Школа Анализа Данных
Яндекса

Курс «Анализ изображений и видео»

Лекция №6
«Свёрточные нейросети 2»

Антон Конушин
Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

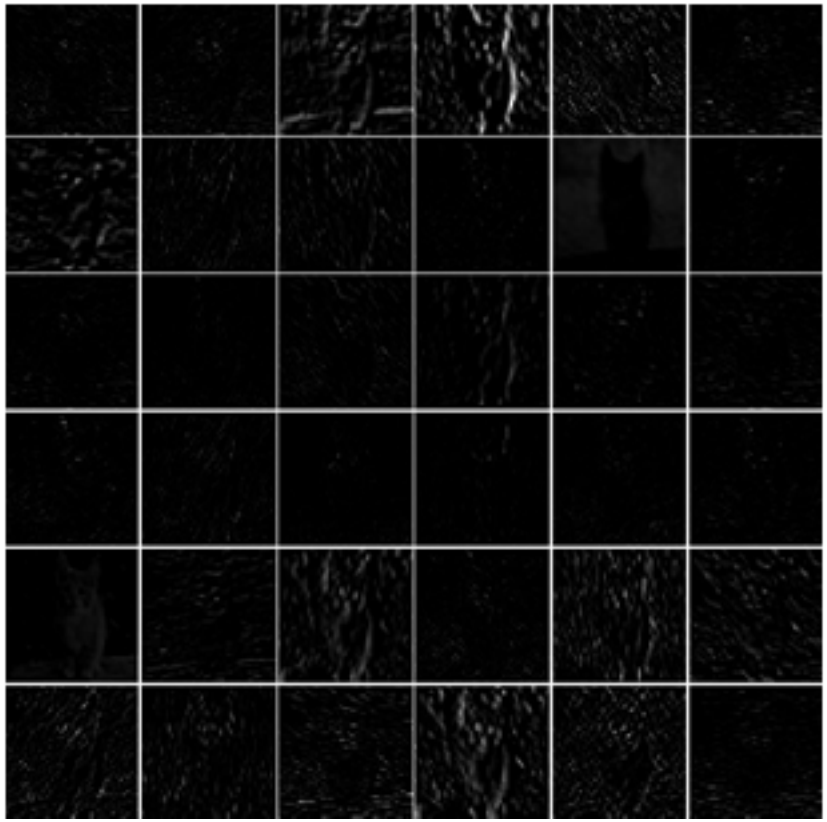
14 октября 2016 года

План

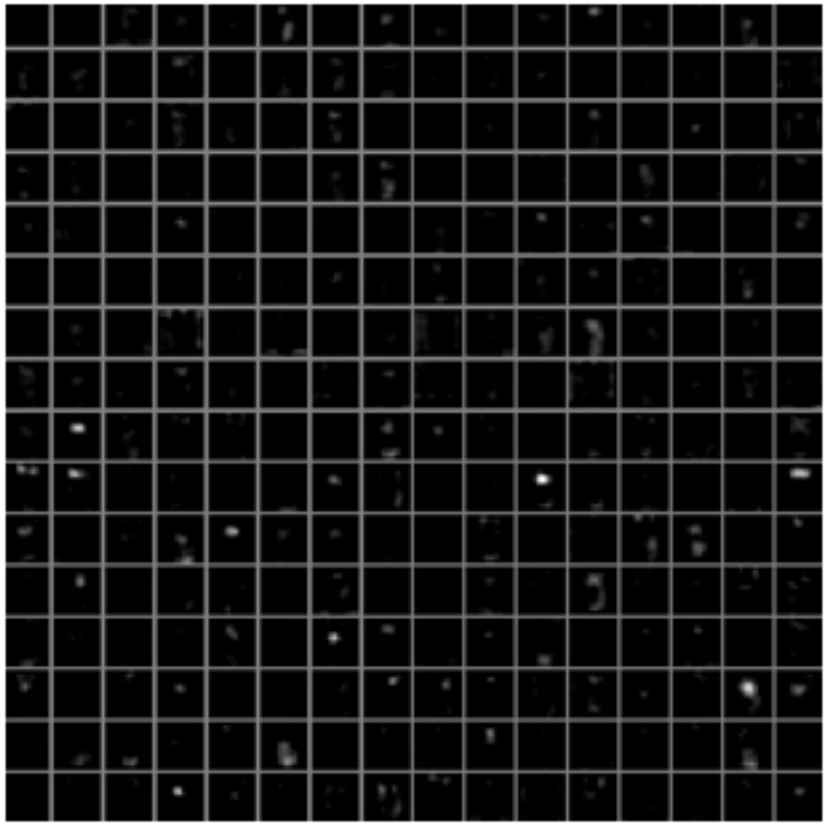
- Визуализация
- Архитектуры

Визуализация работы нейросети

Визуализация активаций (тензоров)



Слой conv1



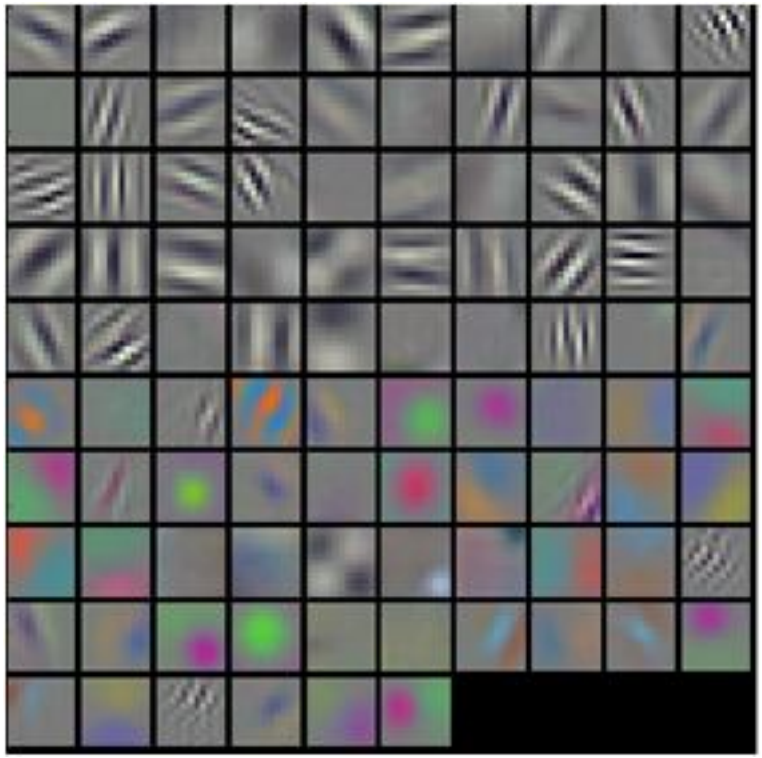
Слой conv5

Обратите внимание на «разреженность» значений

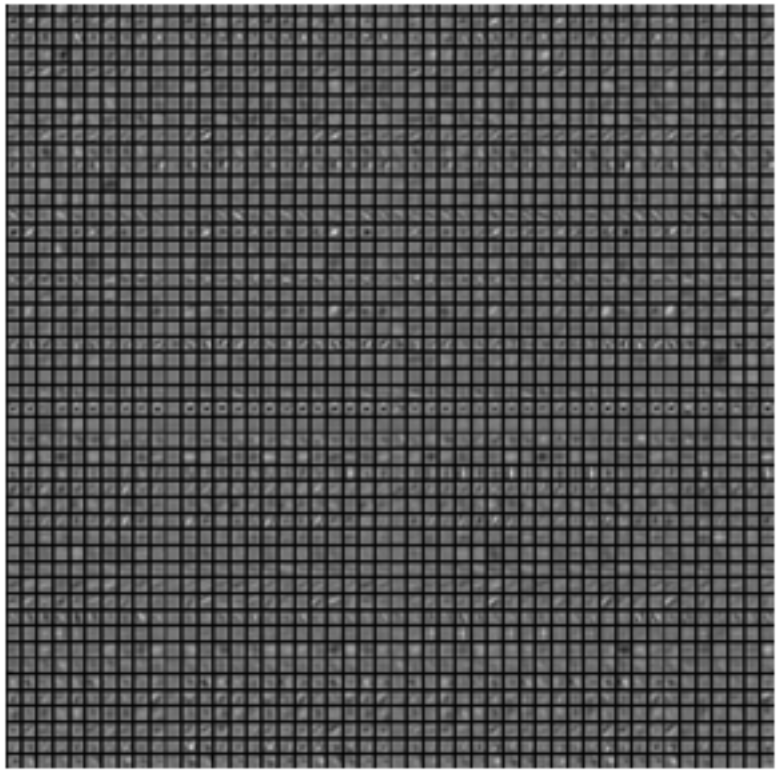
http://cs231n.github.io/understanding-cnn/

Визуализация работы нейросети

Визуализация фильтров



Слой conv1



Слой conv2

http://cs231n.github.io/understanding-cnn/

Визуализация работы нейросети

Изображения, на которых достигается максимальный отклик фильтра

Фильтры слоя pool5

http://cs231n.github.io/understanding-cnn/

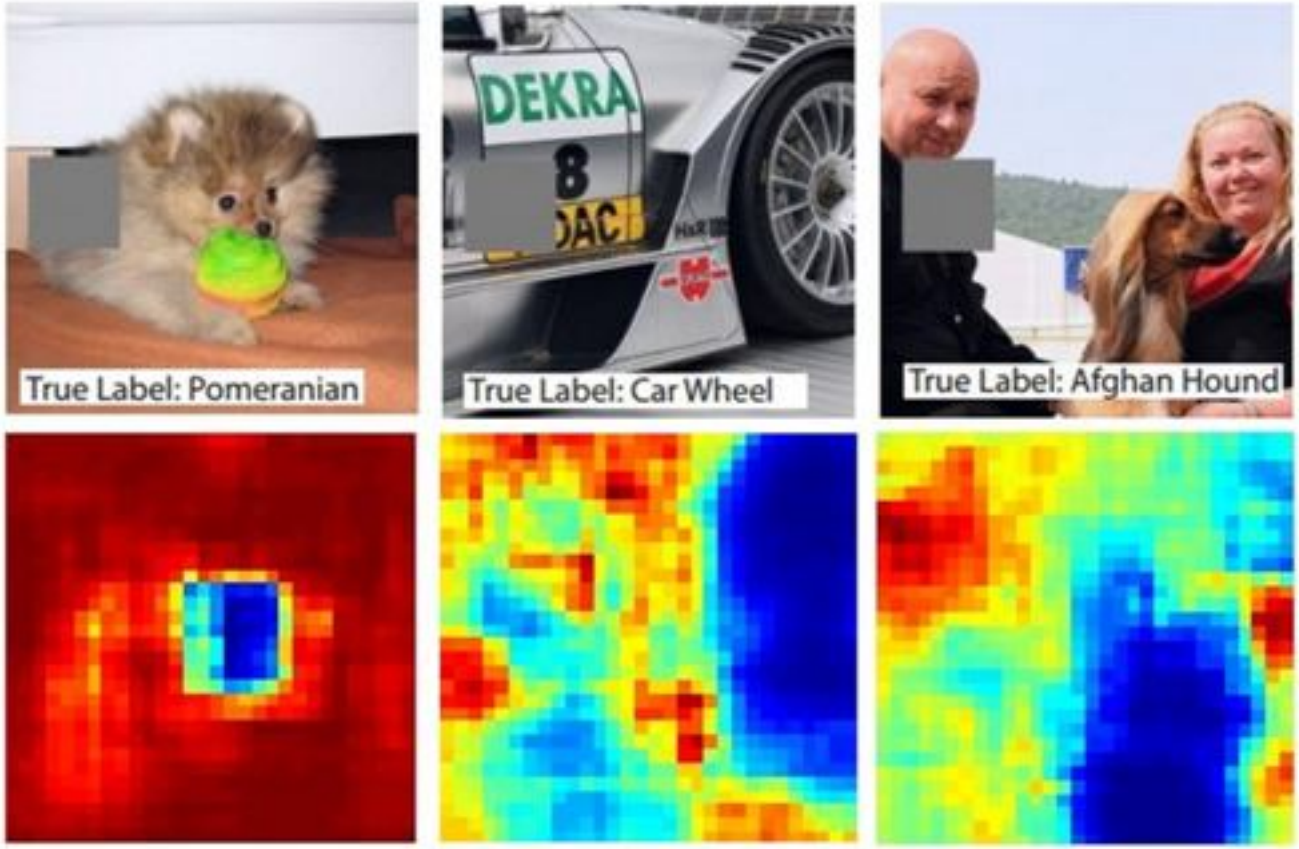
t-SNE

- Можем вычислить L2 расстояние между выходами full6 или full7 слоёв
- Воспользуемся отображением точек из 4096-мерного пространства на 2х мерное, сохраняющее L2 расстояния (приближенно)
- Визуализируем изображения
- Видим, что близкие по смыслу изображения оказываются близки друг к другу

