

Визуализация нейронных сетей и генерация изображений

Александр Дьяконов

4 ноября 2021 года

План

Зачем наблюдать? – За чем наблюдать?

**Визуализация весов: свёртки первого слоя / промежуточных слоёв
«deconvnet» / Class Activation Maps (CAM) / Guided Backpropagation
Interpretable Convolutional Neural Networks / Grad-CAM / FullGrad**

Стандартные средства в признаковых пространствах

Анализ активации нейронов

Occlusion sensitivity / «Saliency maps»

Анализ отдельных нейронов / каналов / слоёв

Нейроискусство

Генерация изображений

Генерация текстур

Генерация пейзажей

Стилизация (перенос стиля)

Быстрая стилизация

Зачем наблюдать?

- как и почему NN работает
- наблюдать преобразования признаковых пространств
- помогает изобретать новые подходы
- помогает видеть проблемы в данных / моделях
- помогает улучшать модель
ex: фильтры 1 уровня должны быть чёткие, безызбыточные и т.п.

За чем можно наблюдать в NN?

- **параметры (ex: фильтры как картинки)**
- **внутренние активации – как картинки**
- **распределения активаций (на отдельных объектах)**
- **производные по входу**
- **входы, максимизирующие какой-то ответ**
- **визуализация в промежуточных признаковых пространствах
(любые средства стандартного ML)**
- **«adversarial samples» / обманные изображения (Fooling Images)**
будет отдельная тема

Визуализация весов: свёртки первого слоя

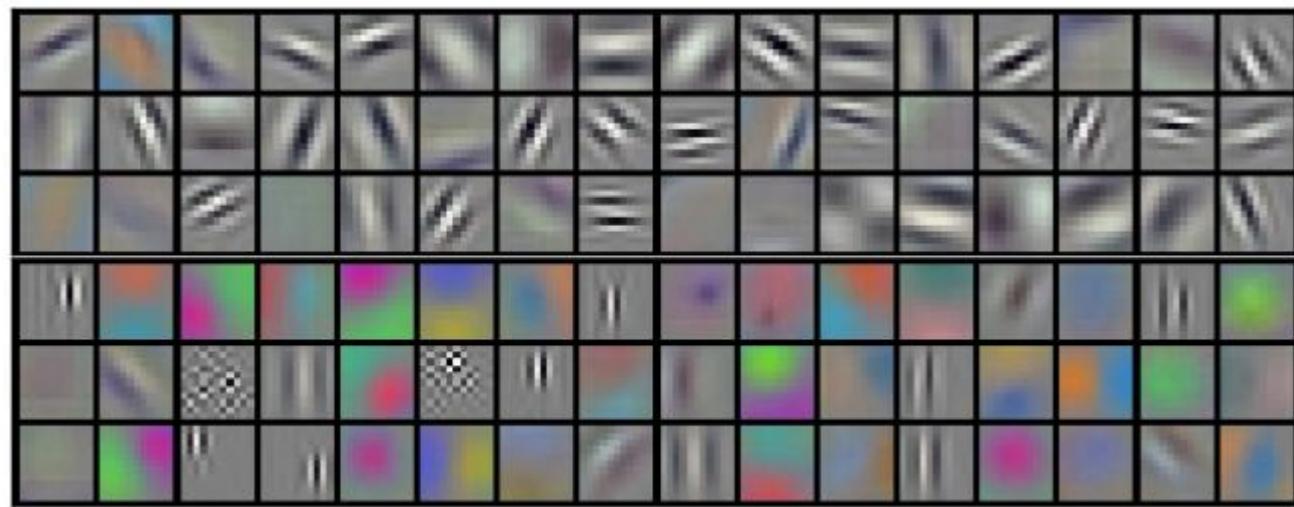
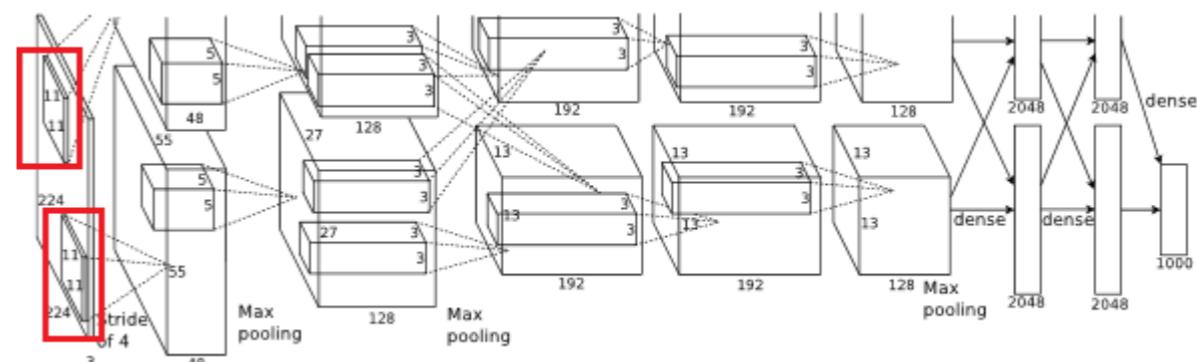


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

Смотрим на свёртки (фильтры) как на картинки



**для первых слоёв понятная интерпретация,
для других – делают по-другому**

- каналов > 3
- нет смысла «паттерн пикселей»

Krizhevsky A. et al. «ImageNet Classification with Deep Convolutional Neural Networks»

Визуализация весов / нейронов промежуточных слоёв: «deconvnet»

**Идея: как-то пропустить сигнал обратно и понять,
на какие картинки (паттерны) «заточены» свёртки
deconvnets (deconvolutional network) – как бы обратная сеть
тензоры → изображения**

изначально придуманы в USL, «switches» – для запоминания положения максимумов

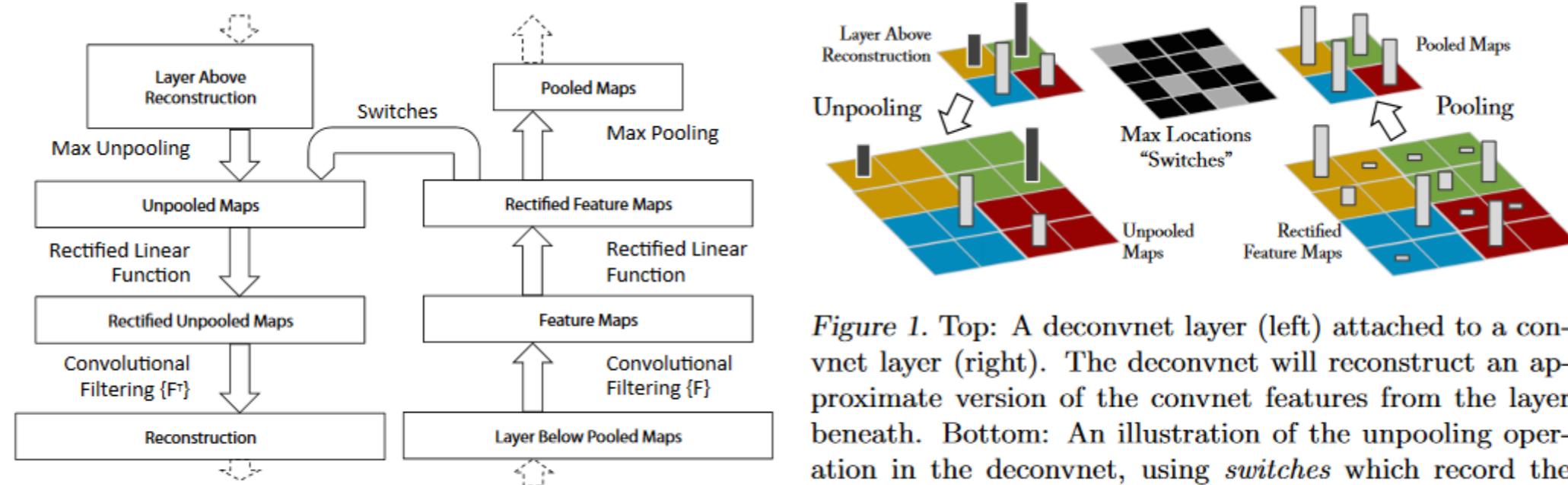
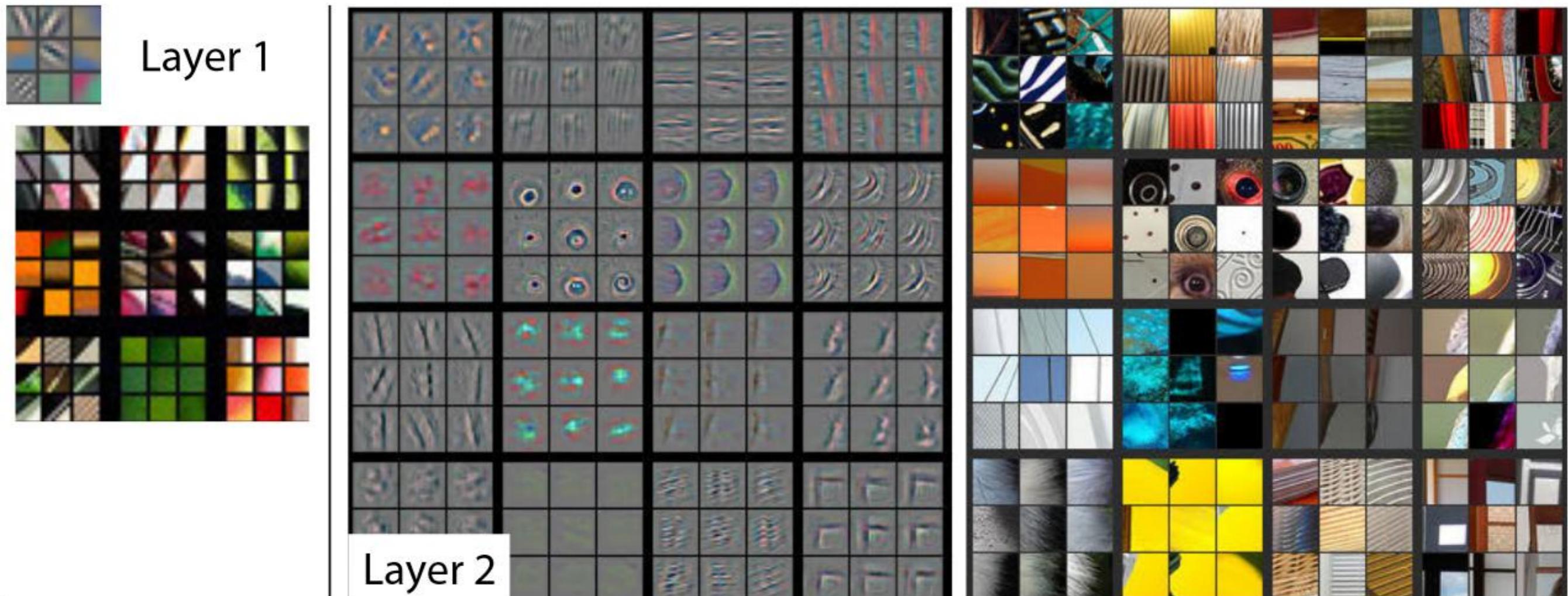


Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

M.D. Zeiler, R. Fergus «Visualizing and understanding convolutional networks» // European conference on computer vision, Springer (2014), pp. 818-833 <https://arxiv.org/pdf/1311.2901.pdf>

