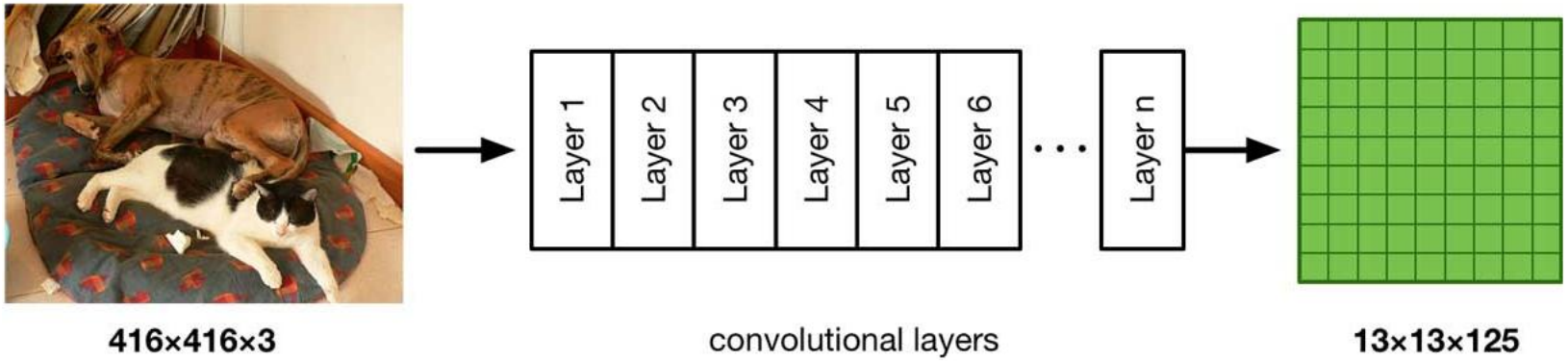


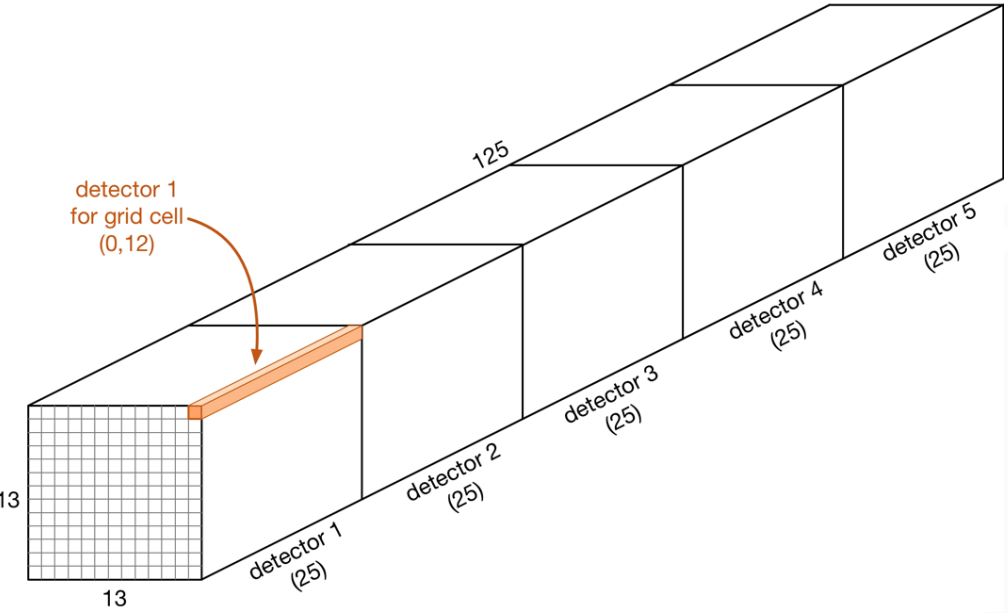




http://machinethink.net/blog/object-detection/

One-shot detection (2). Fixed grid of detectors



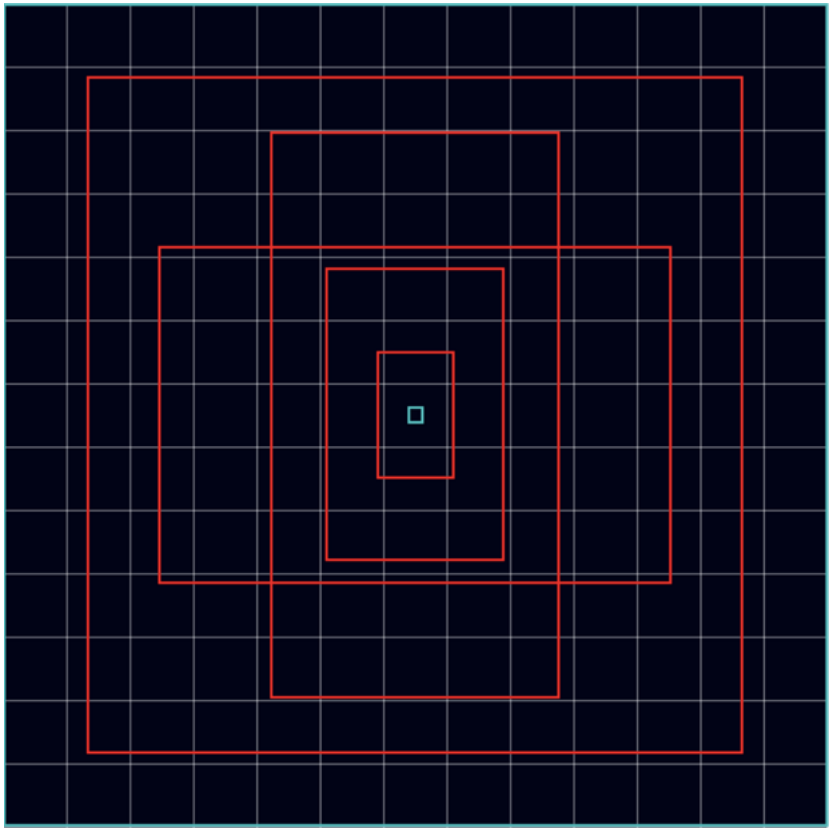