http://cs.stanford.edu/people/karpathy/cnnembed/

Визуализация работы нейросети

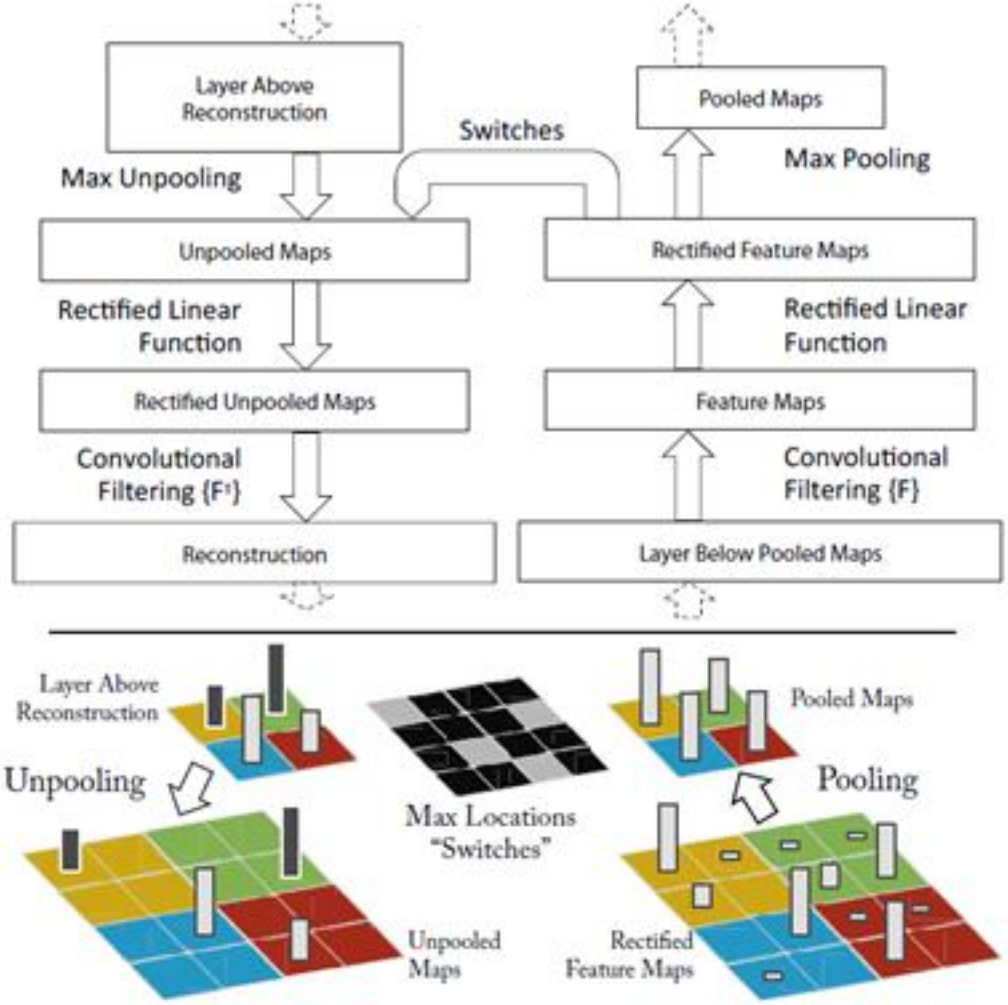
Какой объект на изображении определяет метку?



- Закрываем фрагмент изображения, вычисляем вероятность целевого класса
- Сканируем изображение и строим «heatmap» вероятности объекта целевого класса

http://cs231n.github.io/understanding-cnn/

Deconvolutional network



- Построили сеть для визуализации стимулов, вызвавших активацию определённого нейрона
- Обнуляем все активации, кроме одной
- Запускаем сеть «в обратную сторону»
- Самое важное – «max unpooling»

M. Zeiler and R. Fergus, Visualizing and Understanding Convolutional Networks,ECCV 2014

Пример работы

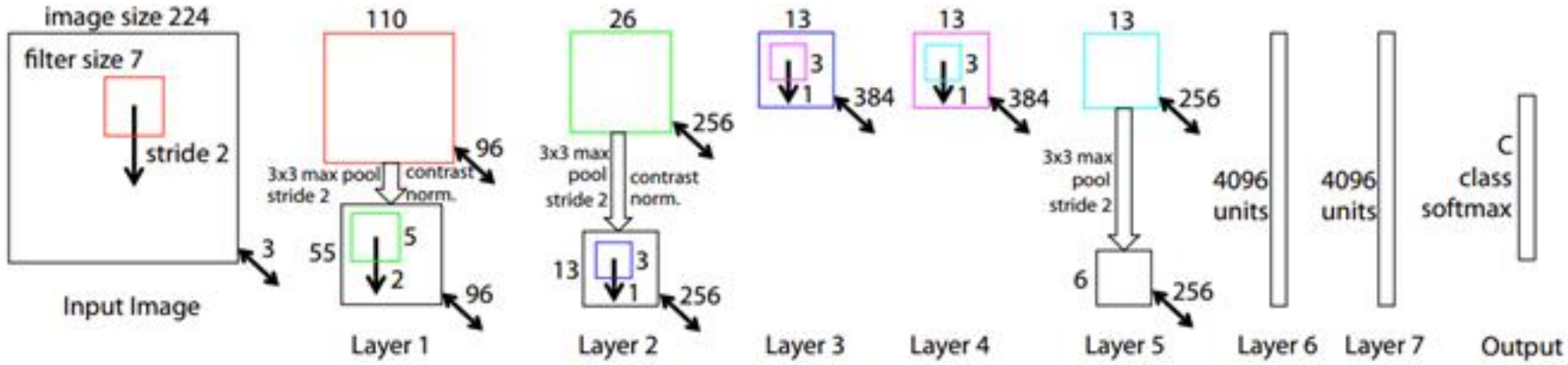
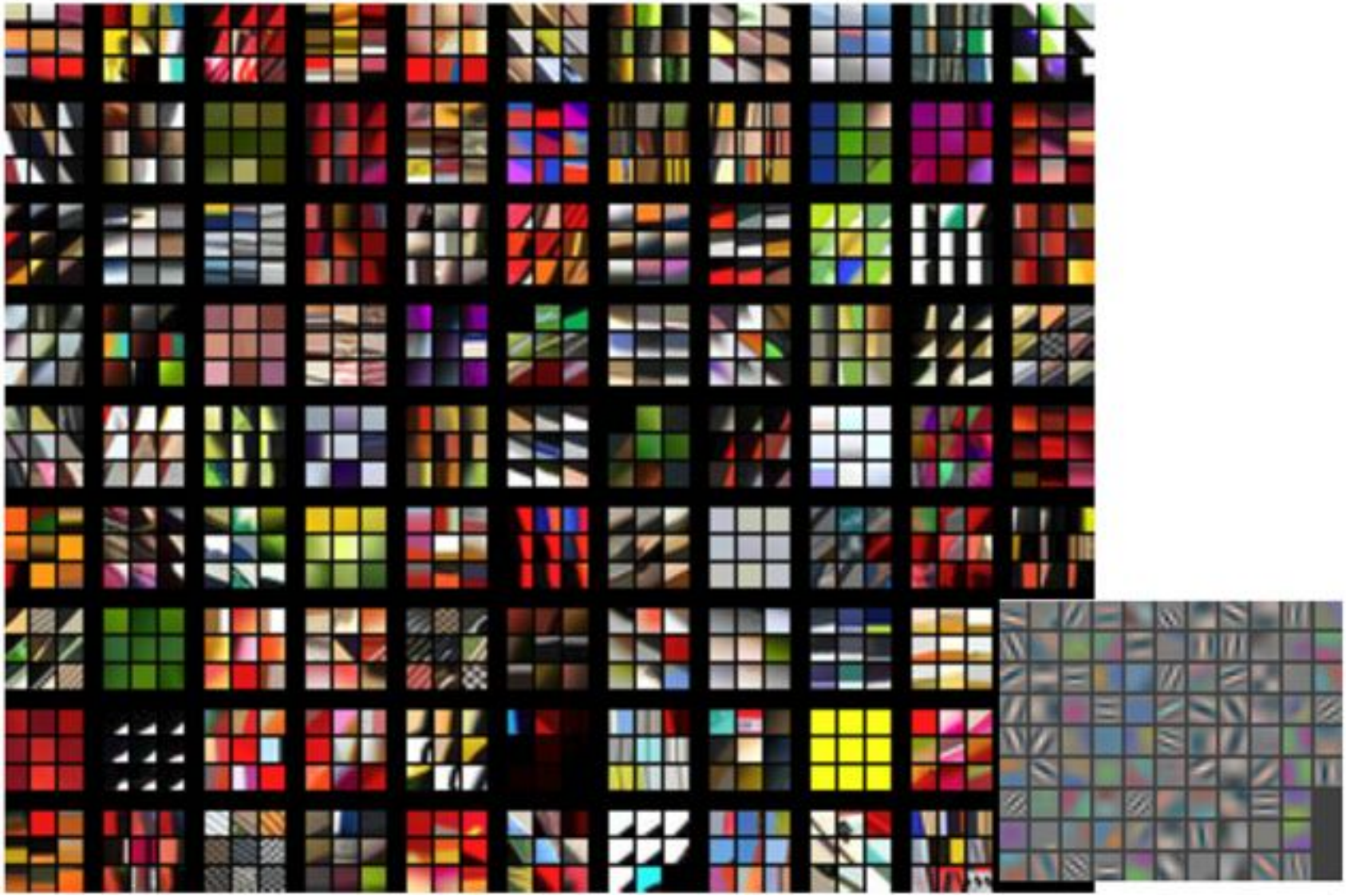


Figure 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C-way softmax function, C being the number of classes. All filters and feature maps are square in shape.