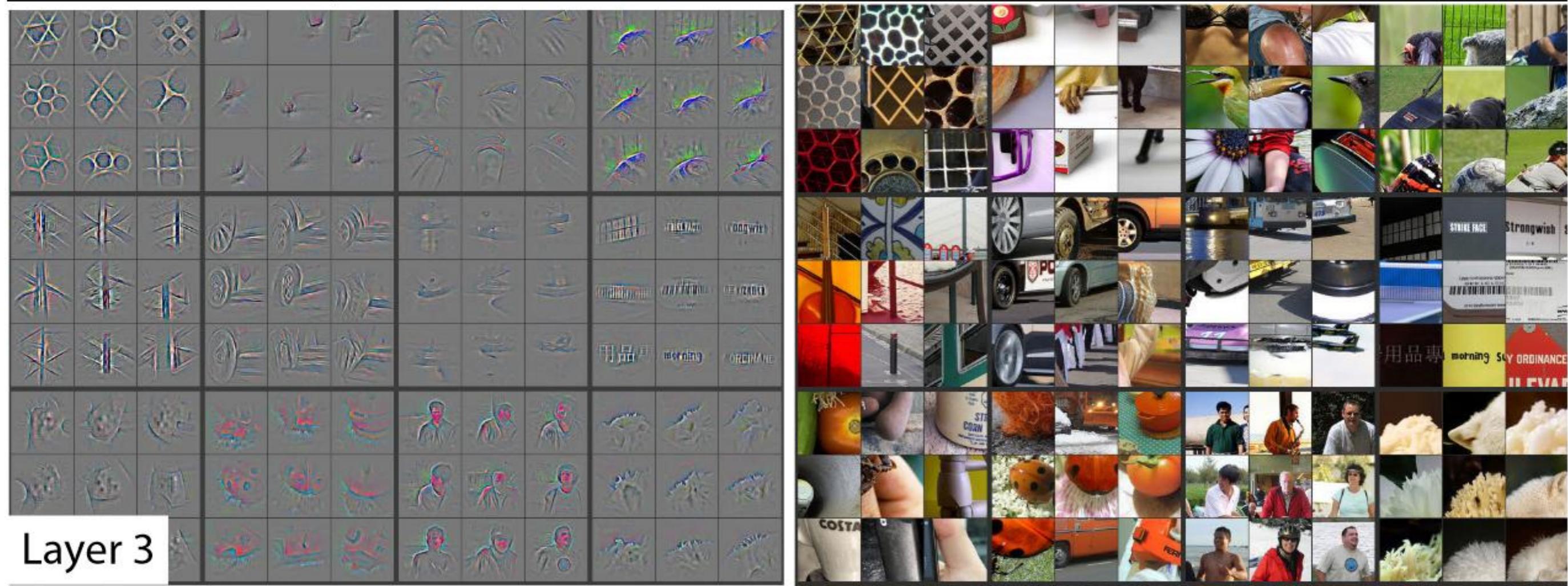
Визуализация весов / нейронов промежуточных слоёв: «deconvnet»

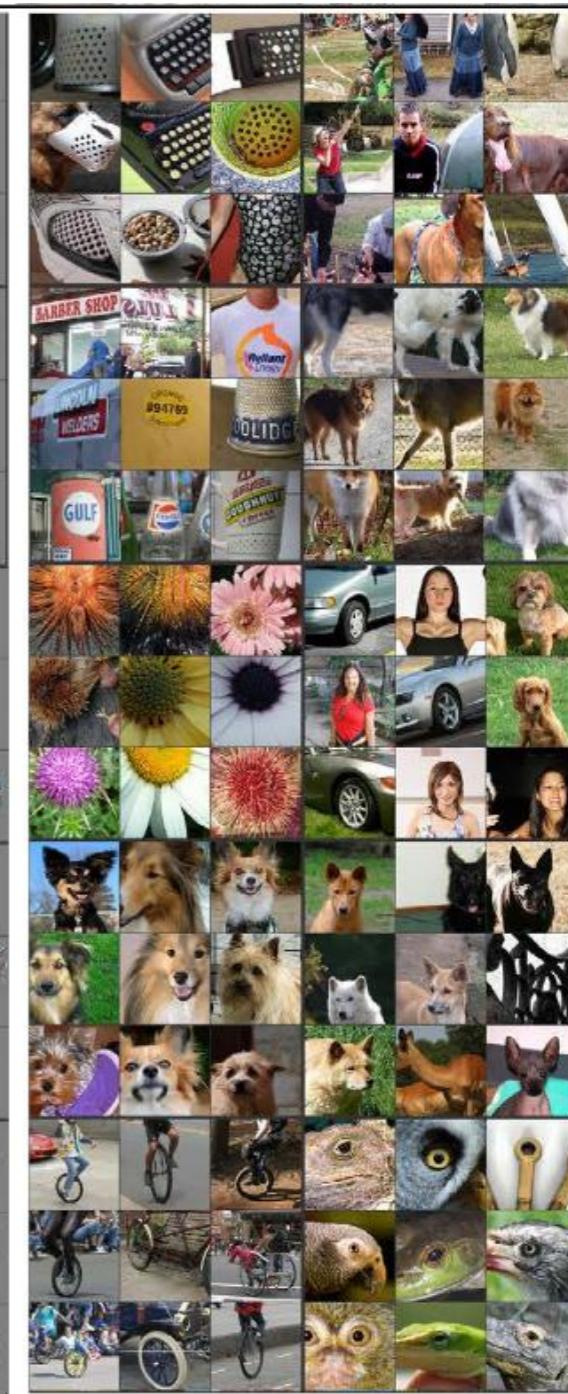
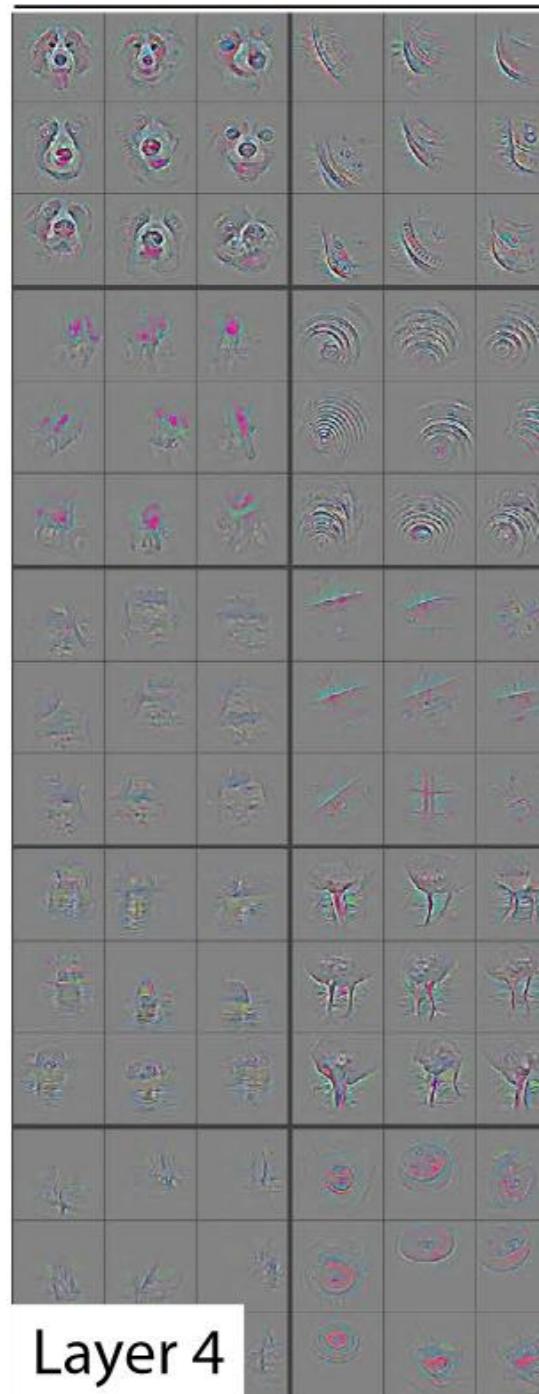
для слоёв 2-5 выбираем нейрон, выбираем 9 картинок,
на которых его активация максимальна, «делаем deconv»



Zeiler, Rob Fergus «Visualizing and Understanding Convolutional Networks» <https://arxiv.org/pdf/1311.2901.pdf>

Визуализация весов / нейронов промежуточных слоёв: «deconvnet»





Визуализация весов / нейронов промежуточных слоёв: «deconvnet»

Как формируются представления...

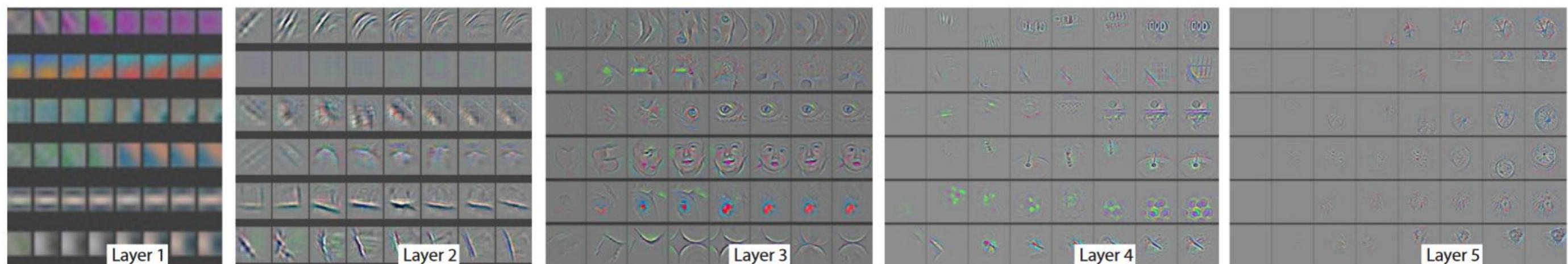


Figure 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to pixel space using our deconvnet approach. Color contrast is artificially enhanced and the figure is best viewed in electronic form.

Class Activation Maps (CAM)

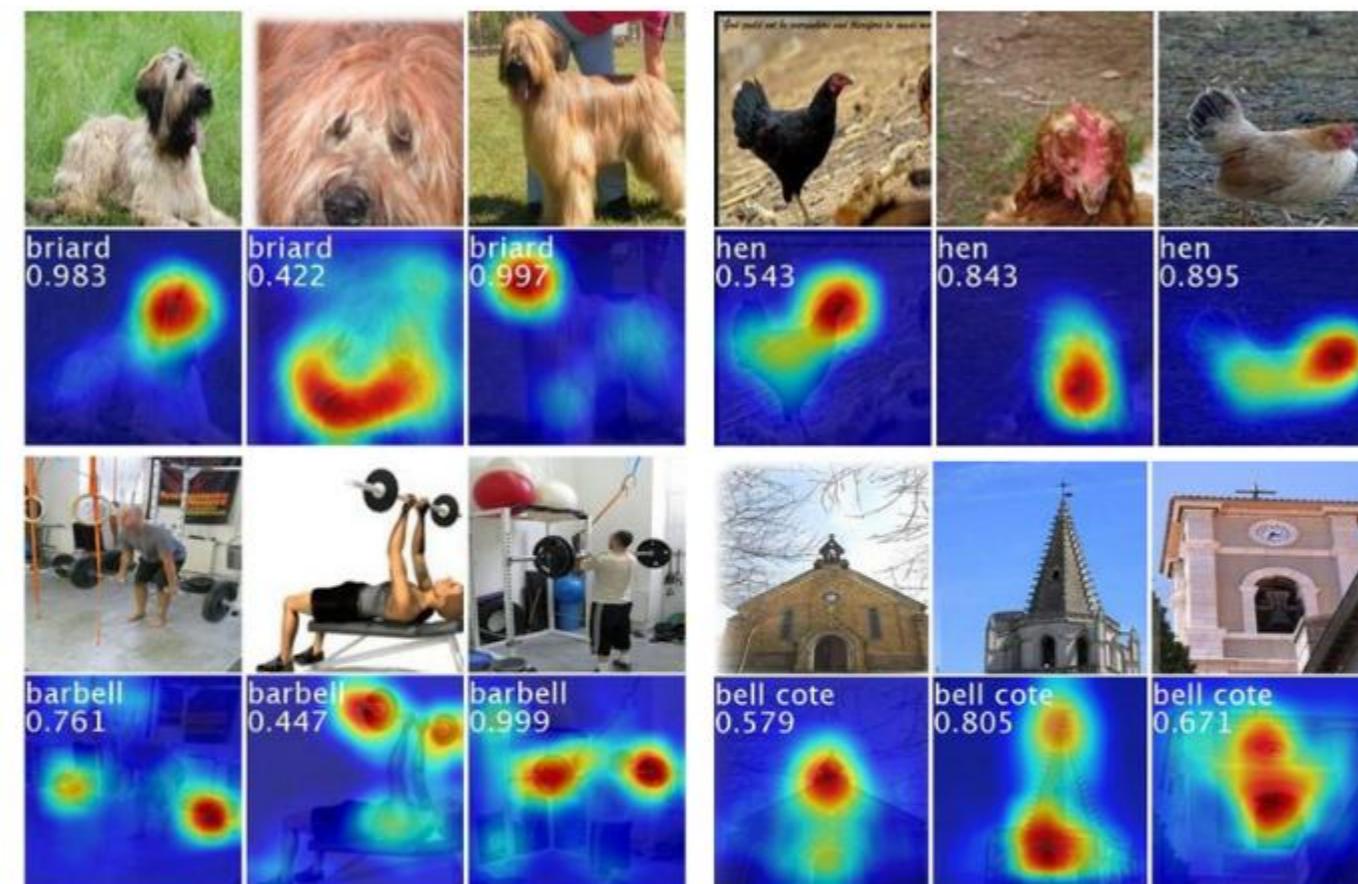


Figure 3. The CAMs of four classes from ILSVRC [20]. The maps highlight the discriminative image regions used for image classification e.g., the head of the animal for *briard* and *hen*, the plates in *barbell*, and the bell in *bell cote*.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba «Learning deep features for discriminative localization» IEEE CVPR (2016)
<https://arxiv.org/pdf/1512.04150.pdf>