25 выходов для VOC (20 классов):

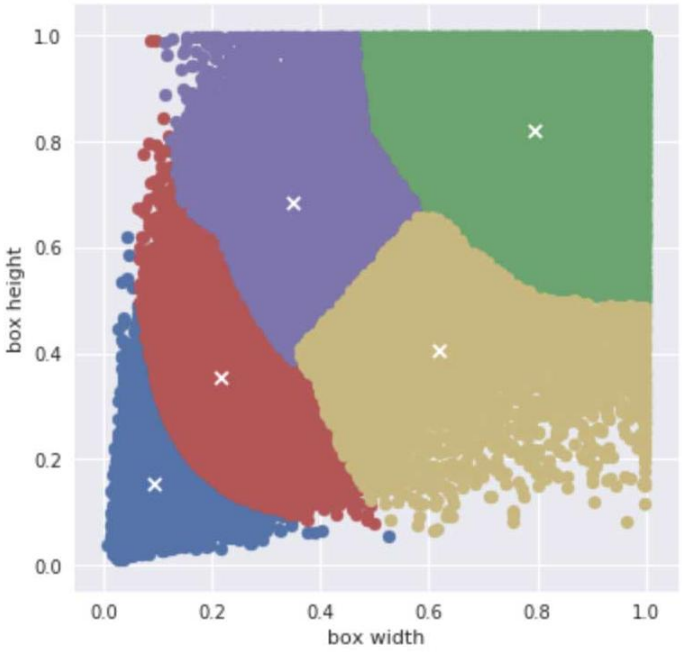
- **20 апостериорных вероятностей**
- **4 bounding box**
- **1 “object-ness” score**

http://machinethink.net/blog/object-detection/

Несколько (5) детекторов для объектов разной формы!