Взяли сеть, похожую на AlexNet с небольшими изменениями

Слой 1: Top-9 фрагментов



Слой 2: Топ-9 фрагментов

Слой 2: Топ-9 фрагментов

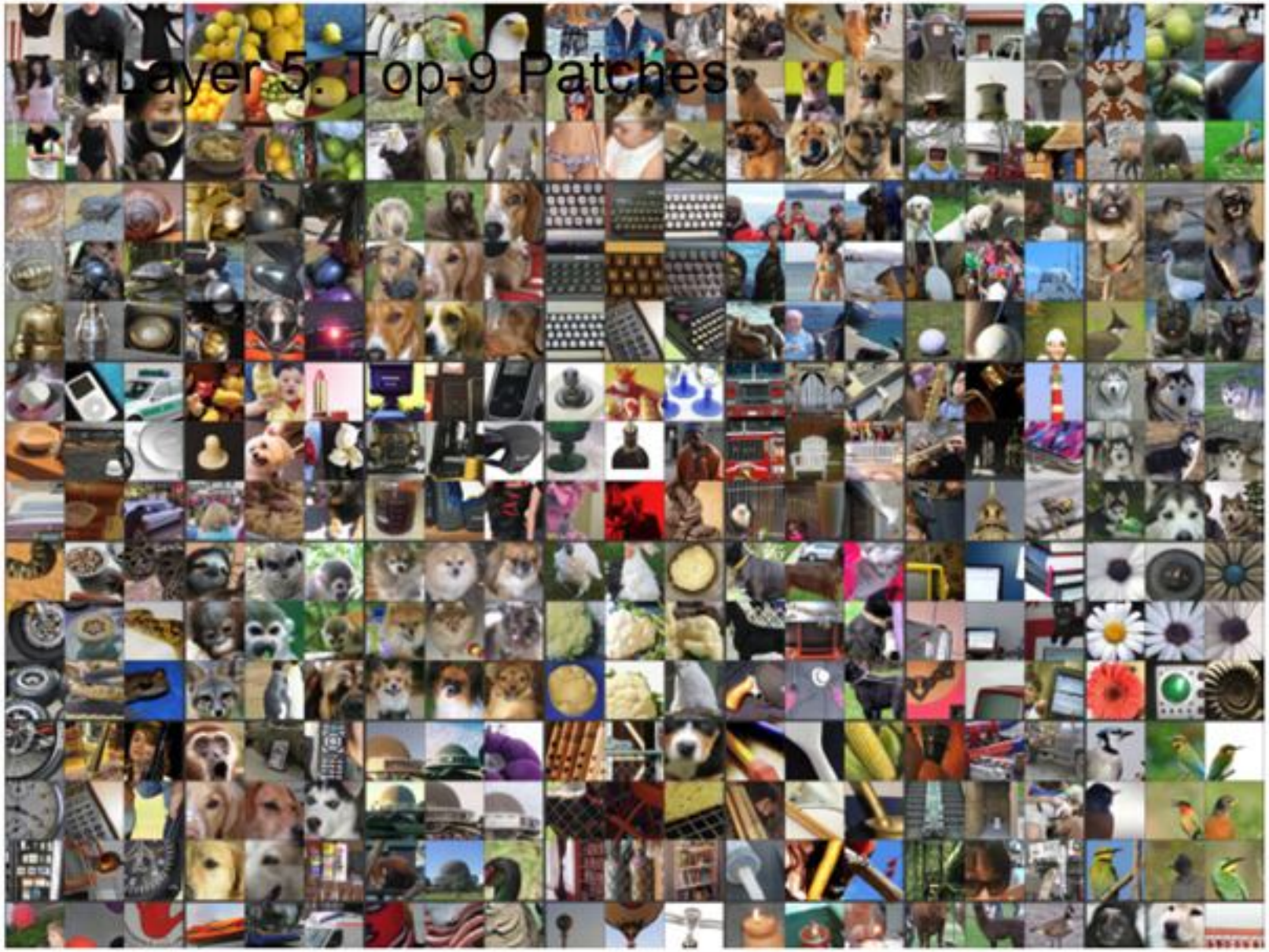
Слой 3: Топ-9 фрагментов

Слой 3: Топ-9 фрагментов

Слой 4: Топ-9 фрагментов

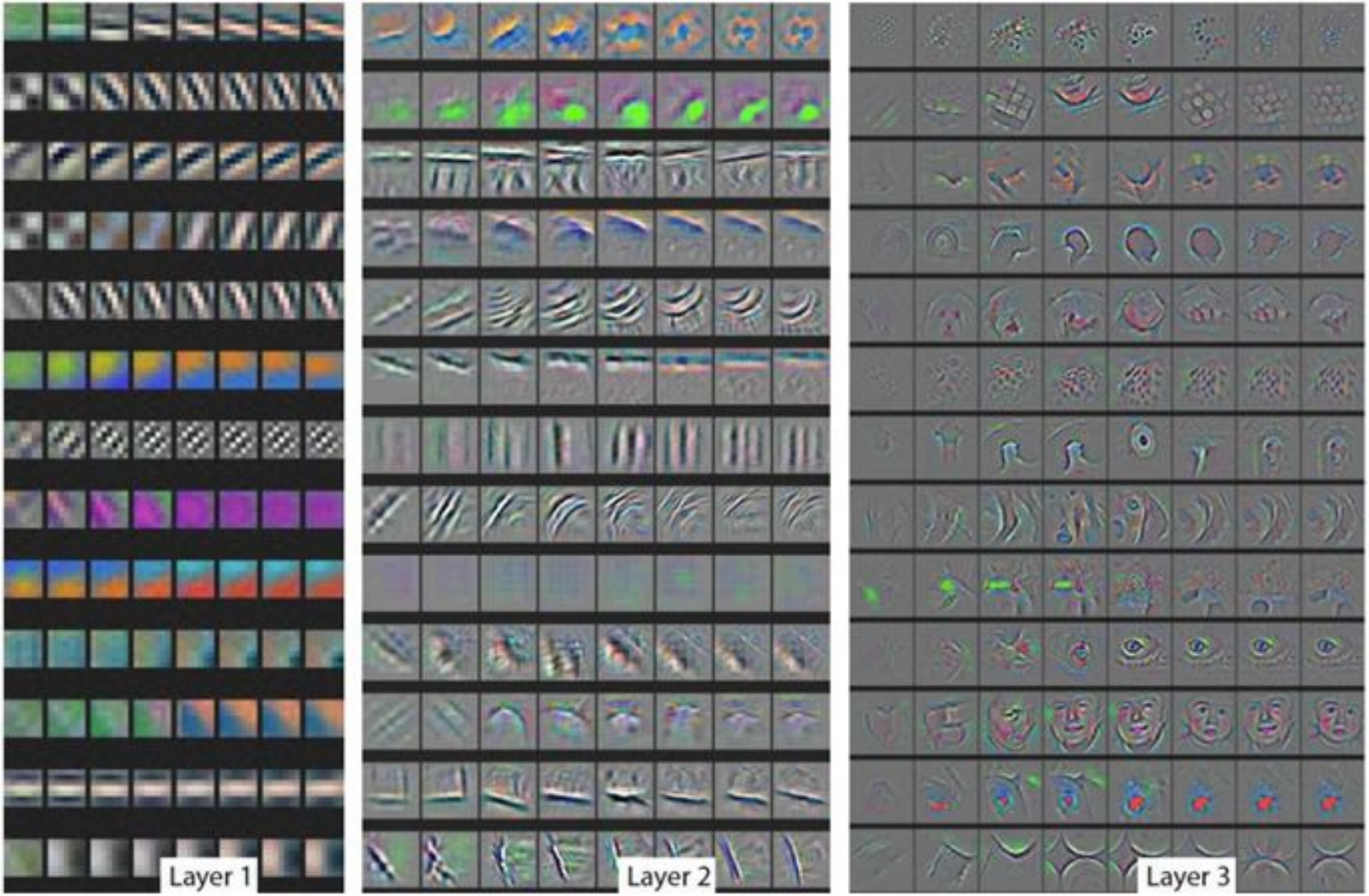
Слой 4: Топ-9 фрагментов

Слой 5: Топ-9 фрагментов

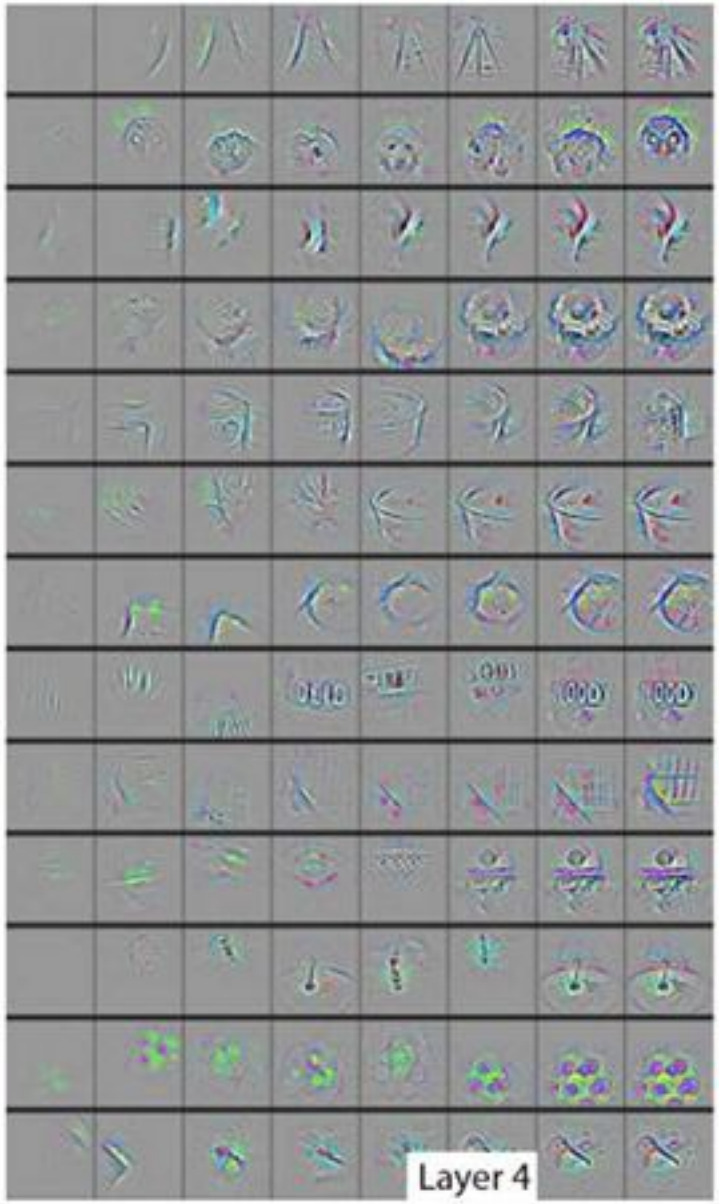


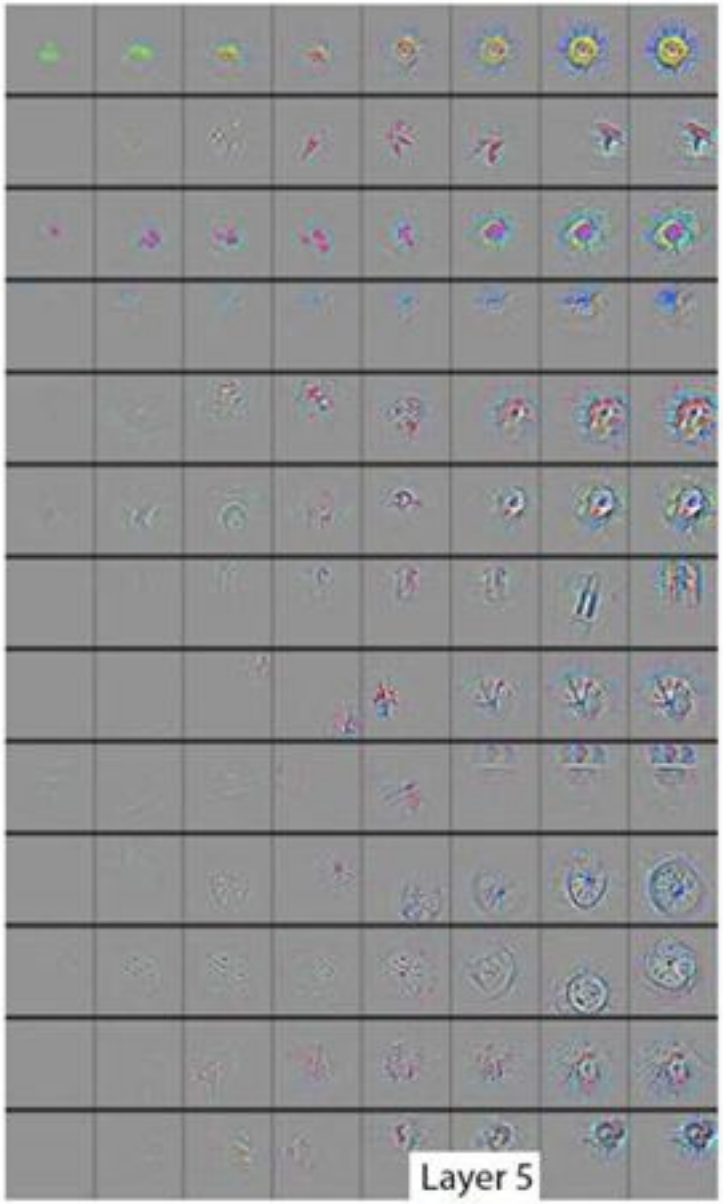
Слой 5: Топ-9 фрагментов

Эволюция признаков



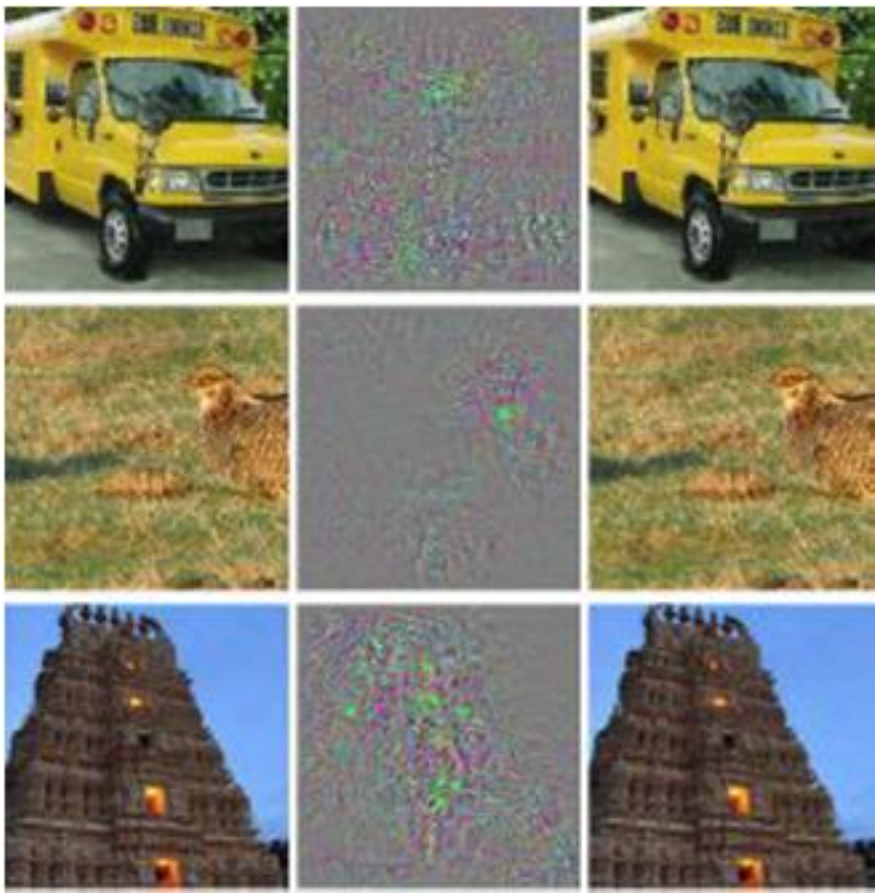
Эволюция признаков





Примеры-соперники

Оптимизацией найдём минимальный сдвиг картинки, достаточный для того, чтобы сеть выдавала другую метку