Class Activation Maps (CAM)

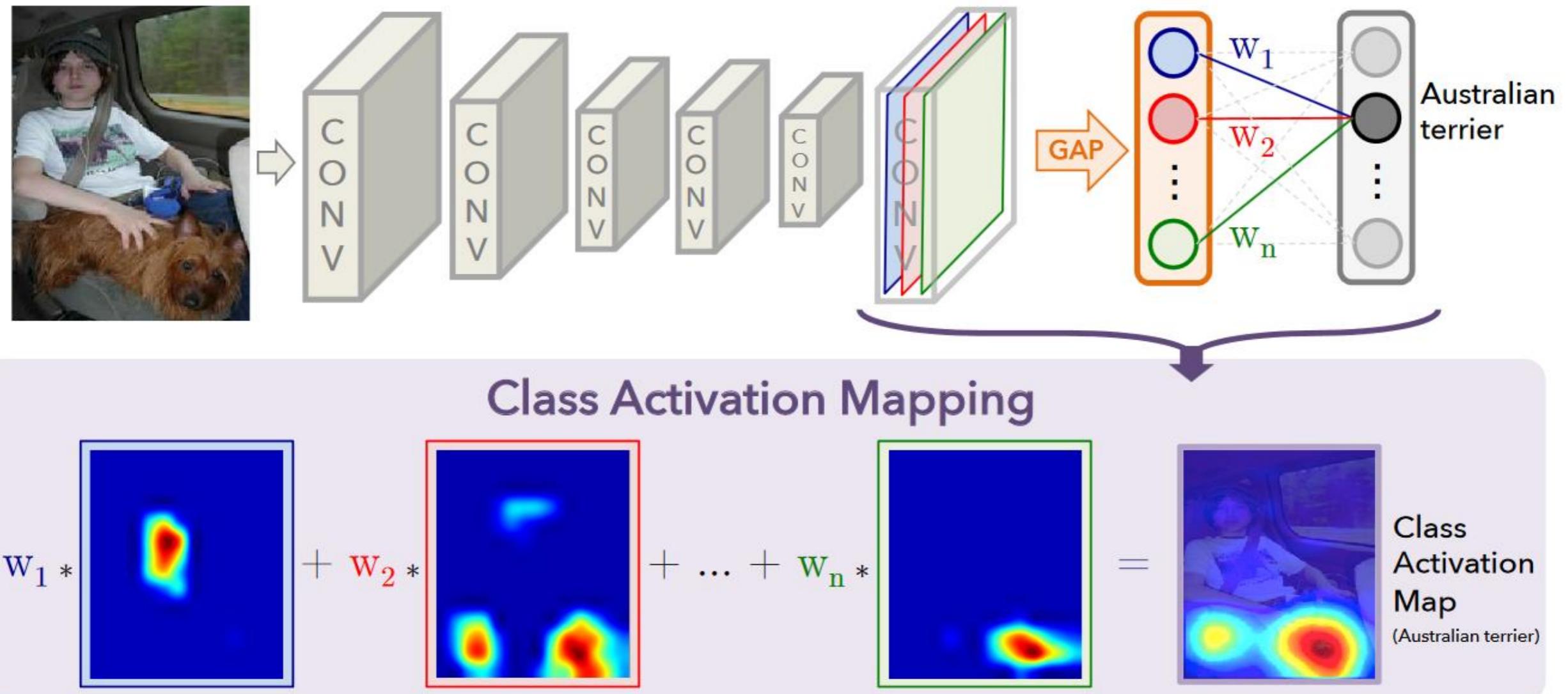


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

для сетей с GAP (Global Average Pooling) – только для них!

Class Activation Maps (CAM)

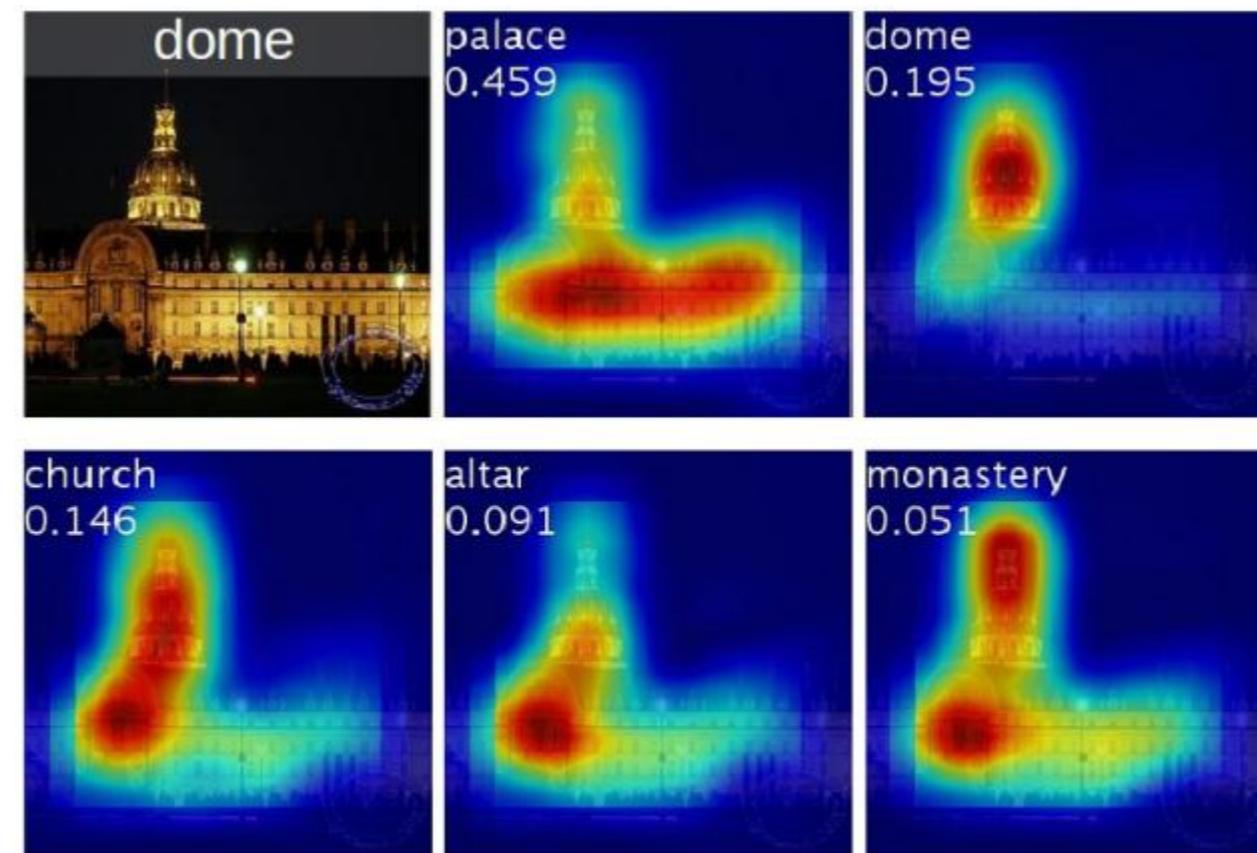


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper round part while *palace* activates the lower flat part of the compound.

Guided Backpropagation

**дальнейшее развитие CAM
 для сетей без max pooling-a**

**т.к. в задаче детектирования объектов оказалось,
что пулинг можно заменить свёртками с большим stride
хотя применялось и к сетям с max-pooling-ом
но нормально работает и без max-switch-ей**

J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, «Striving for simplicity: The all convolutional net», 2014. <https://arxiv.org/pdf/1412.6806.pdf>

Guided Backpropagation: как пропускается сигнал обратно по сети...

уже был «deconvnet»...

сигнал пускаем обратно по сети

часто конкретный нейрон = 1, остальные = 0

если используется pooling – запоминаем, где был максимум
но тогда есть зависимость от конкретного изображения!!!

смотрим на получившееся изображение

интерпретация – что максимально активирует выделенный нейрон

Другой подход – использовать обратное распространение – **The backward pass**
по большому счёту, разница в том, как осуществляется обратный проход через ReLU

«guided backpropagation» – комбинация этих методов, см **след. слайд**
не пропускаются отрицательные градиенты

Guided Backpropagation: Как пропускается сигнал обратно по сети...

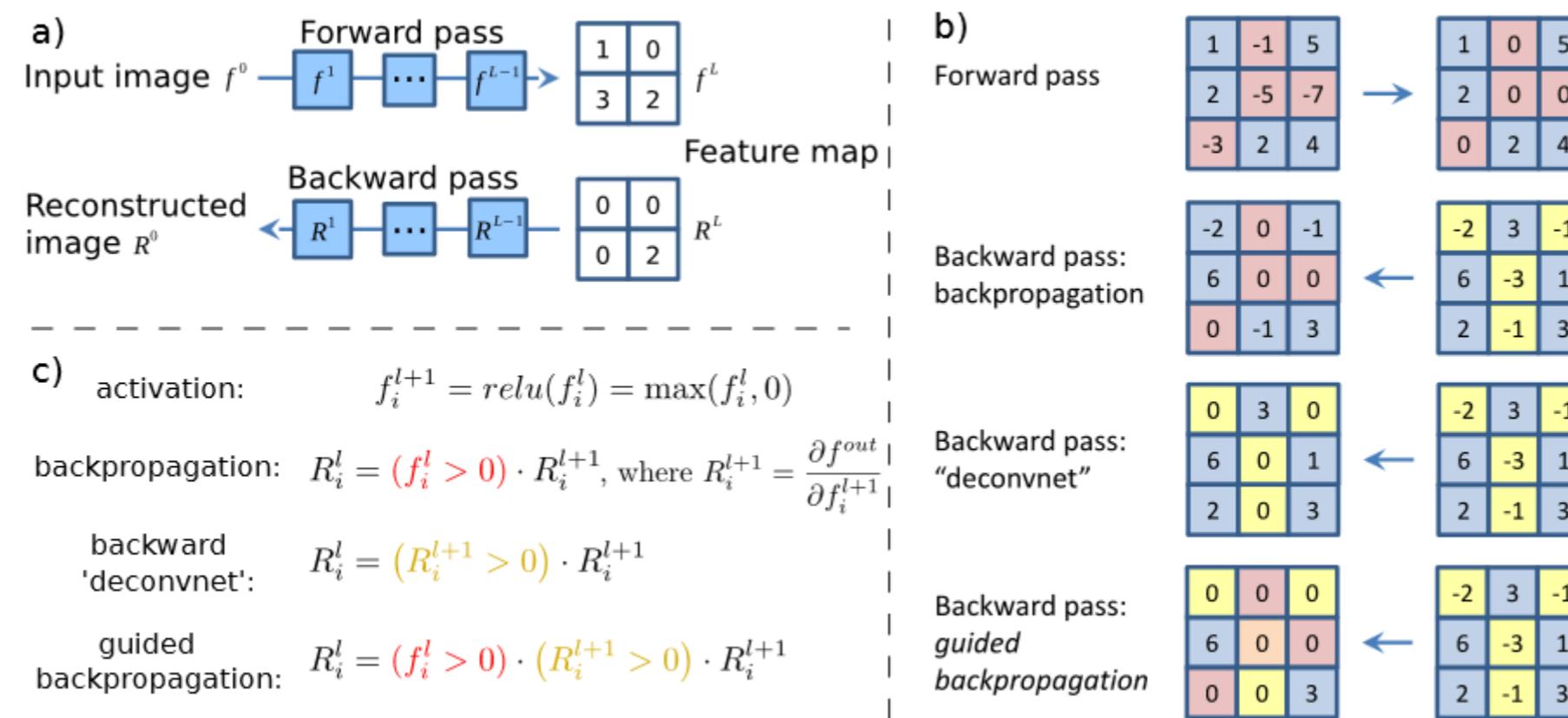
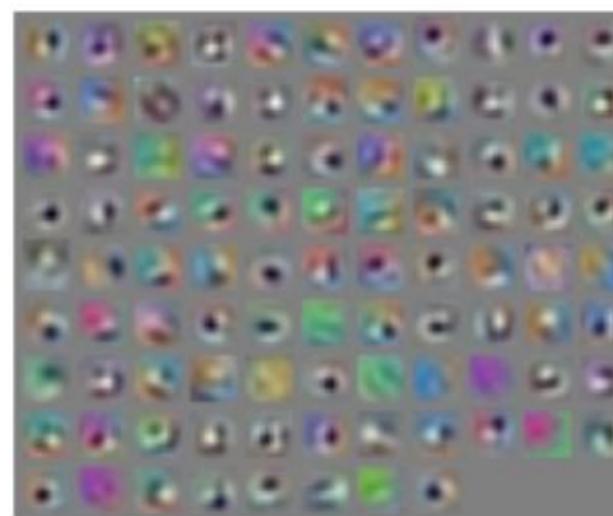


Figure 1: Schematic of visualizing the activations of high layer neurons. a) Given an input image, we perform the forward pass to the layer we are interested in, then set to zero all activations except one and propagate back to the image to get a reconstruction. b) Different methods of propagating back through a ReLU nonlinearity. c) Formal definition of different methods for propagating a output activation *out* back through a ReLU unit in layer l ; note that the 'deconvnet' approach and guided backpropagation do not compute a true gradient but rather an imputed version.