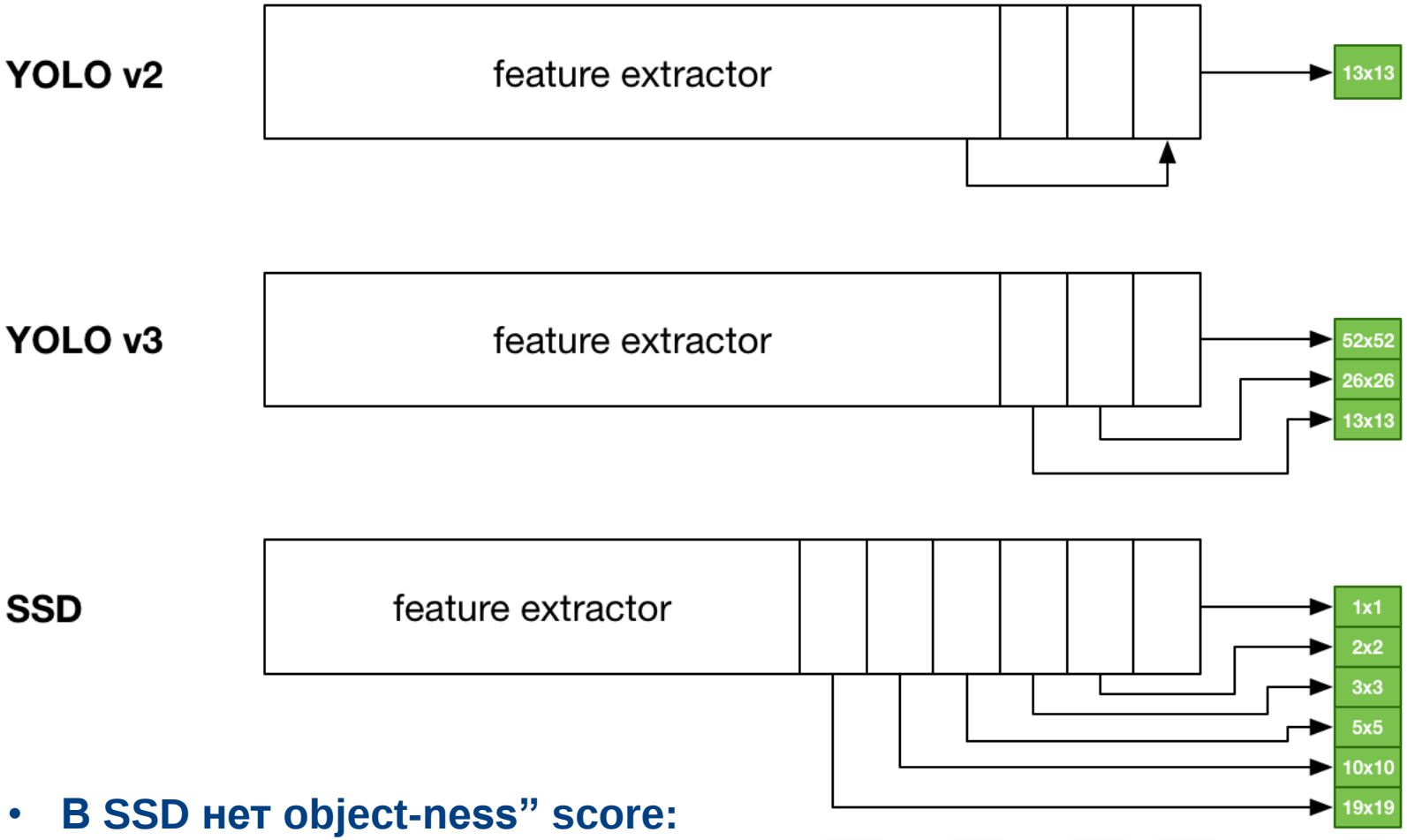


- **SSD: размеры выбираются заранее**
- **YOLO: k-means clustering (обучается на заданном наборе данных)**



http://machinethink.net/blog/object-detection/

One-shot detection (4). YOLO vs SSD



- **В SSD нет object-ness" score: добавляется класс фона**

http://machinethink.net/blog/object-detection/

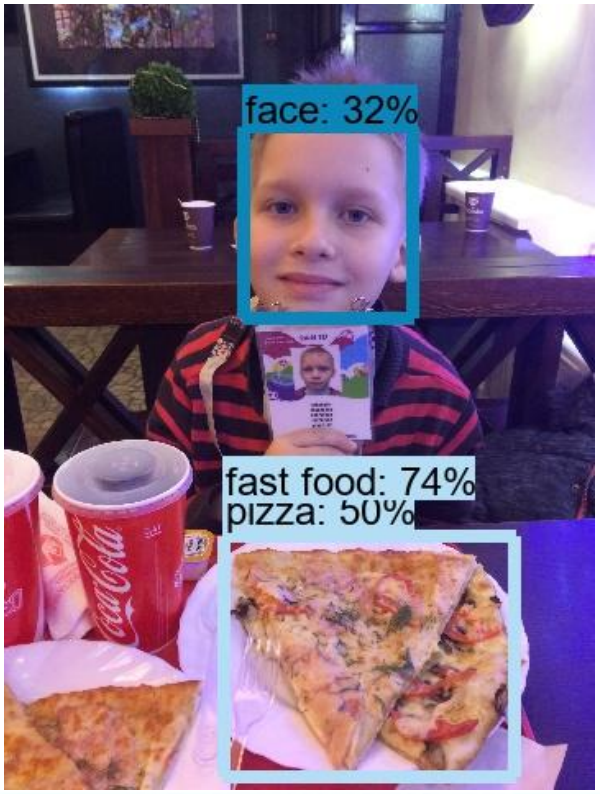
Как выбрать детектор?