Исходный Дистория Соперник

Исходный Дистория Соперник

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, **Intriguing properties of neural networks**,ICLR 2014

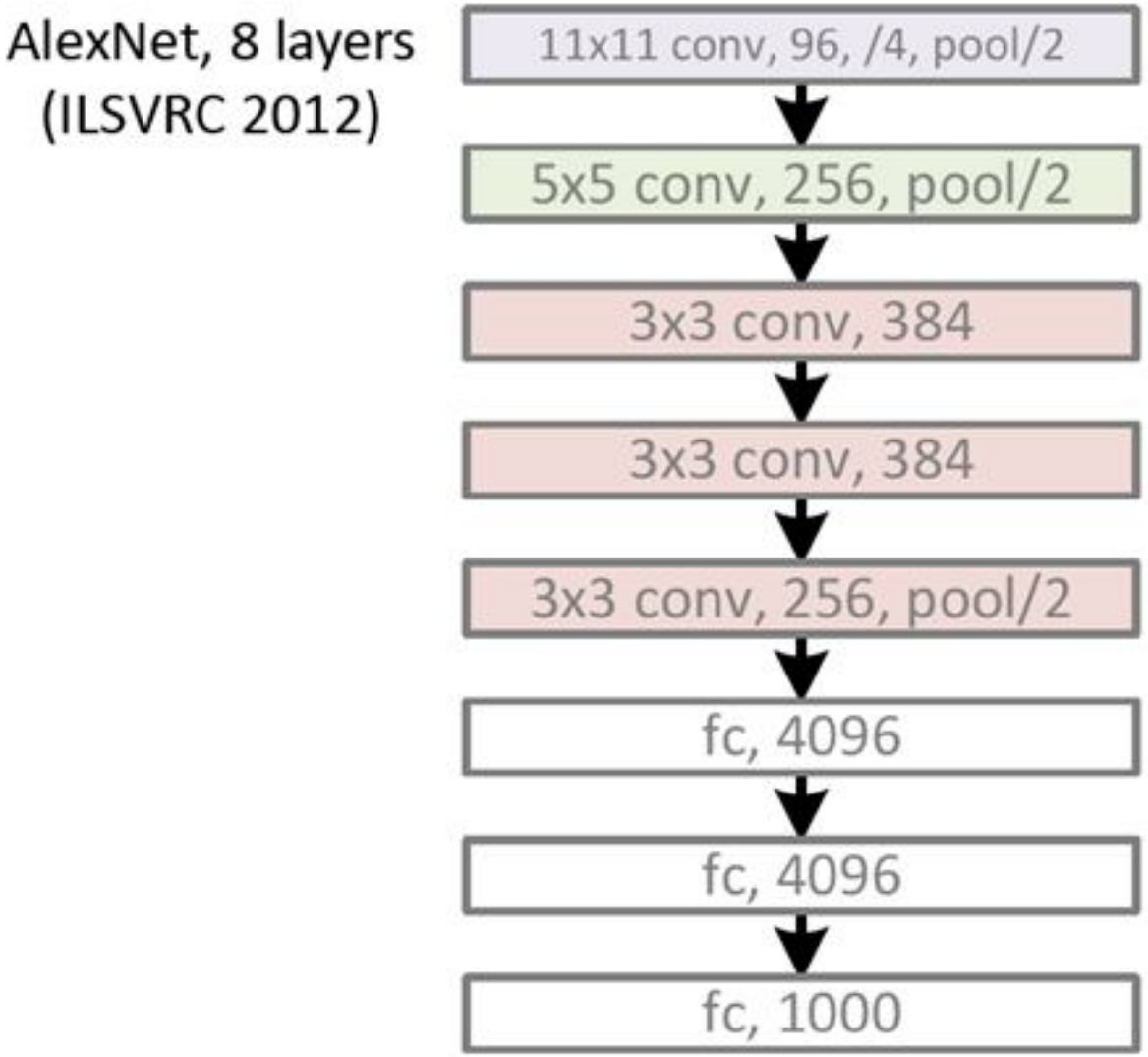
Резюме визуализации

- Признаки более высоких слоёв имеют «семантическое» значение
- Можно наблюдать, как оно проявляется при обучении нейросети
- Сеть при классификации учится и локализовывать объект
- Идея «деконволюции» оказалась интересной, и в дальнейшем используется
- Нейросеть «очень» нелинейная, и не всегда соблюдаются условия локальности

План

- Визуализация
- Архитектуры

AlexNet



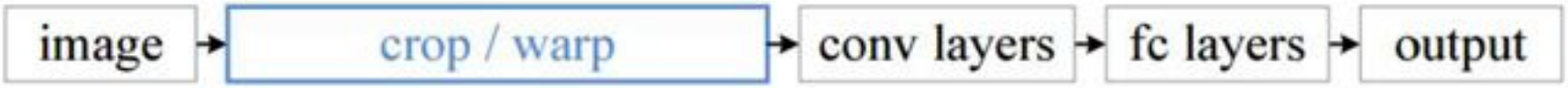
Krizhevsky A., Sutskever I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012

Spatial Pyramid Pooling (SPP)

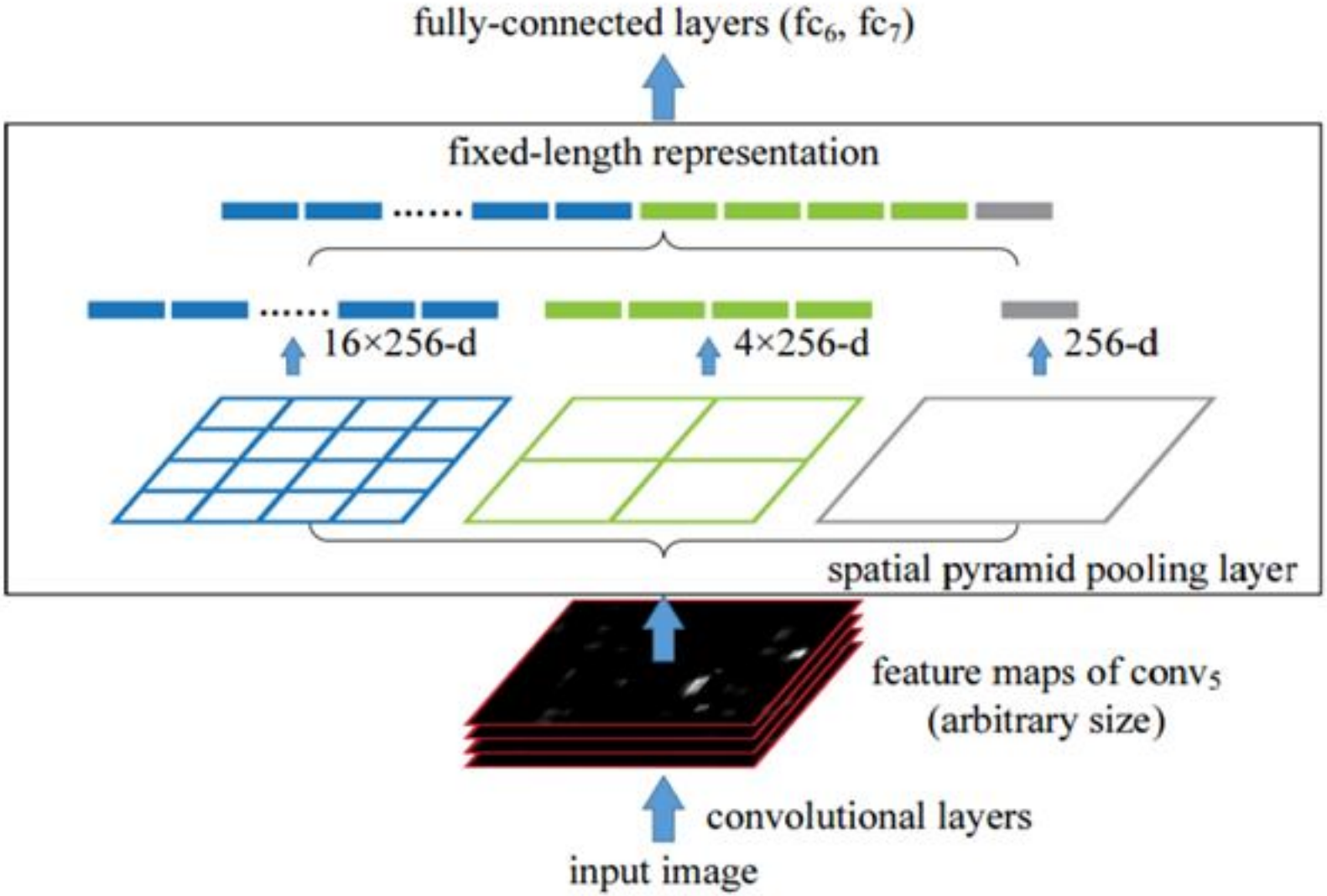
В большинстве архитектур присутствует проблема фиксированного размера входного изображения, виновник – полносвязный слой.

crop

warp



Spatial Pyramid Pooling (SPP)



Spatial Pyramid Pooling (SPP)

Практически любая архитектура может быть адаптирована для работы с изображениями разного размера путем замены последнего pooling слоя перед полносвязными на SPP слой

		top-1 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	35.99	34.93	34.13	32.01
(b)	SPP single-size trained	34.98 (1.01)	34.38 (0.55)	32.87 (1.26)	30.36 (1.65)
(c)	SPP multi-size trained	34.60 (1.39)	33.94 (0.99)	32.26 (1.87)	29.68 (2.33)

		top-5 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	14.76	13.92	13.52	11.97
(b)	SPP single-size trained	14.14 (0.62)	13.54 (0.38)	12.80 (0.72)	11.12 (0.85)
(c)	SPP multi-size trained	13.64 (1.12)	13.33 (0.59)	12.33 (1.19)	10.95 (1.02)

Очень глубокие (VGG)