Guided Backpropagation: веса первых слоёв

conv1



conv2



conv3

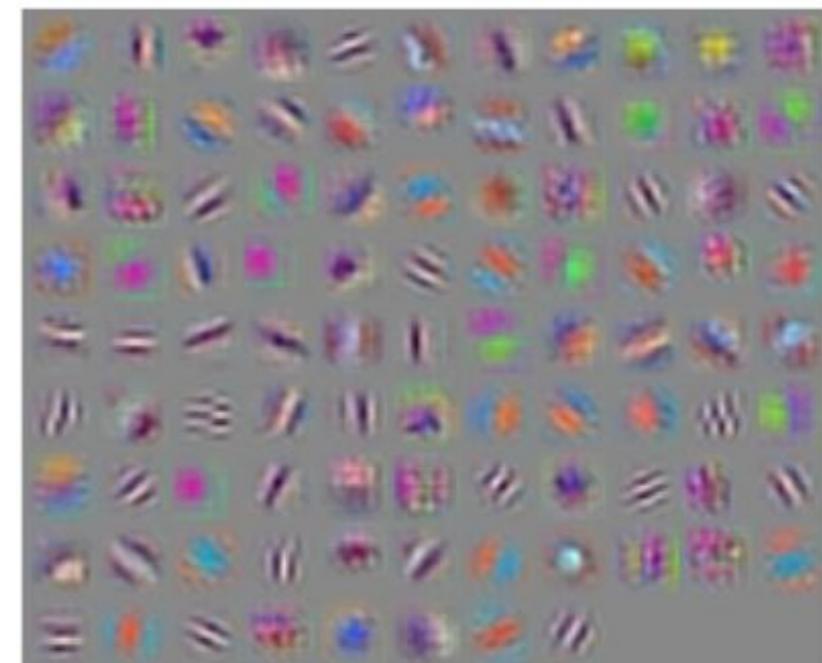


Figure 2: Visualizations of patterns learned by the lower layers (conv1-conv3) of the network trained on ImageNet. Each single patch corresponds to one filter. Interestingly, Gabor filters only appear in the third layer.

если недостаточно чёткие картинки – мало учили!

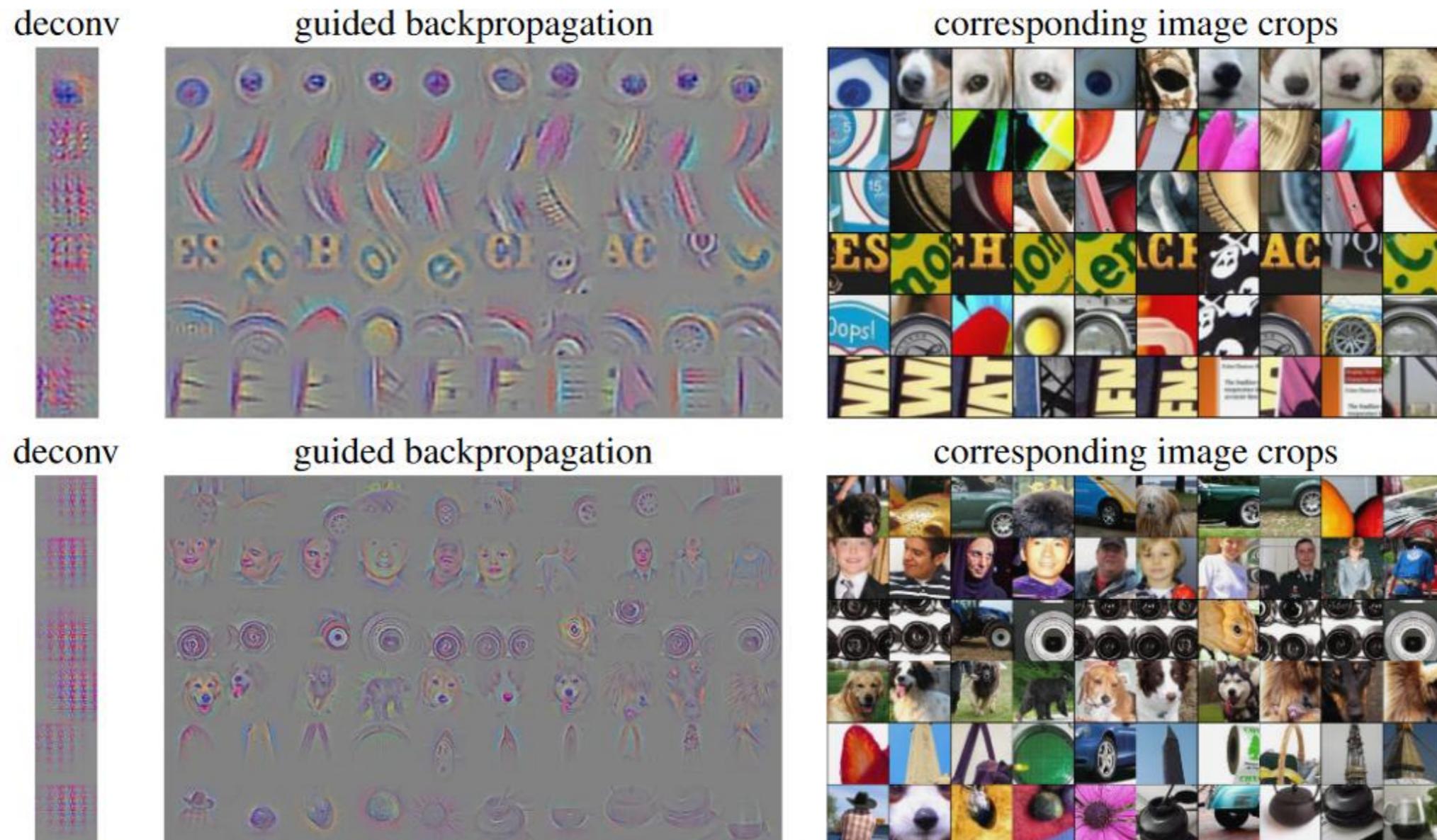


Figure 3: Visualization of patterns learned by the layer conv6 (top) and layer conv9 (bottom) of the network trained on ImageNet. Each row corresponds to one filter. The visualization using “guided backpropagation” is based on the top 10 image patches activating this filter taken from the ImageNet dataset. Note that image sizes are not preserved (in order to save space).

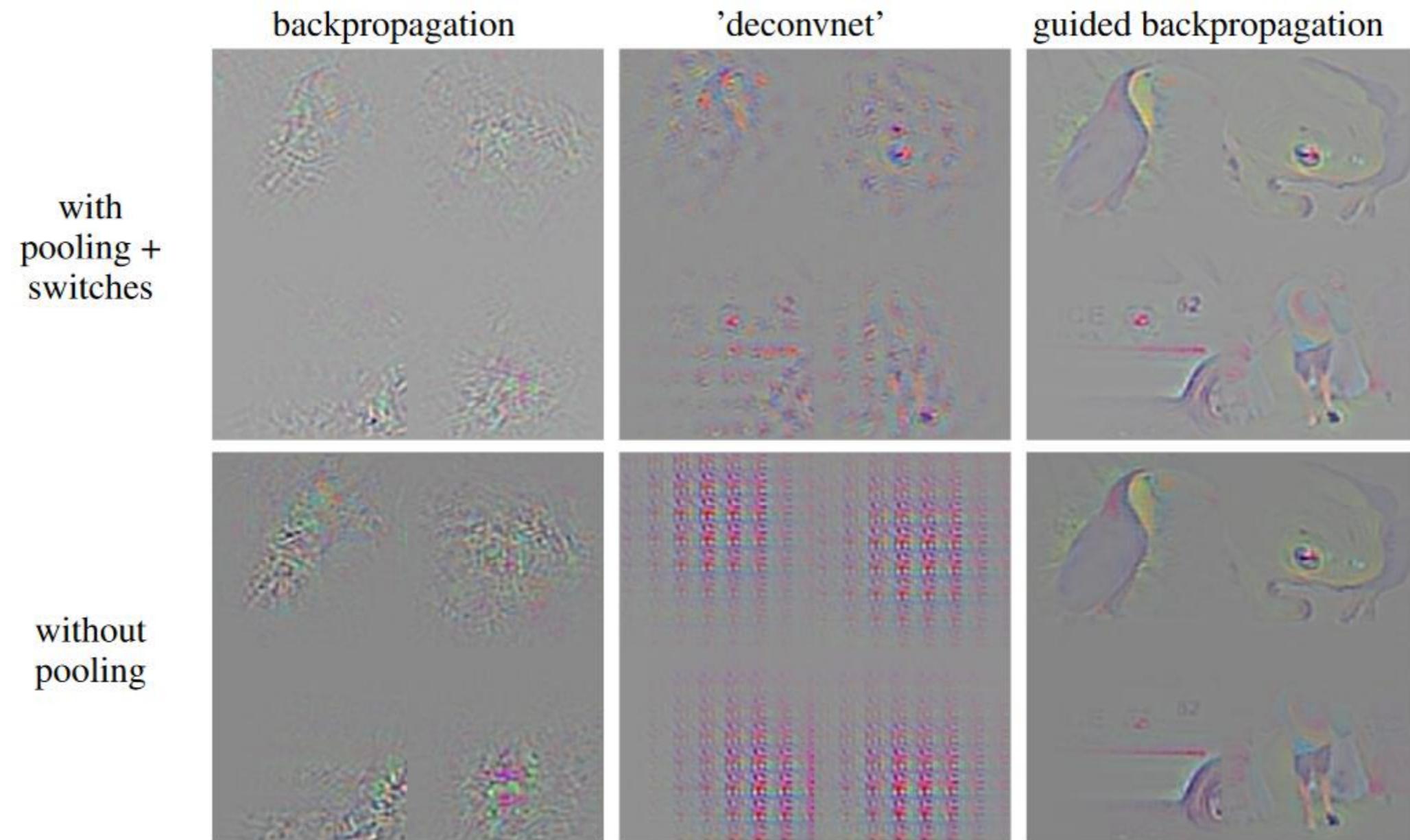
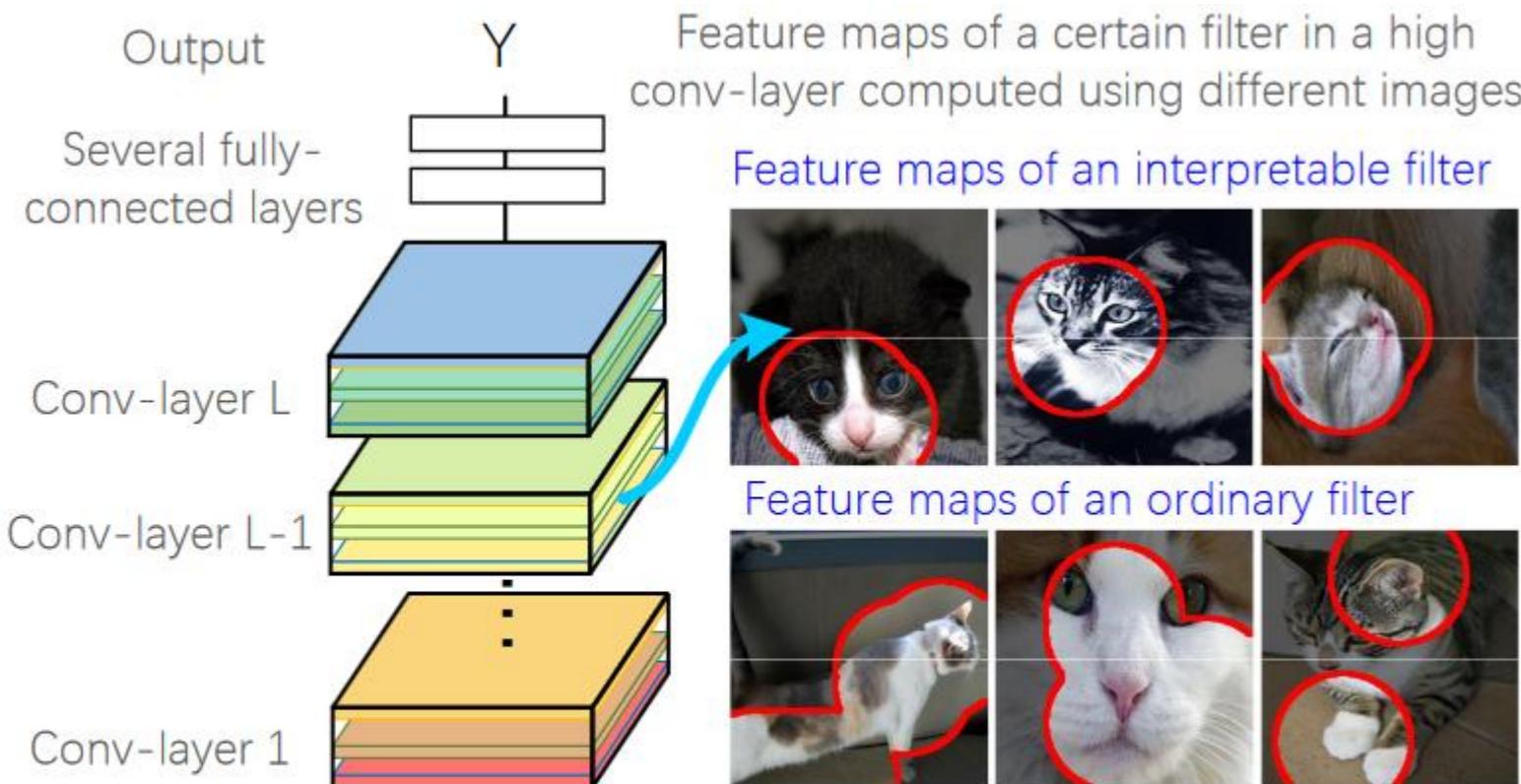