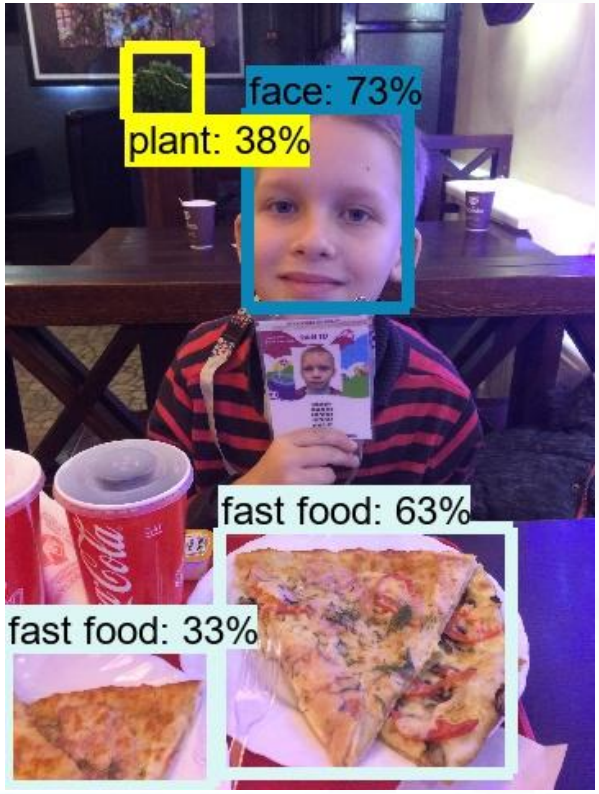
Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_nas	1833	43	Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks

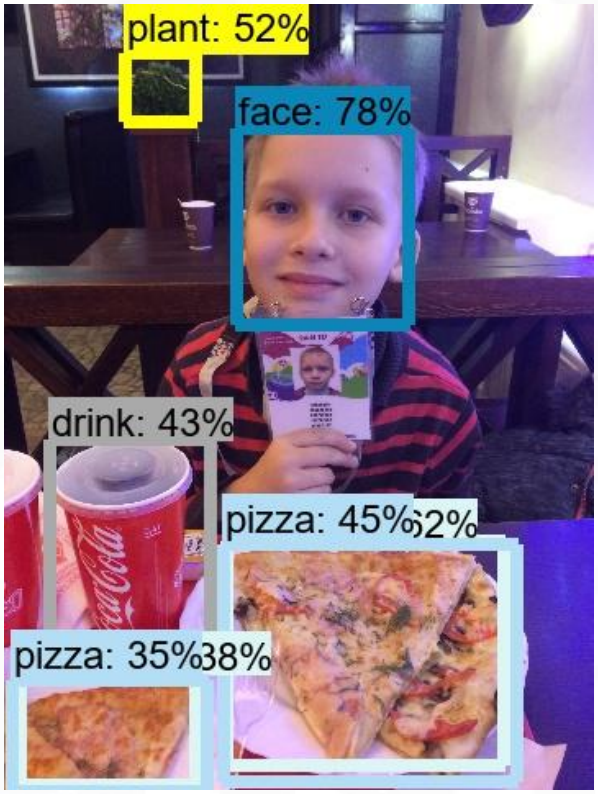
github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

Результаты (1)

	SSDLite+ MobileNet v2	Faster RCNN+ Inception v2	Faster RCNN+ InceptionResNet
Время	30-100 мс	400-500 мс	6000-7000 мс (6-7 с)
Размер	22 Мб	55 Мб	250 Мб (после квантования: 68 Мб)