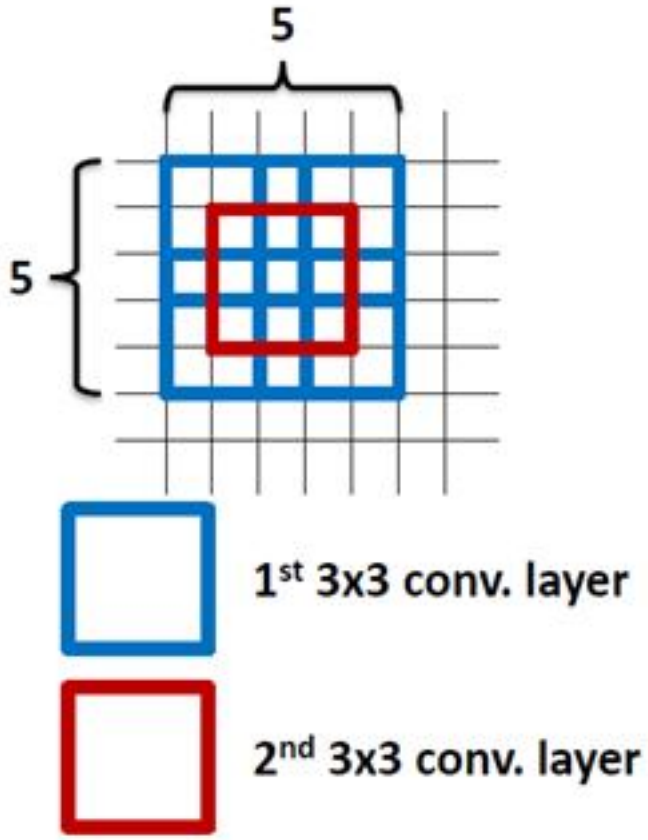
Идеи:

- Исследовать рост качества за счёт увеличения глубины нейросети
- Использовать только маленькие 3x3 свёртки
- Stride 1 в свёртках чтобы не терять информацию
- ReLU активация
- Нет нормализации
- Уменьшение разрешения через maxpooling
- Число фильтров x2 при уменьшении разрешения в 2 раза

K. Simonyan, A. Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition . ICLR 2015

Свёртки 3x3

- Стек свёрток позволяет обеспечить бОльшее рецептивное поле (reception field)
- 5x5 для 2-х свёрток
- 7x7 для 3-х свёрток
- БОльшая нелинейность за счёт ReLU активаций
- Меньше параметров
  - 18x (2 3x3) vs 25x (5x5)
  - 27x (3 3x3) vs 49x (7x7)



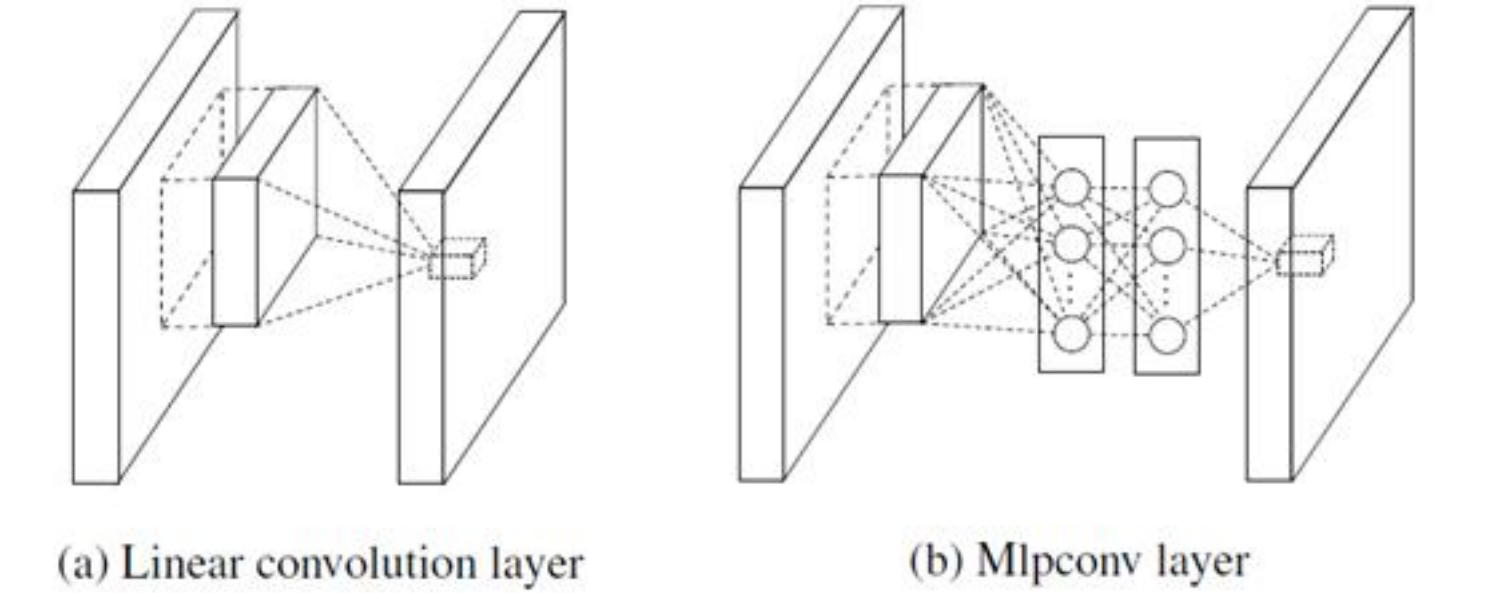
Исследование вариантов

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					

D - VGG-16

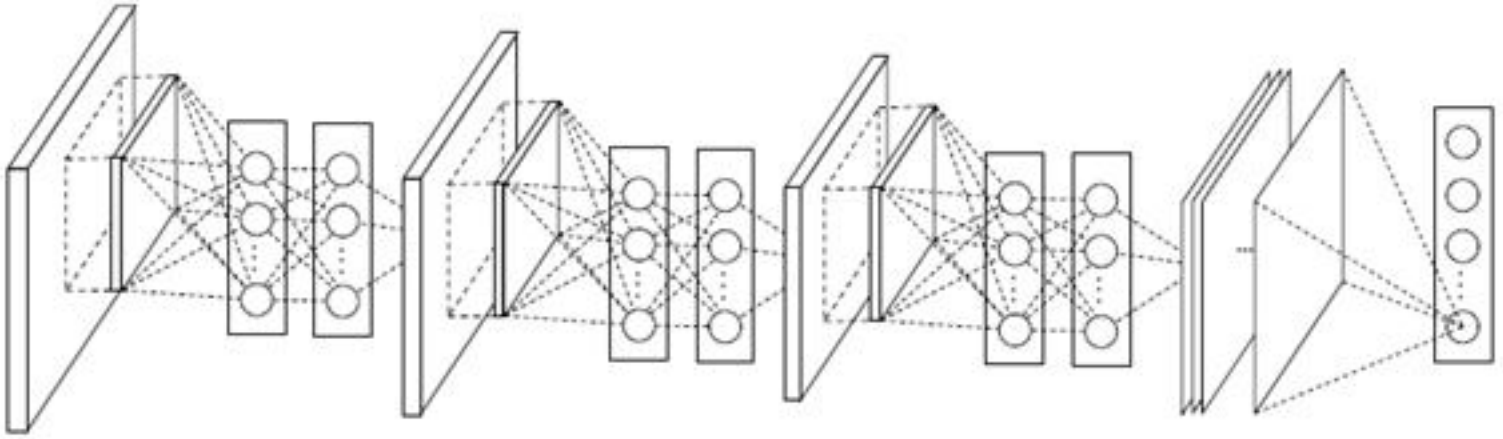
E - VGG-19

Network in Network



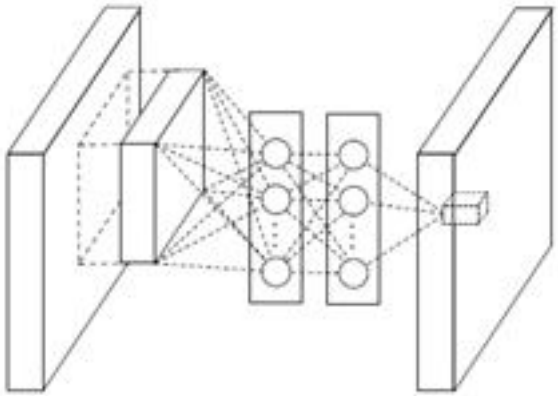
(a) Linear convolution layer

(b) Mlpconv layer

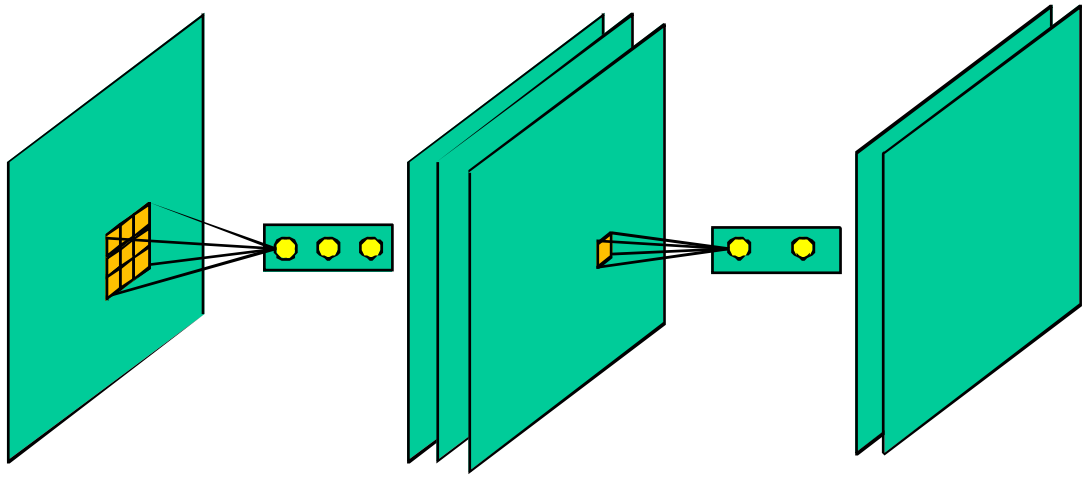


M. Lin, Q. Chen, S. Yan Network In Network. ICLR, 2014

Свёртка 1x1

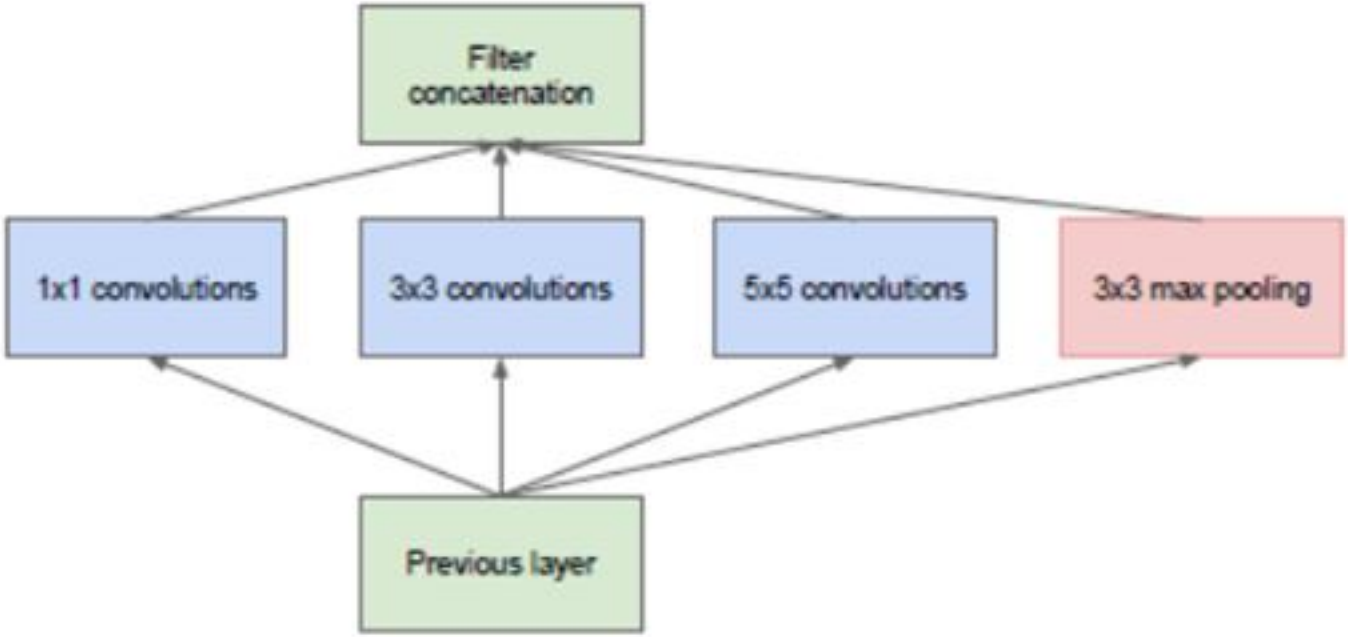


(b) Mlpconv layer



- Мы можем реализовать второй и далее слои «вложенного» перспептрона как 1x1 свёртку с предыдущим слоем
- Можем управлять «глубиной» тензора, регулируя k - число свёрток 1x1, по сравнению с n – глубиной предыдущего тензора
  - K < N, значит мы уменьшили до k глубину тензора (сжали)
  - K > N, значим мы увиличили глубину тензора
- Можно трактовать как набор локальных классификаторов