Figure 4: Visualization of descriptive image regions with different methods from the single largest activation in the last convolutional layer conv12 of the network trained on ImageNet. Reconstructions for 4 different images are shown.

Interpretable Convolutional Neural Networks



Идея – фильтр отвечает за «конкретное место» на изображении

Figure 1. Comparison of a filter's feature maps in an interpretable CNN and those in a traditional CNN.

Q. Zhang, Y. Nian Wu, S.-C. Zhu Interpretable convolutional neural networks // CVPR, 2018
<https://arxiv.org/pdf/1710.00935.pdf>

Interpretable Convolutional Neural Networks

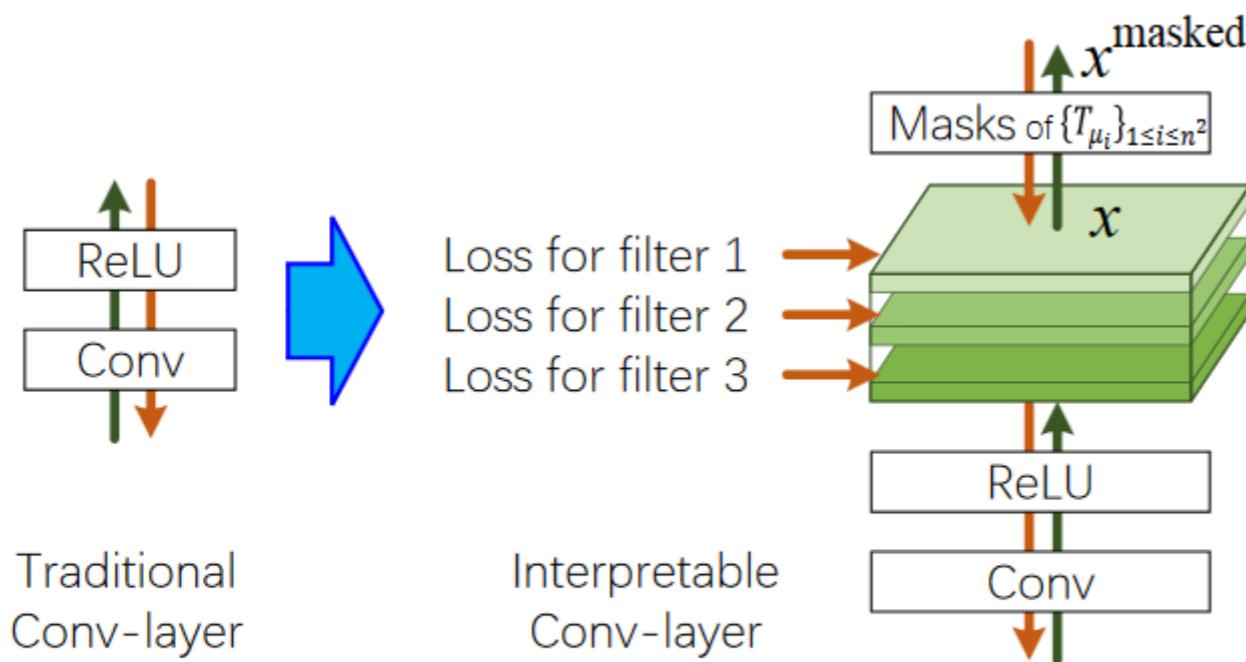


Figure 2. Structures of an ordinary conv-layer and an interpretable conv-layer. Green and red lines indicate the forward and backward propagations, respectively.

для интерпретации
добавляем **Loss** к каждому каналу

чтобы он получал представление части
объекта

**активировался в каком-то месте для
конкретного класса**

Interpretable Convolutional Neural Networks

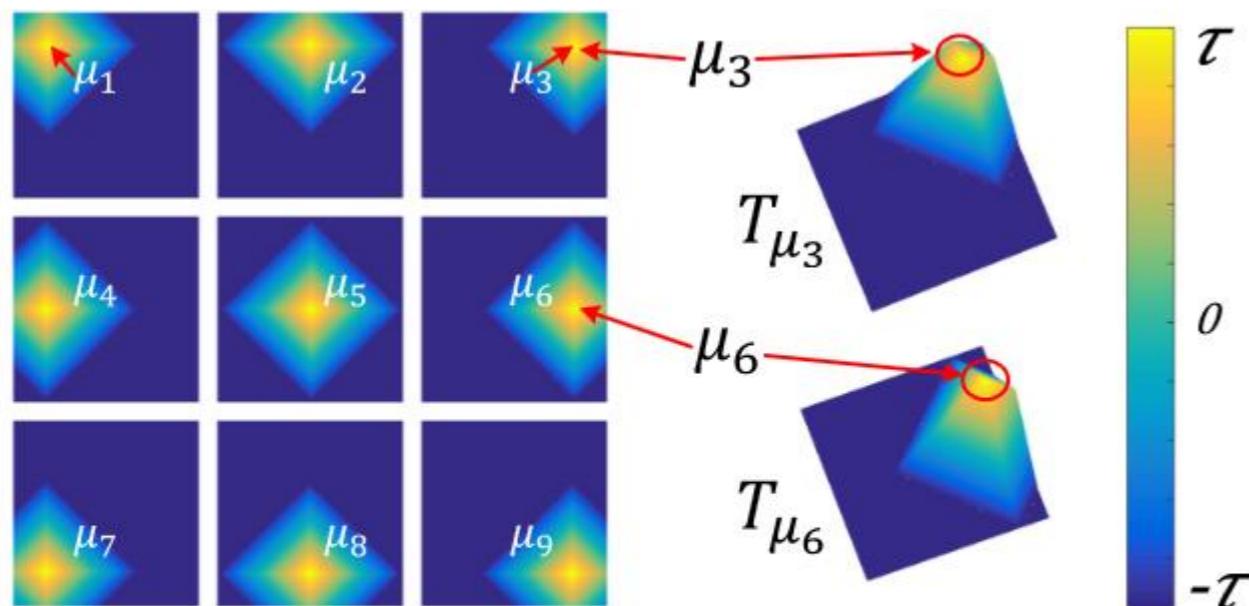


Figure 3. Templates of T_{μ_i} . In fact, the algorithm also supports a round template based on the L-2 norm distance. Here, we use the L-1 norm distance instead to speed up the computation.

Для матрицы $n \times n$ делаем n^2 масок, каждая также является матрицей $n \times n$

- Просто идея:**
- На прямом проходе находим в канале max элемент и маскируем соотв. маской
 - На обратном принуждаем относить к одному классу (опускаем как)

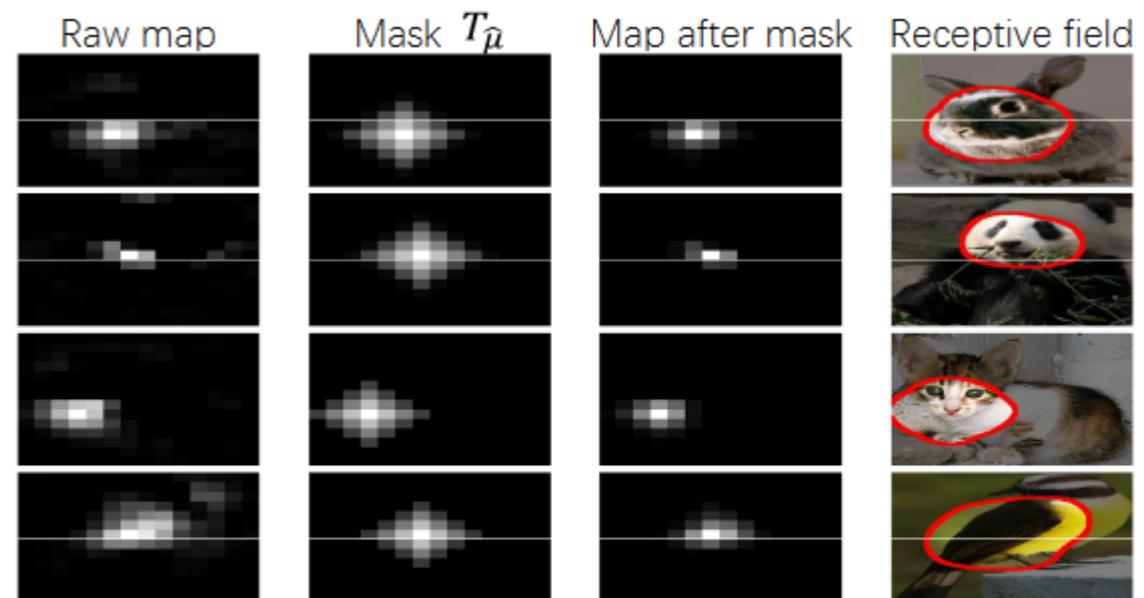


Figure 4. Given an input image I , from the left to the right, we consequently show the feature map of a filter after the ReLU layer x , the assigned mask \hat{T}_{μ} , the masked feature map x^{masked} , and the image-resolution RF of activations in x^{masked} computed by [38].