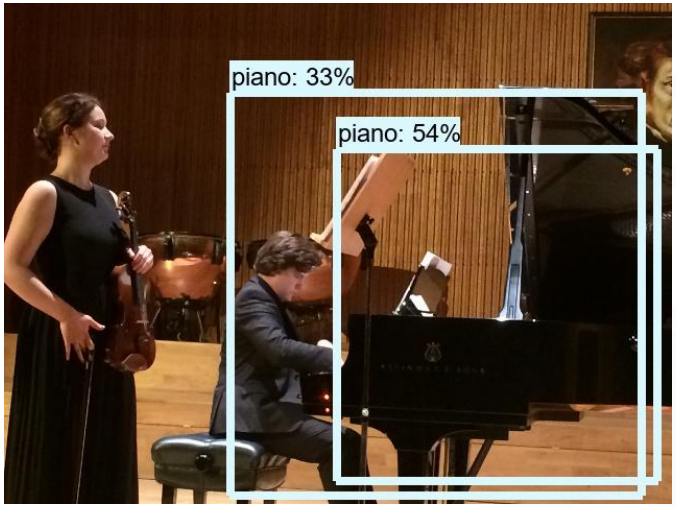




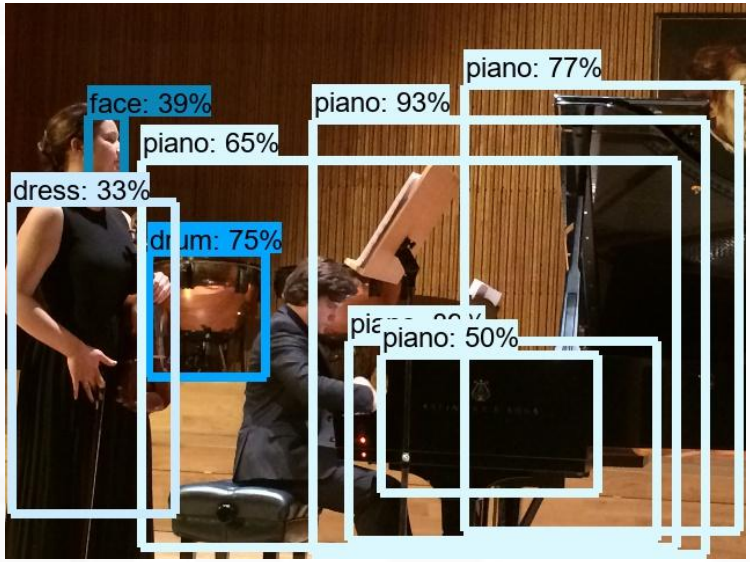


SSDLite+ MobileNet v2

Faster RCNN+ Inception v2



Faster RCNN+ InceptionResNet



Результаты (3)

SSDLite+ MobileNet v2

Faster RCNN+ Inception v2

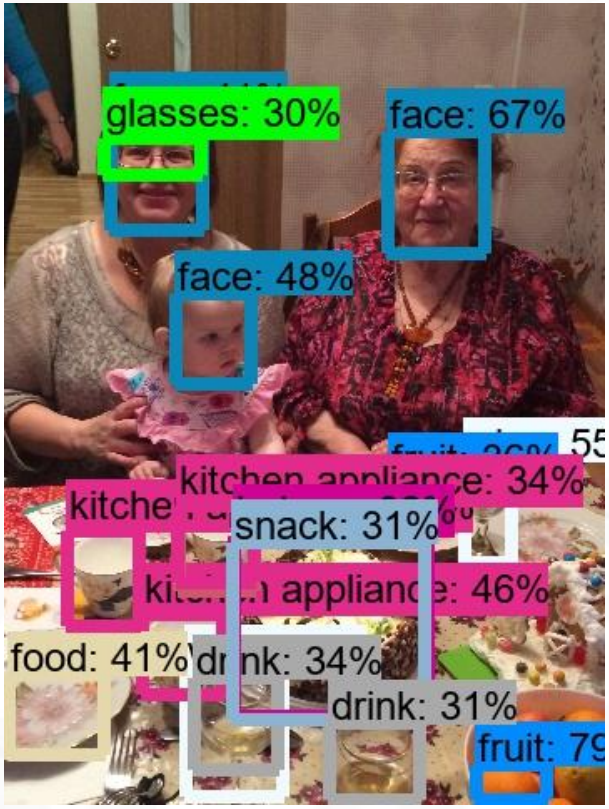
Faster RCNN+ InceptionResNet

Результаты (4)

SSDLite+ MobileNet v2

Faster RCNN+ Inception v2

Faster RCNN+ InceptionResNet



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Спасибо за внимание!