Модуль Inception

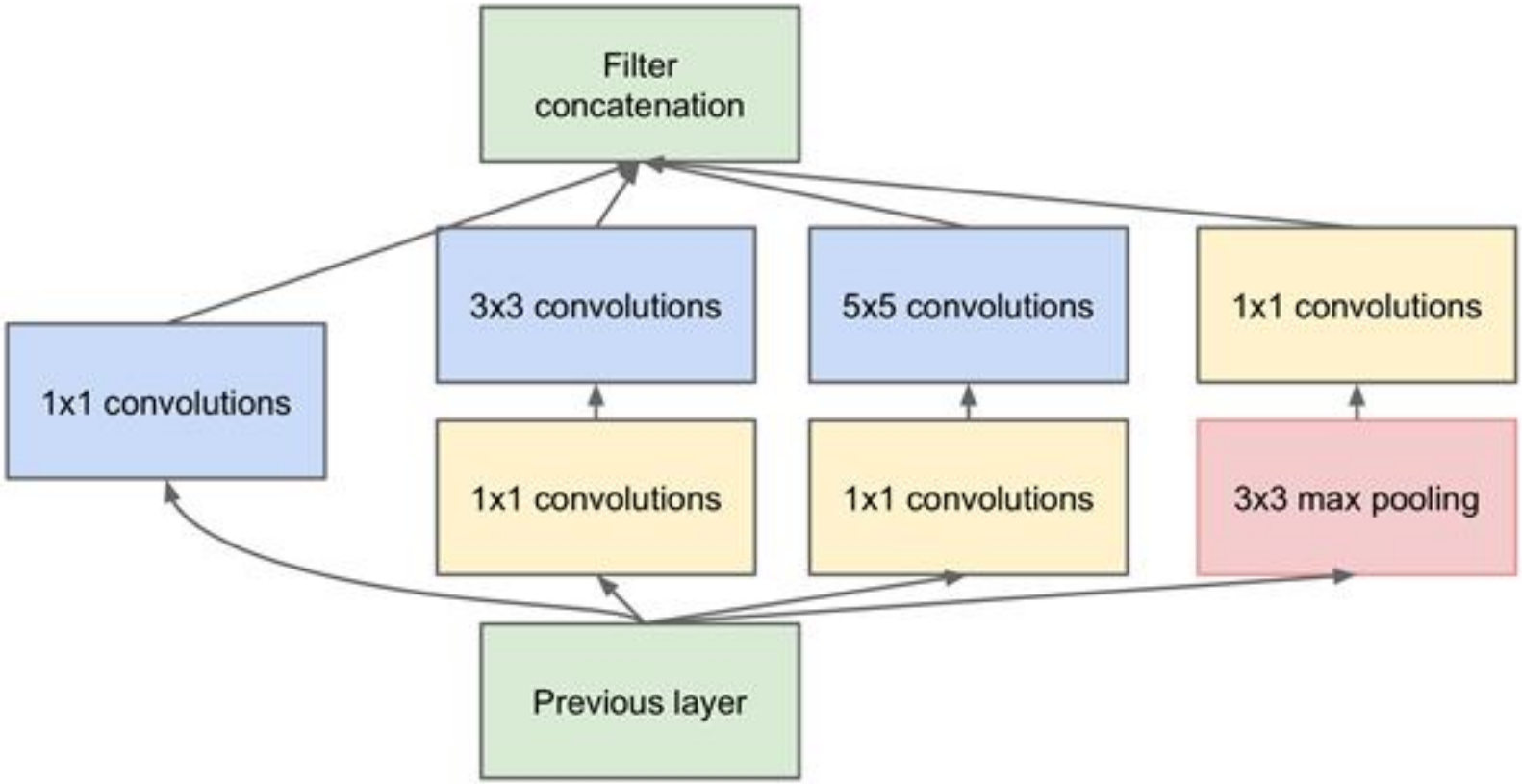


(a) Inception module, naïve version

В чём смысл 1x1 свёрток?

Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

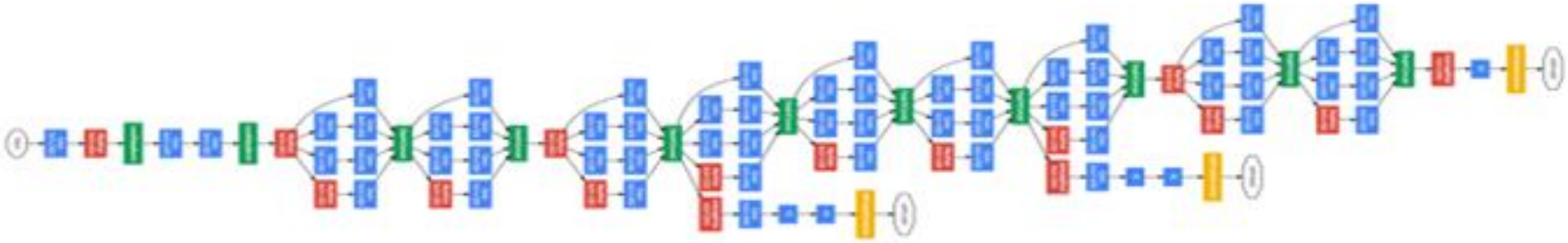
Модуль Inception



(b) Inception module with dimension reductions

Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

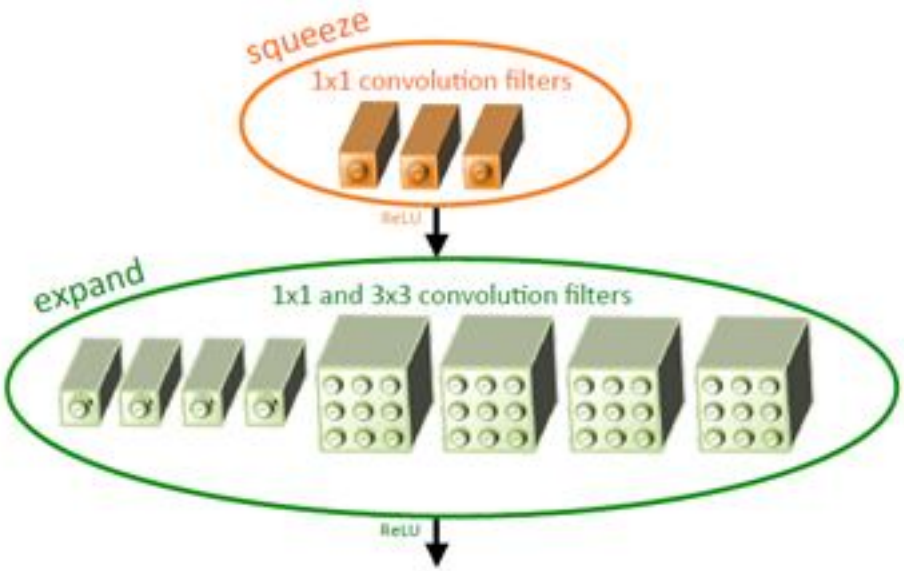
Архитектура Inception

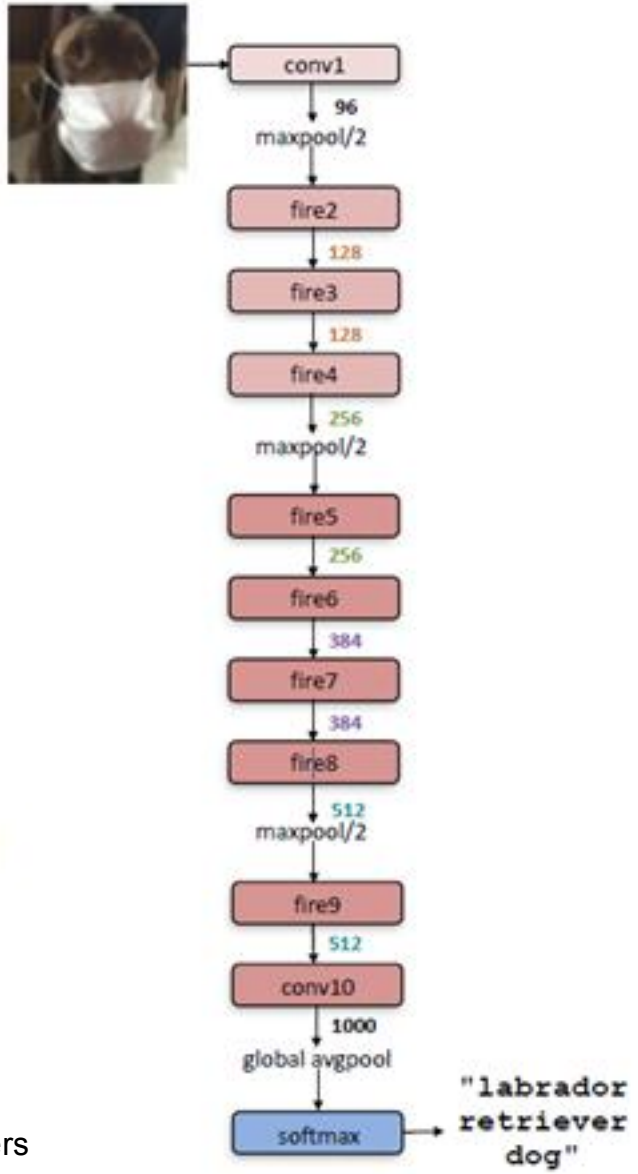


- Глубокая сеть
- Inception-модули
- Несколько уровней supervision

Christian Szegedy et. al. Going deeper with convolutions. CVPR 2015

SqueezeNet





- Активно использовать 1x1 свёртки для уменьшения числа параметров
- «Сжимать» мы будем для того, чтобы на вход 3x3 фильтрам подавать меньше данных

Forrest N. Iandola et.al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. ArXiv 2016

SqueezeNet

layer name/type	output size	filter size / stride (if not a fire layer)	depth	s_{1x1} (#1x1 squeeze)	e_{1x1} (#1x1 expand)	e_{3x3} (#3x3 expand)	s_{1x1} sparsity	e_{1x1} sparsity	e_{3x3} sparsity	# bits	#parameter before pruning	#parameter after pruning
input image	224x224x3										-	-
conv1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6bit	14,208	14,208
maxpool1	55x55x96	3x3/2	0									
fire2	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	11,920	5,746
fire3	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	12,432	6,258
fire4	55x55x256		2	32	128	128	100%	100%	**33%**	6bit	45,344	20,646
maxpool4	27x27x256	3x3/2	0									
fire5	27x27x256		2	32	128	128	100%	100%	**33%**	6bit	49,440	24,742
fire6	27x27x384		2	48	192	192	100%	**50%**	**33%**	6bit	104,880	44,700
fire7	27x27x384		2	48	192	192	**50%**	100%	**33%**	6bit	111,024	46,236
fire8	27x27x512		2	64	256	256	100%	**50%**	**33%**	6bit	188,992	77,581
maxpool8	13x12x512	3x3/2	0									
fire9	13x13x512		2	64	256	256	**50%**	100%	**30%**	6bit	197,184	77,581
conv10	13x13x1000	1x1/1 (x1000)	1				**20%** (3x3)			6bit	513,000	103,400
avgpool10	1x1x1000	13x13/1	0									
	activations			parameters				compression info			1,248,424 (total)	**421,098** (total)

Сжатие модели

CNN architecture	Compression Approach	Data Type	Original → Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
AlexNet	None (baseline)	32 bit	240MB	1x	57.2%	80.3%
AlexNet	SVD [5]	32 bit	240MB → 48MB	5x	56.0%	79.4%
AlexNet	Network Pruning [11]	32 bit	240MB → 27MB	9x	57.2%	80.3%
AlexNet	Deep Compression [10]	5-8 bit	240MB → 6.9MB	35x	57.2%	80.3%
SqueezeNet (ours)	None	32 bit	4.8MB	**50x**	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	8 bit	4.8MB → 0.66MB	**363x**	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	6 bit	4.8MB → 0.47MB	**510x**	57.5%	80.3%

Размер – объём данных на диске.