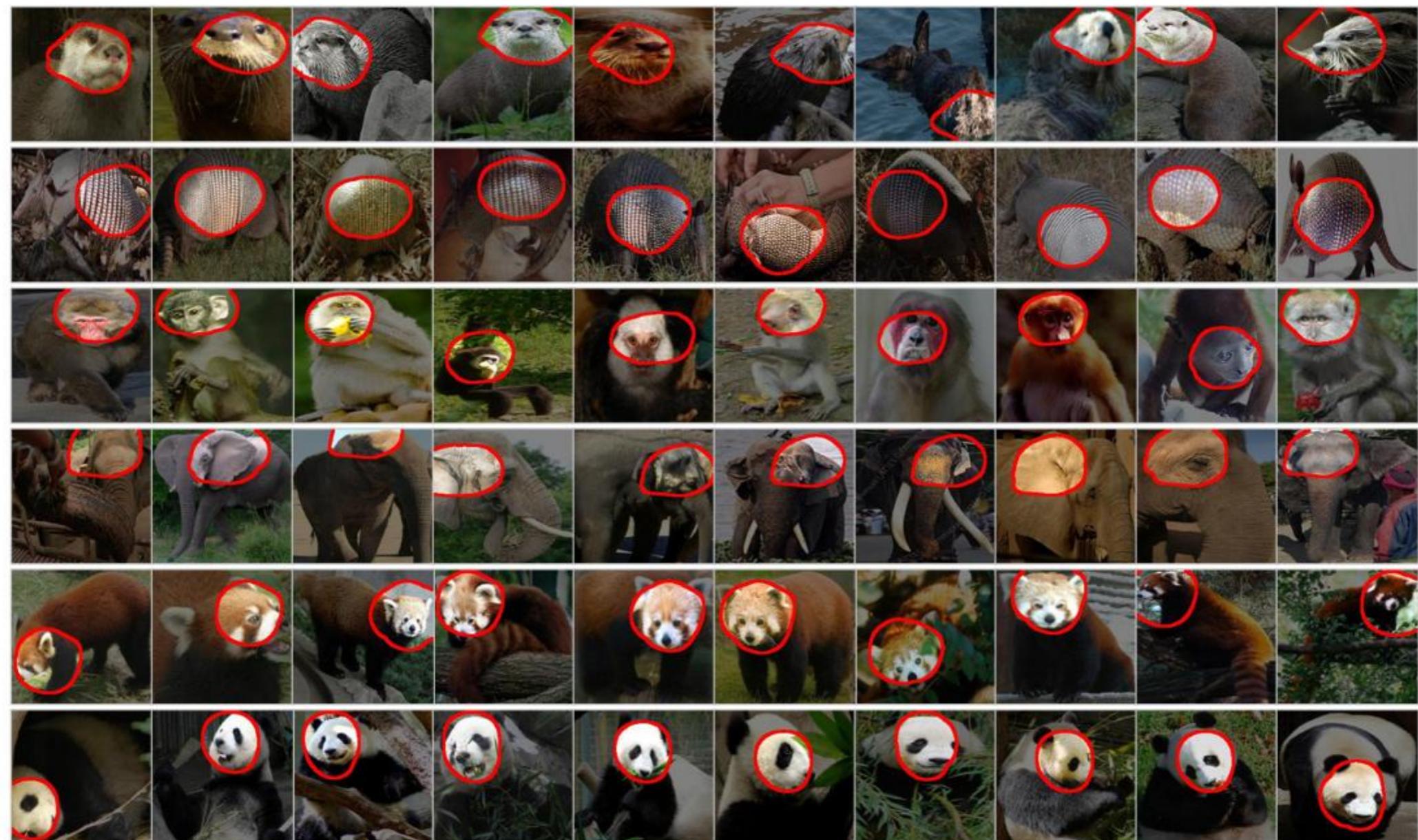


Figure 8. Visualization of filters in the top interpretable conv-layer. Each row corresponds to feature maps of a filter in a CNN that is learned to classify a certain category.

Grad-CAM / Guided Grad-CAM

для широкого класса свёрточных сетей

Guided Propagation + ~CAM

GP – чётко что-то подсвечивает, но не всегда

это только объект нужного класса

т.е. визуализация не разделяет классы!

**(1) позволяет ввести понятие «важности»
нейрона (по значению α) – будет
использовано потом**

we first compute the gradient of the score for class c , y^c (before the softmax), with respect to feature map activations A^k of a convolutional layer, i.e. $\frac{\partial y^c}{\partial A^k}$. These gradients flowing back are global-average-pooled over the width and height dimensions (indexed by i and j respectively) to obtain the neuron importance weights α_k^c :

$$\alpha_k^c = \underbrace{\frac{1}{Z} \sum_i \sum_j}_{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \quad (1)$$

We perform a weighted combination of forward activation maps, and follow it by a ReLU to obtain,

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right) \quad (2)$$

R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra

«Grad-cam: Visual explanations from deep networks via gradient-based localization» // IEEE ICCV (2017) <https://arxiv.org/pdf/1610.02391.pdf>

Grad-CAM / Guided Grad-CAM

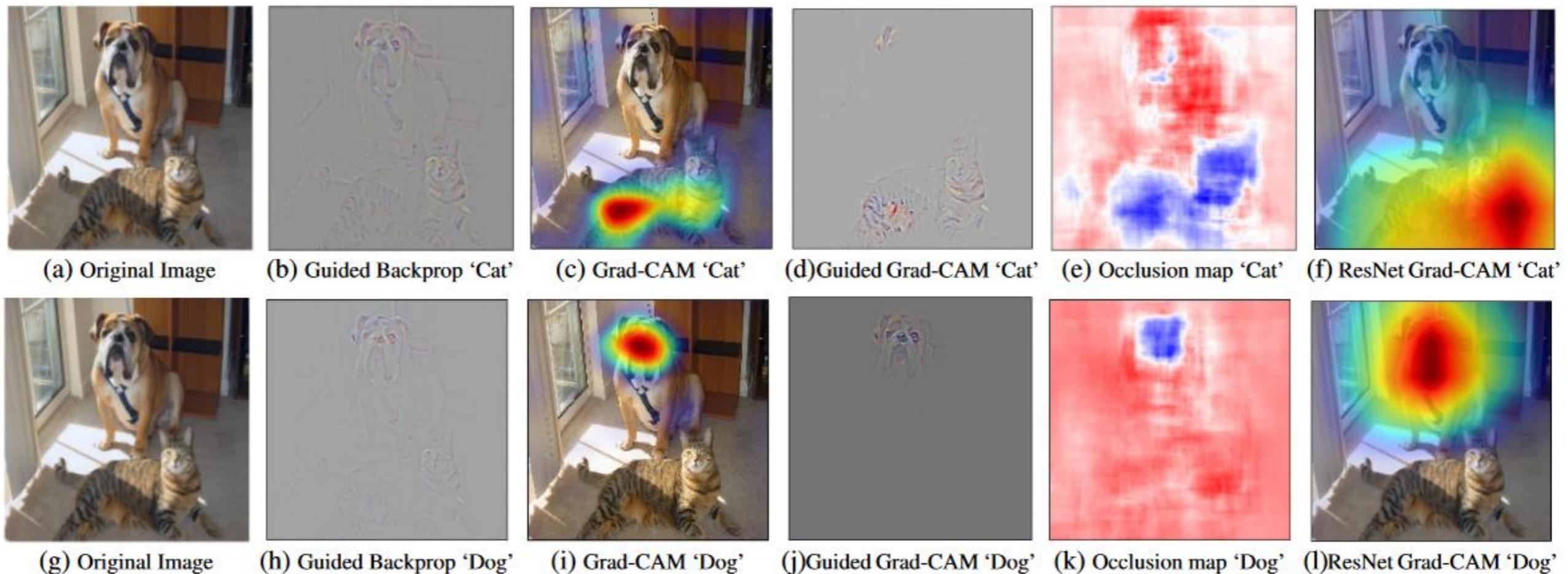


Fig. 1: (a) Original image with a cat and a dog. (b-f) Support for the cat category according to various visualizations for VGG-16 and ResNet. (b) Guided Backpropagation [53]: highlights all contributing features. (c, f) Grad-CAM (Ours): localizes class-discriminative regions, (d) Combining (b) and (c) gives Guided Grad-CAM, which gives high-resolution class-discriminative visualizations. Interestingly, the localizations achieved by our Grad-CAM technique, (c) are very similar to results from occlusion sensitivity (e), while being orders of magnitude cheaper to compute. (f, l) are Grad-CAM visualizations for ResNet-18 layer. Note that in (c, f, i, l), red regions corresponds to high score for class, while in (e, k), blue corresponds to evidence for the class. Figure best viewed in color.

Grad-CAM / Guided Grad-CAM

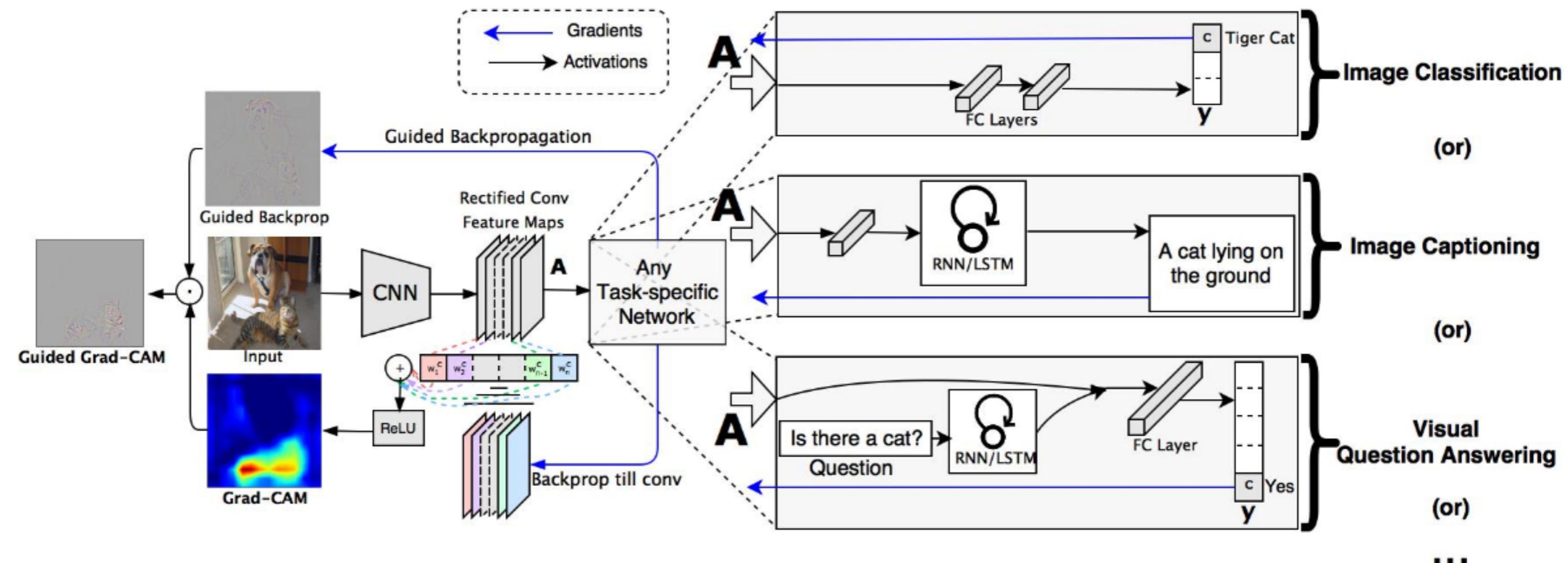


Fig. 2: Grad-CAM overview: Given an image and a class of interest (e.g., ‘tiger cat’ or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Grad-CAM / Guided Grad-CAM



(a) Raw input image. Note that this is not a part of the tasks (b) and (c)

What do you see?

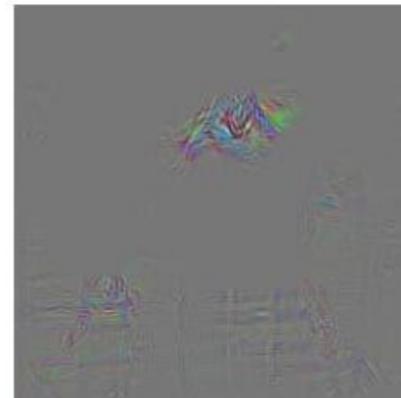


Your options:

- Horse
- Person

(b) AMT interface for evaluating the class-discriminative property

Both robots predicted: Person
Robot A based it's decision on **Robot B** based it's decision on



Which robot is more reasonable?

- Robot A** seems clearly more reasonable than **robot B**
- Robot A** seems slightly more reasonable than **robot B**
- Both robots seem equally reasonable
- Robot B** seems slightly more reasonable than **robot A**
- Robot B** seems clearly more reasonable than **robot A**

(c) AMT interface for evaluating if our visualizations instill trust in an end user

Fig. 5: AMT interfaces for evaluating different visualizations for class discrimination (b) and trustworthiness (c). Guided Grad-CAM outperforms baseline approaches (Guided-backprop and Deconvolution) showing that our visualizations are more class-discriminative and help humans place trust in a more accurate classifier.

обоснование подхода с помощью ассесоров

Задача «текстового объяснения» (имена нейронам даны методом из предыдущей статьи)

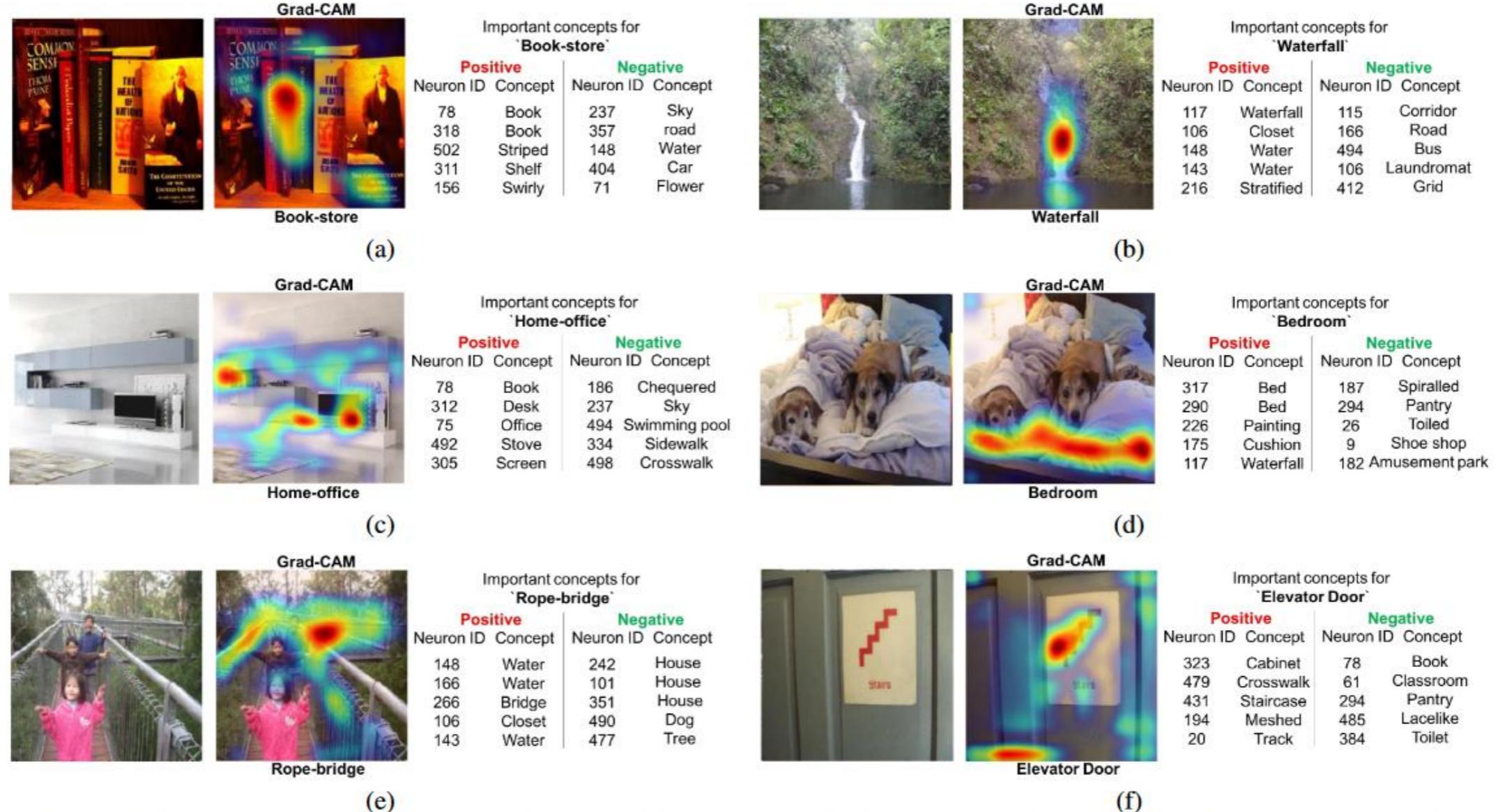


Fig. 9: Examples showing visual explanations and textual explanations for VGG-16 trained on Places365 dataset [61]. For textual explanations we provide the most important neurons for the predicted class along with their names. Important neurons can be either be persuasive (positive importance) or inhibitive (negative importance). The first 2 rows show success cases, and the last row shows 2 failure cases. We see that in (a), the important neurons computed by (1) look for concepts such as book and shelf which are indicative of class 'Book-store' which is fairly intuitive.

Восстановление входа по представлению

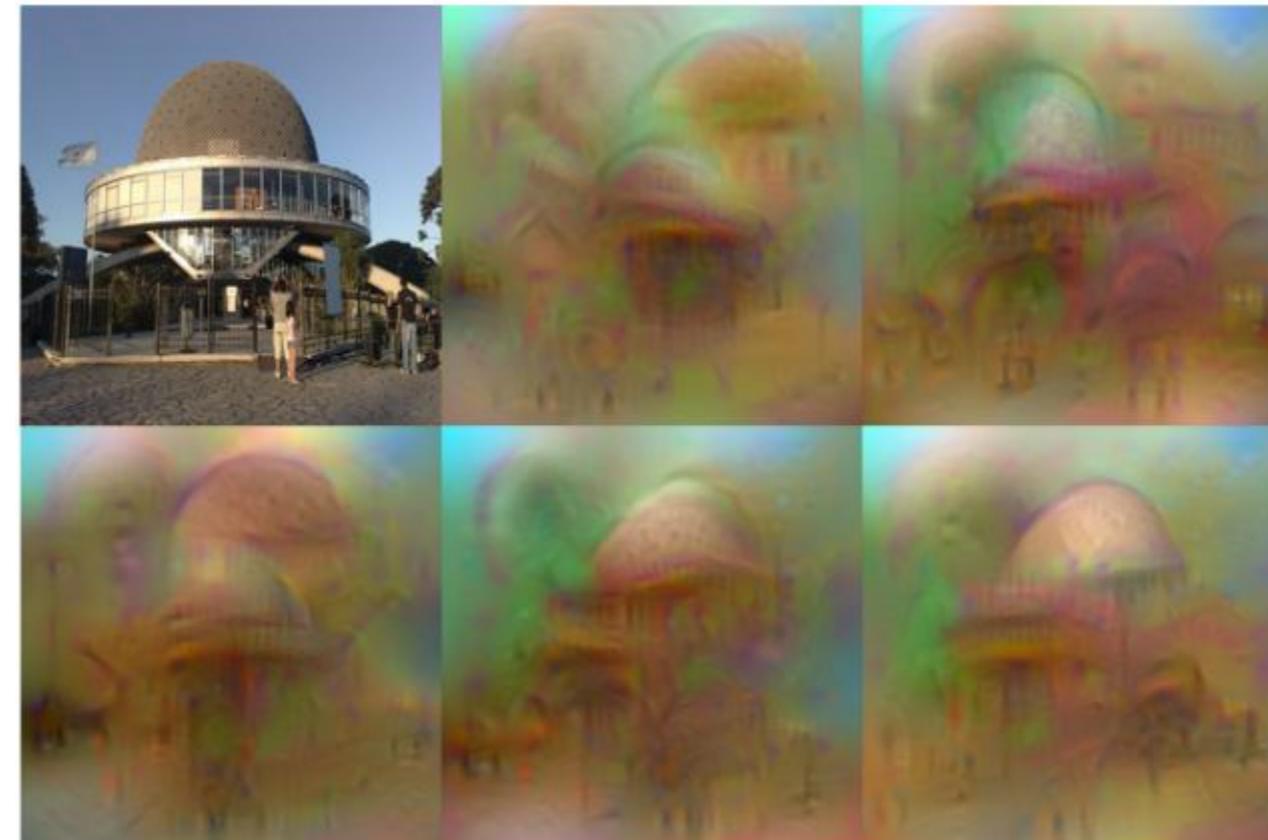


Figure 1. **What is encoded by a CNN?** The figure shows five possible reconstructions of the reference image obtained from the 1,000-dimensional code extracted at the penultimate layer of a reference CNN[13] (before the softmax is applied) trained on the ImageNet data. From the viewpoint of the model, all these images are practically equivalent. This image is best viewed in color/screen.

Aravindh Mahendran, Andrea Vedaldi «Understanding Deep Image Representations by Inverting Them»

<https://arxiv.org/abs/1412.0035>

Восстановление входа по представлению

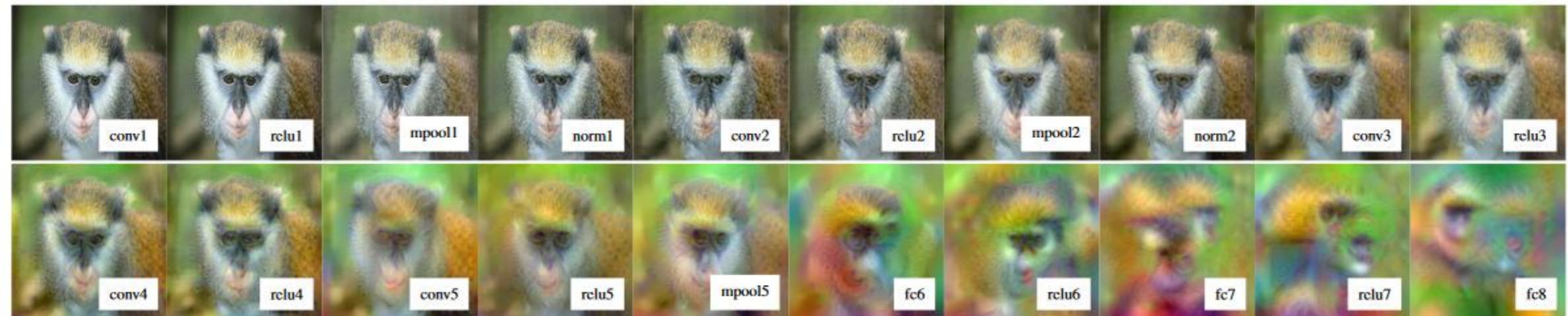


Figure 6. **CNN reconstruction.** Reconstruction of the image of Fig. 5.a from each layer of CNN-A. To generate these results, the regularization coefficient for each layer is chosen to match the highlighted rows in table 3. This figure is best viewed in color/screen.

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

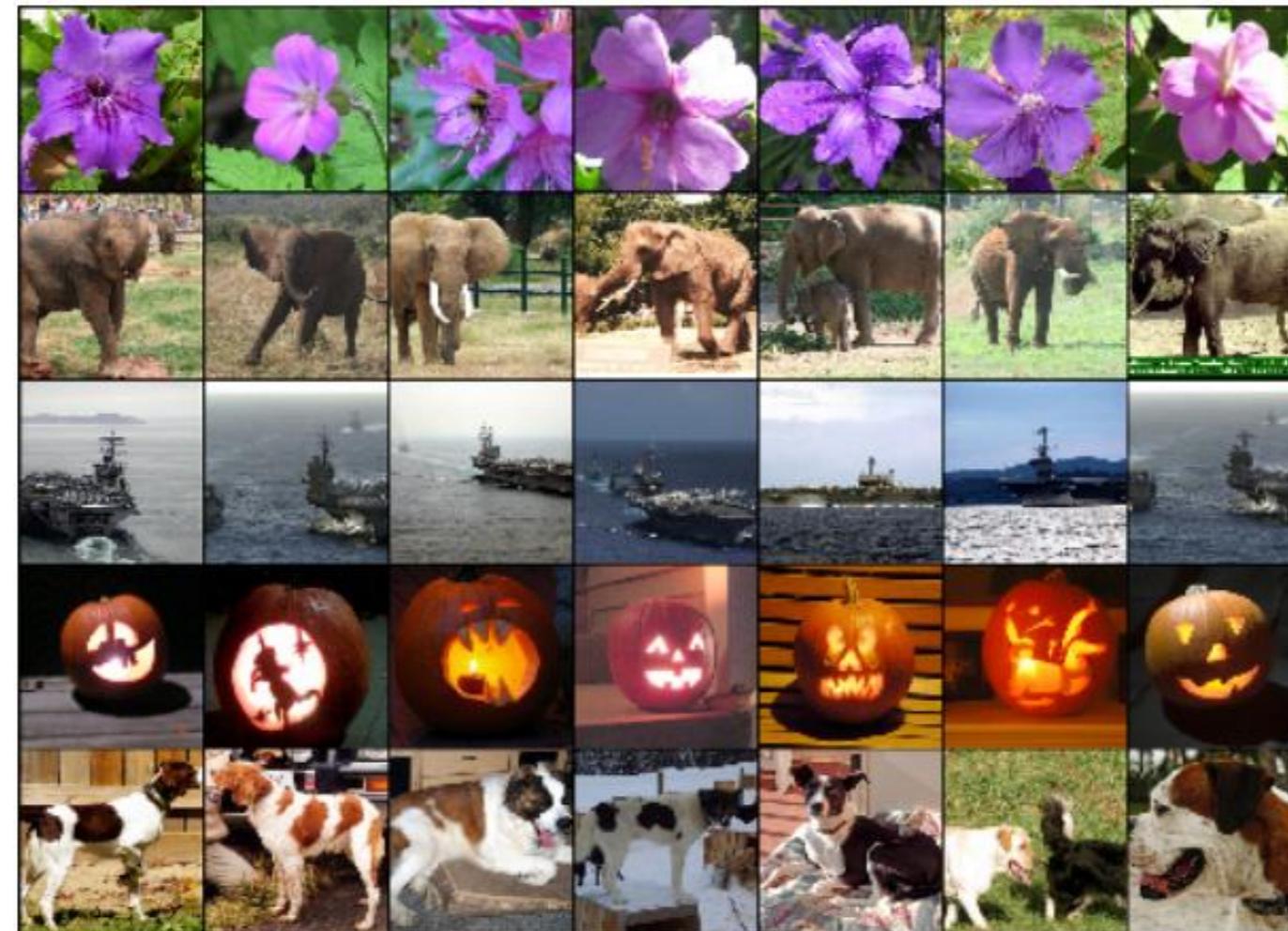
$$\begin{aligned} \Phi : \mathbb{R}^{H \times W \times C} &\rightarrow \mathbb{R}^d \\ \mathbf{x} &\in \mathbb{R}^{H \times W \times C} \end{aligned}$$