S. Han, H. Mao, and W. Dally. Deep compression: Compressing DNNs with pruning, trained quantization and huffman coding. arxiv:1510.00149v3, 2015

Революция глубины

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

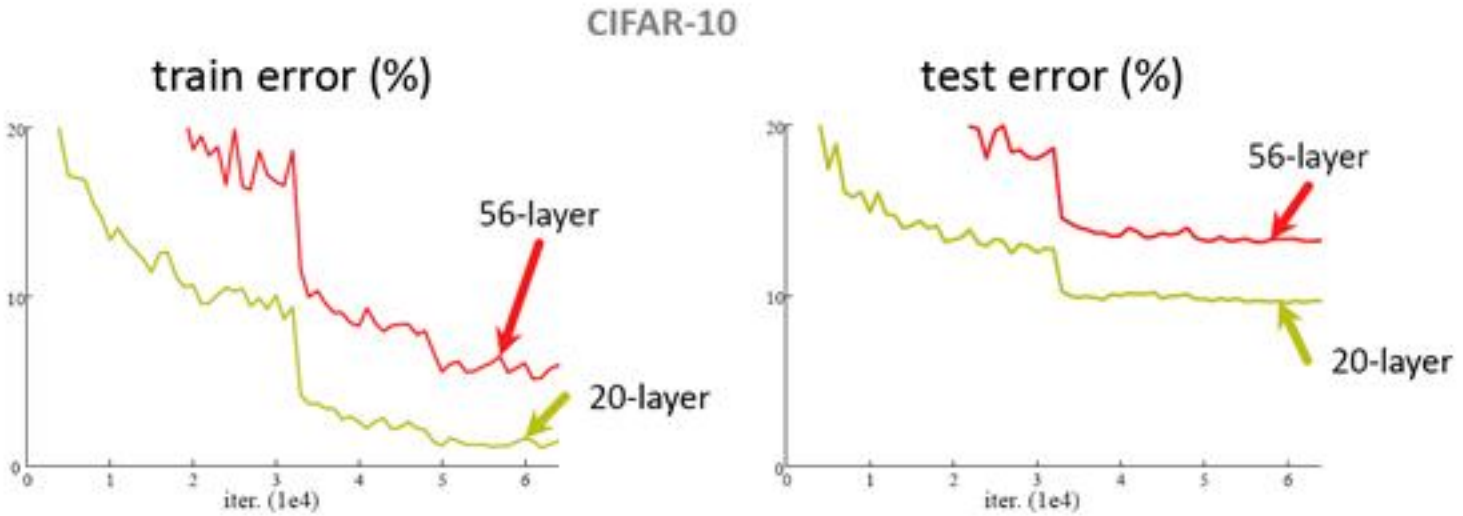
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

ICCV15

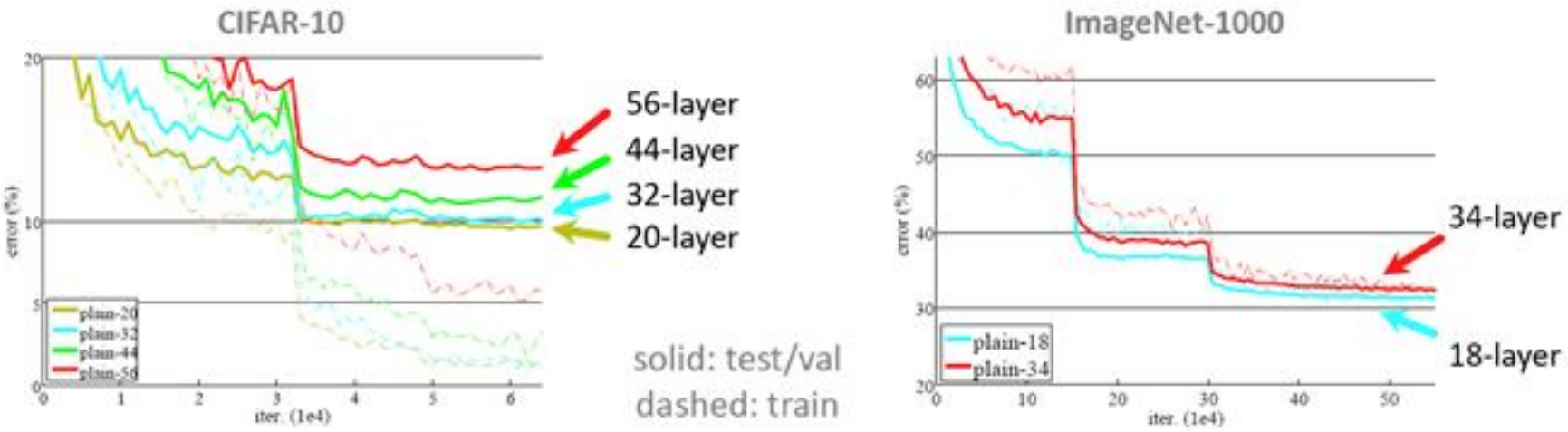
Проблема добавления слоёв

Simply stacking layers?



- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

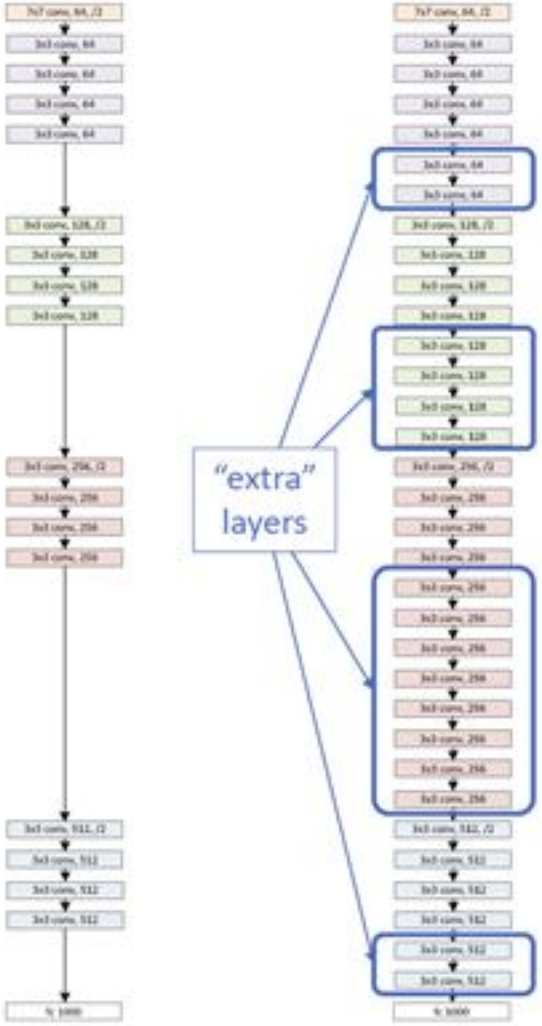
Проблема добавления слоёв



- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Добавление с сохранением функции

a shallower model (18 layers)



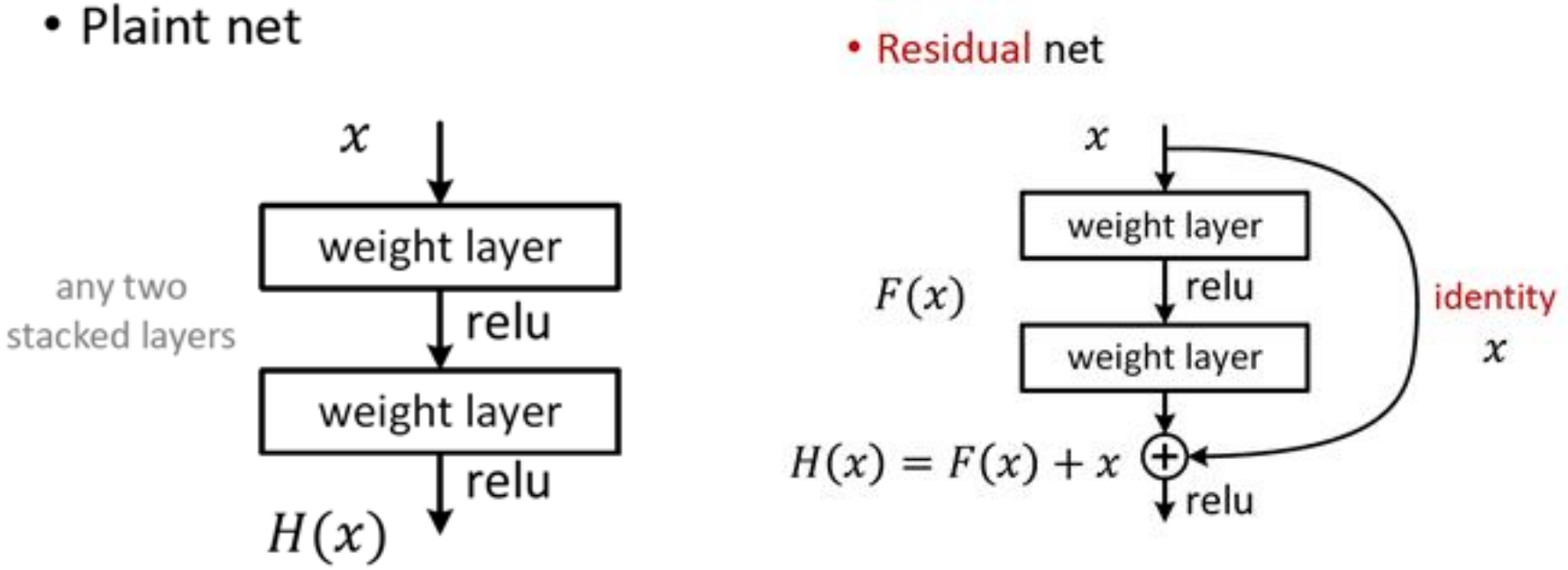
a deeper counterpart (34 layers)

Research

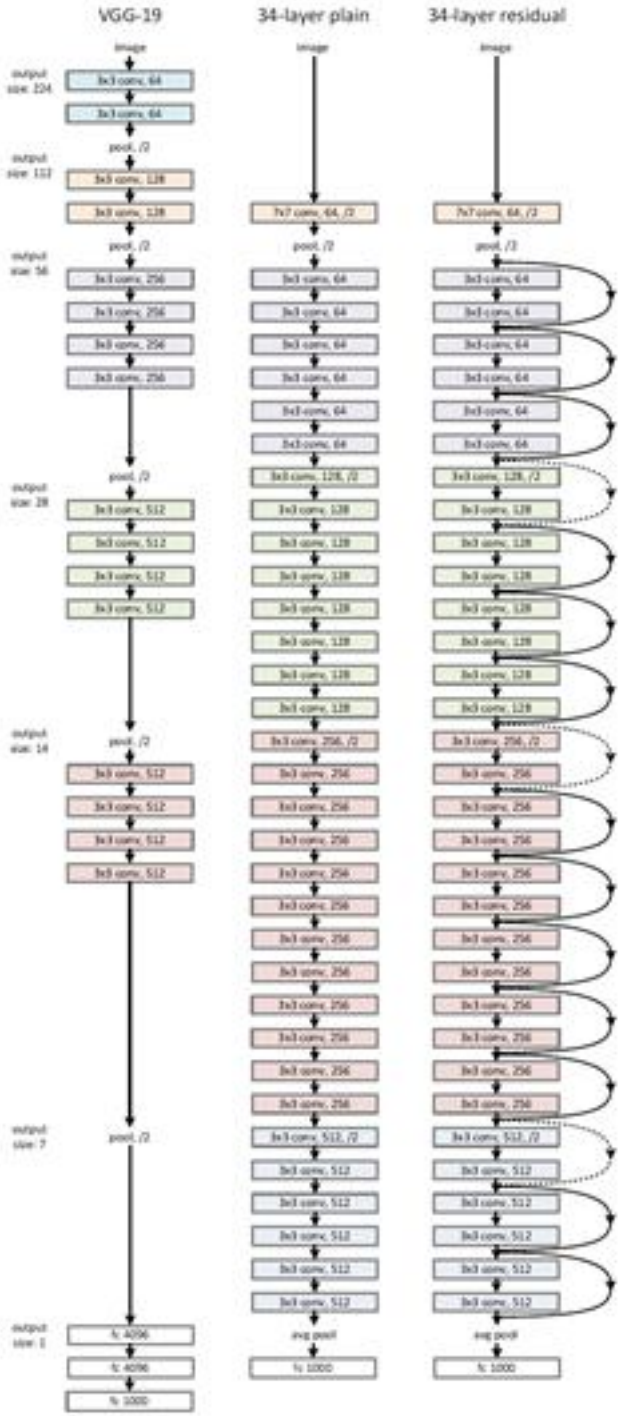
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

ICCV15

Residual net

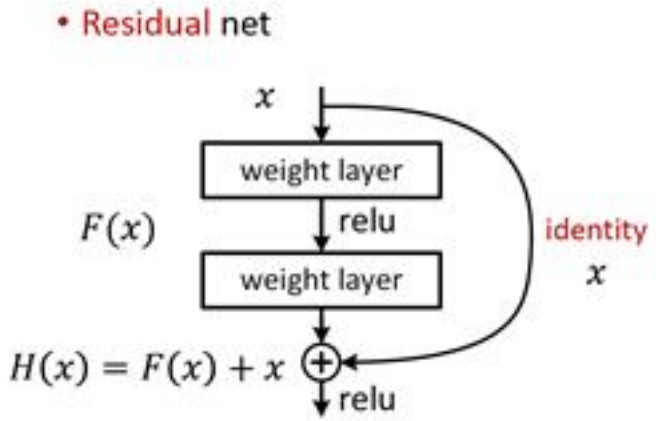


- Будем учить не преобразование, а пертурбацию тождественного преобразования
  - Если единичное преобразование оптимально, тогда мы его сохраняем
  - Небольшие флуктуации оказывается обучать проще



Базовая модель

- Свёртки 3x3
- Subsampling через свёртку с шагом 2

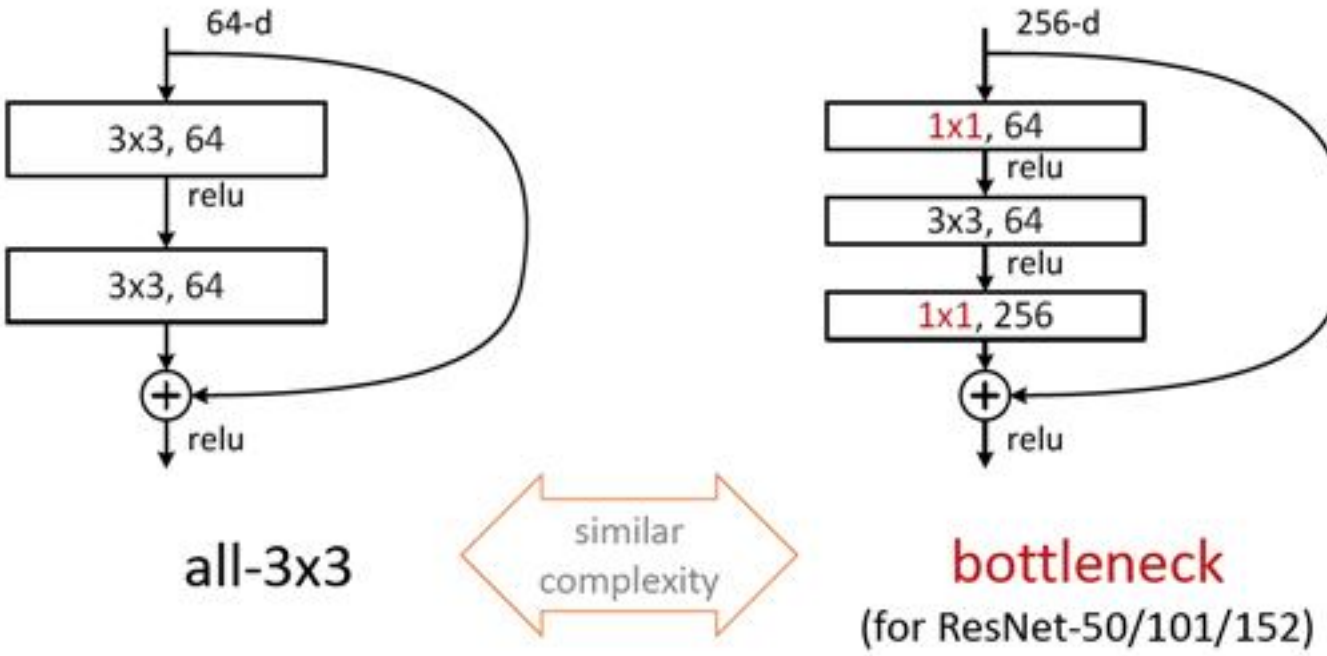


При изменении размеров тензоров пробуют варианты:

- Добавление нулями
- Линейная проекция
- Шаг 2

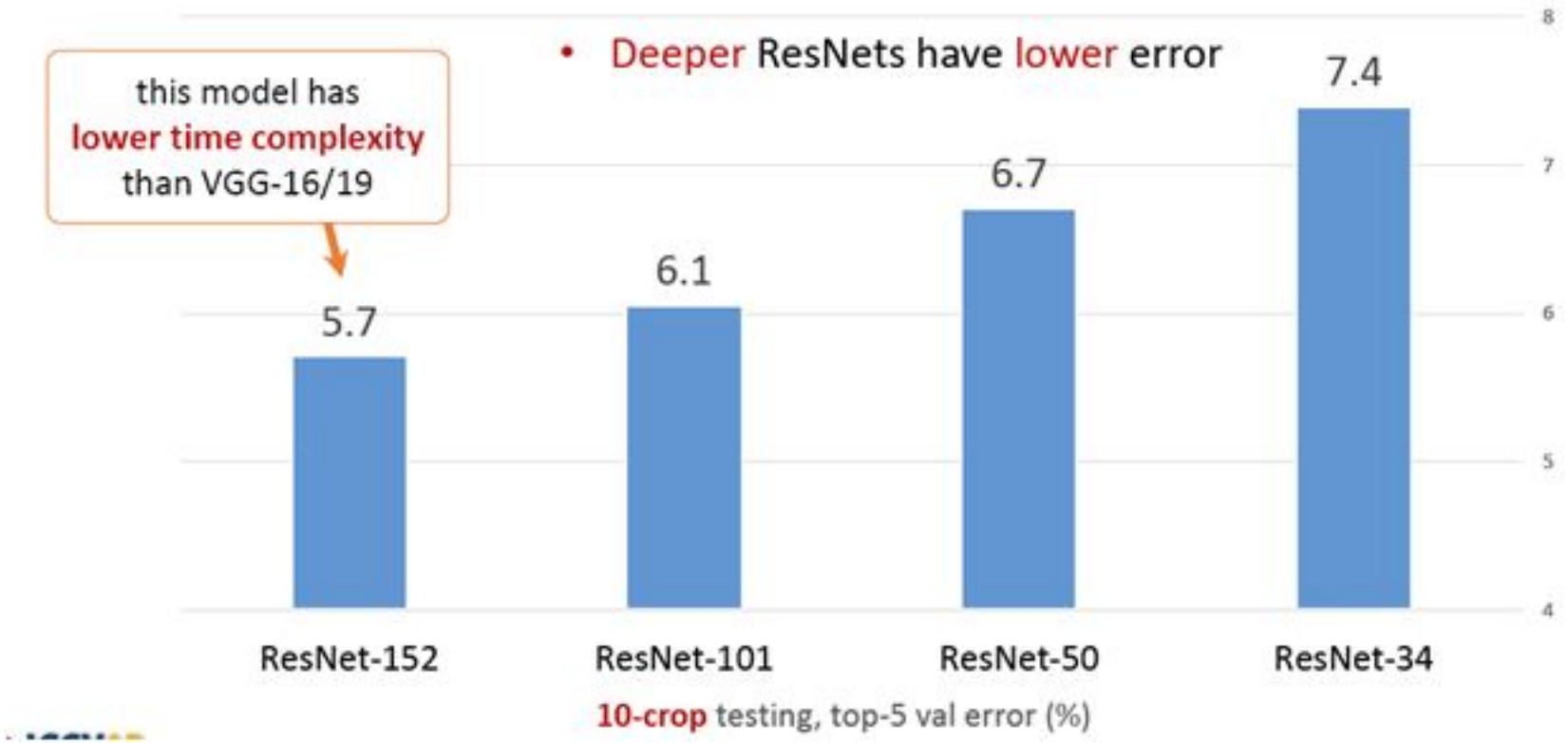
Блок для очень глубоких сетей

- A practical design of going deeper

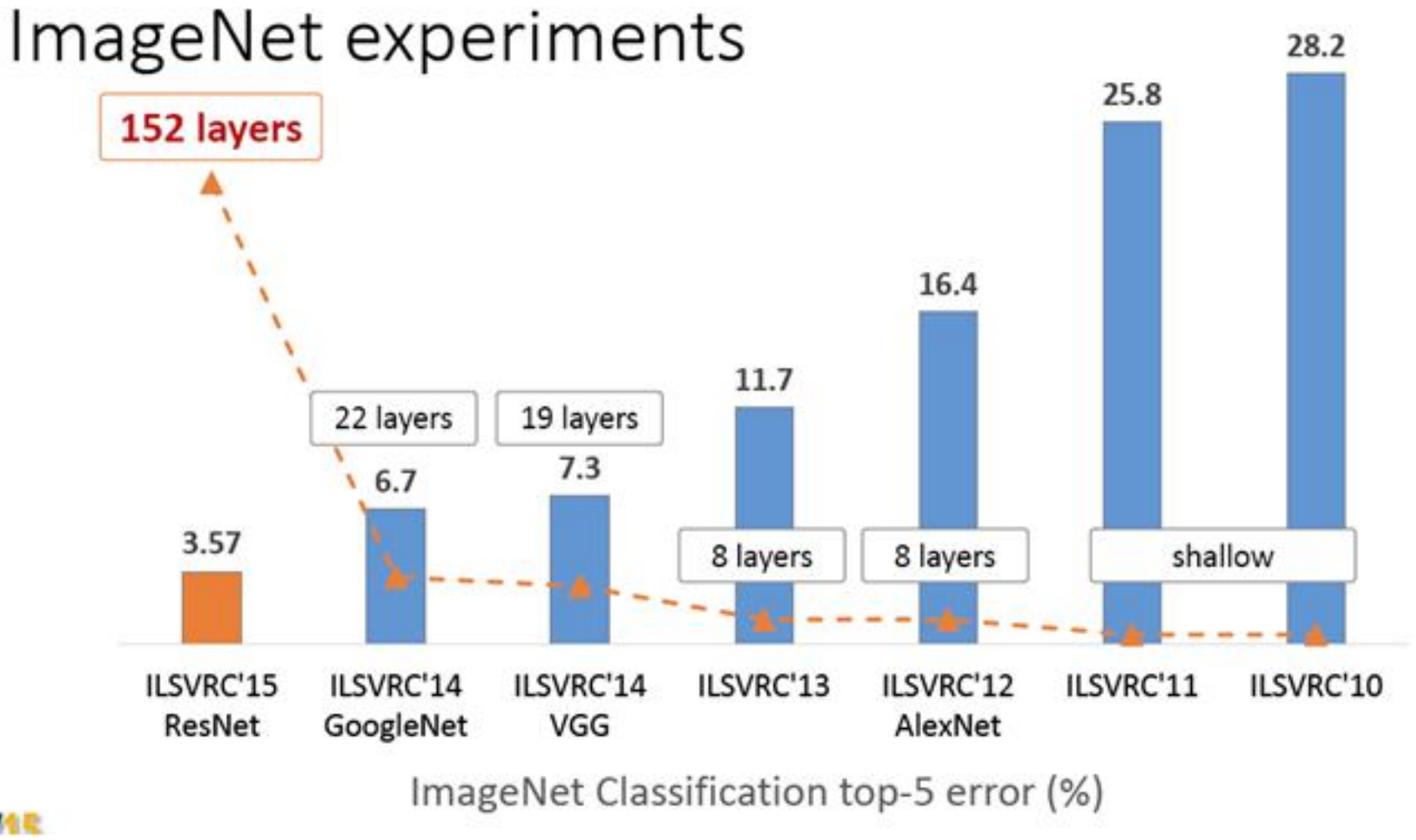


- Понижение размерности (256->64)
- Свёртка 3x3 на тензоре меньшей глубины
- Повышение размерности

Результаты на ImageNet



Результаты на ImageNet



Резюме

- 1x1 свёртки позволяют управлять сжатием/расжатием данных и уменьшать число параметров
- При этом 1x1 свёртки тоже вычисляют интересные признаки
- Residual Learning позволяет обучать «добавку», и за счёт этого обучать сверхглубокие модели
- Есть несколько готовых архитектур и блоков, которые используются как основы для других алгоритмов
  - AlexNet, SqueezeNet, VGG-16, Inception, ResNet