есть тонкости с регуляризацией
также проводились эксперименты с классическими представлениями (**HOG, SIFT**)

Стандартные средства в признаковых пространствах

Последний полносвязный слой

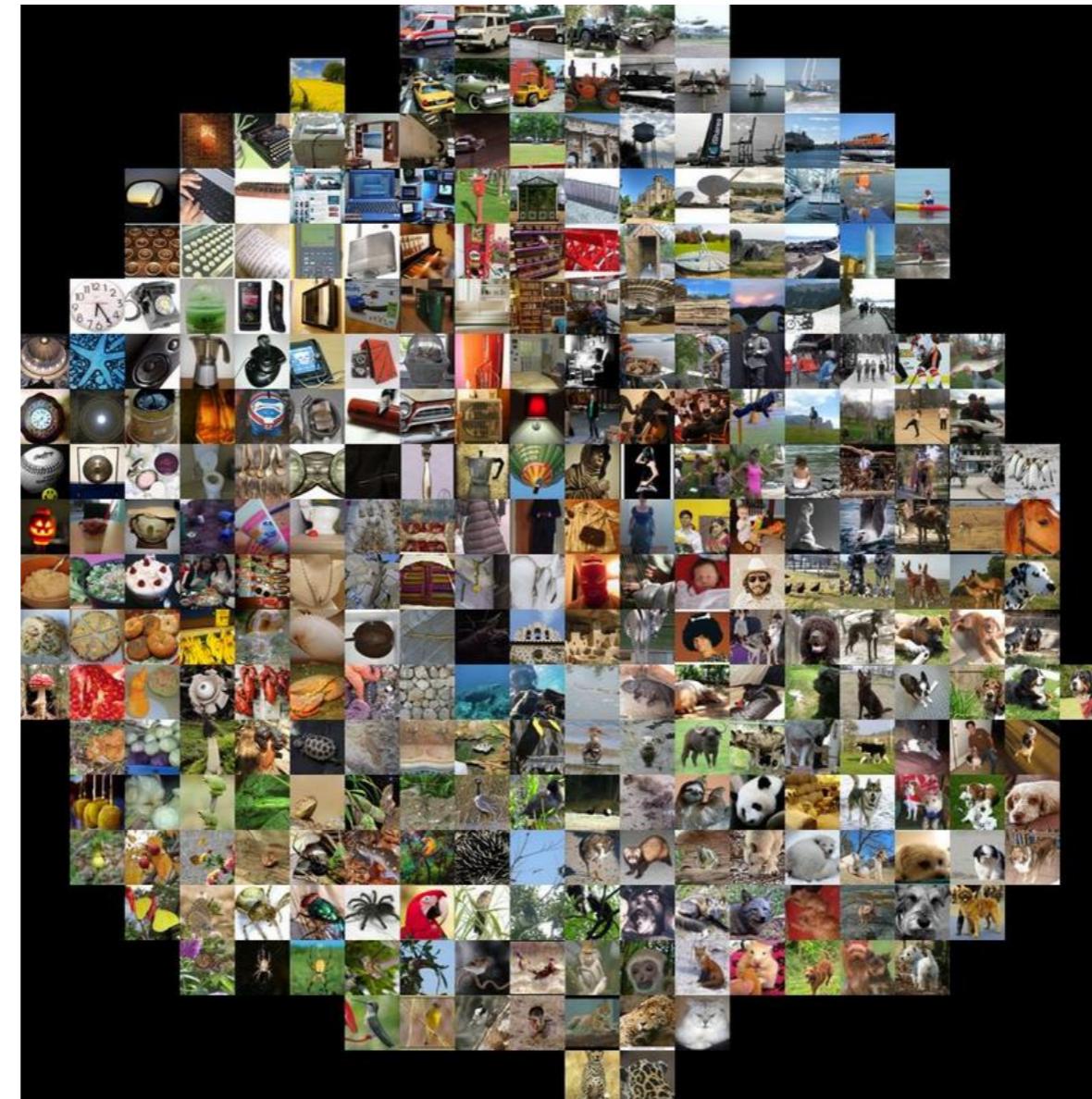
Можно смотреть соседей в этом признаковом пространстве



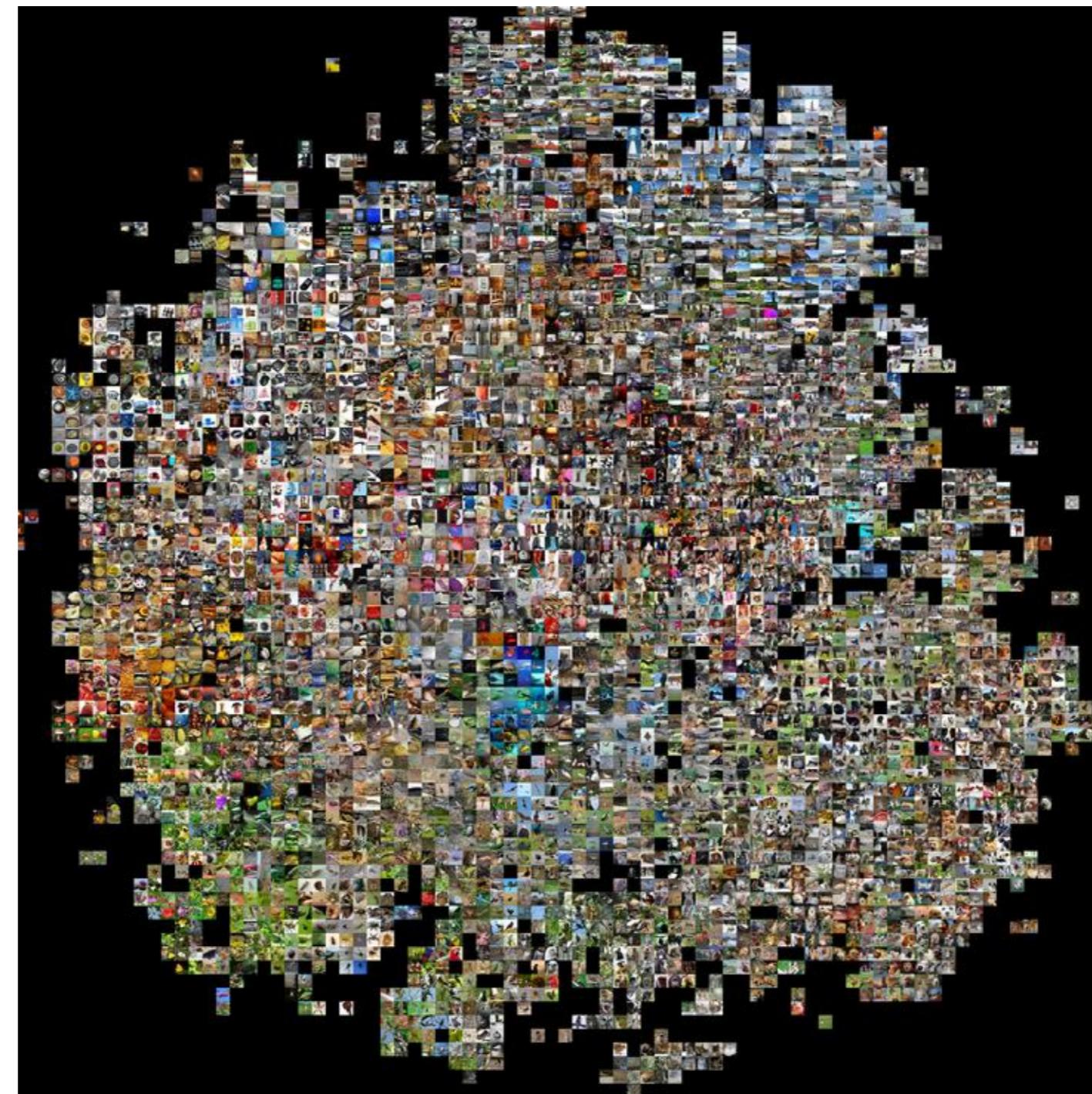
<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

Стандартные средства в признаковых пространствах

Последний полносвязный слой – применить уменьшение размерности... t-SNE в \mathbb{R}^2



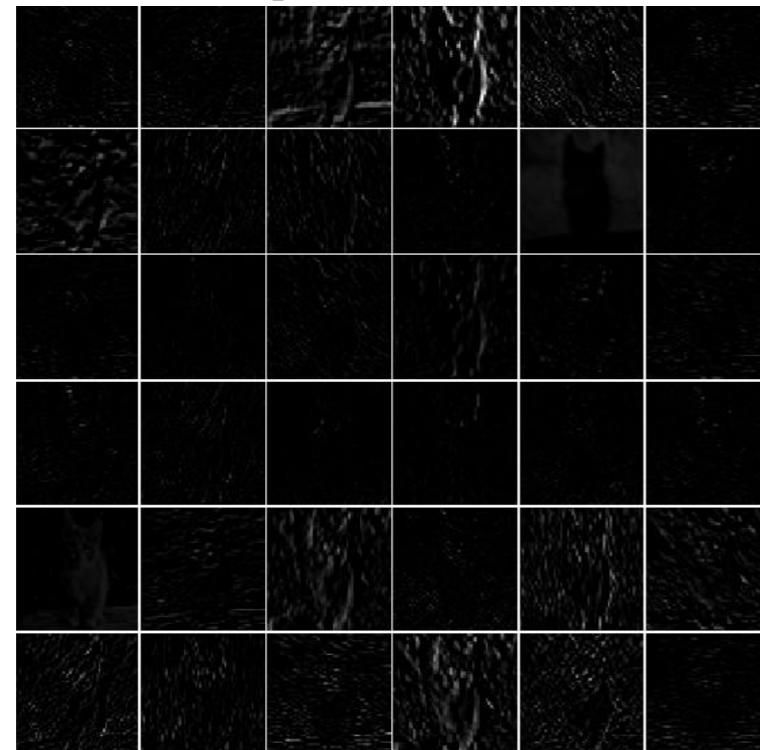
<https://cs.stanford.edu/people/karpathy/cnnembed/>



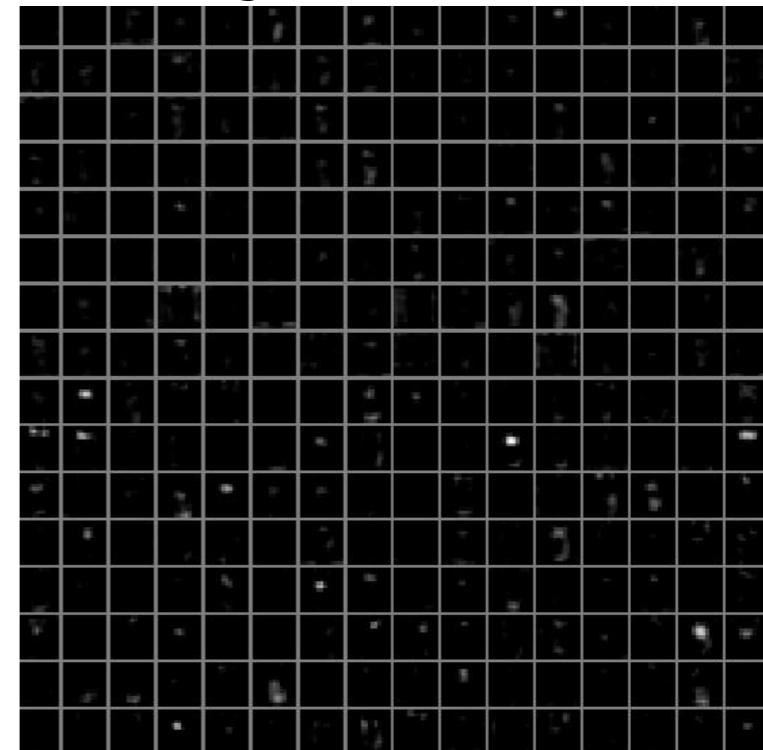
Анализ активации нейронов

**просто показываем активации во время прямого прохода
каждый квадратик – активация какого-то фильтра:**

первые слои



глубокие слои



чёрный цвет – ноль, на поздних слоях активации разреженные и локализованные!

<https://stevenrush.github.io/understanding-cnn/>

Анализ активации нейронов

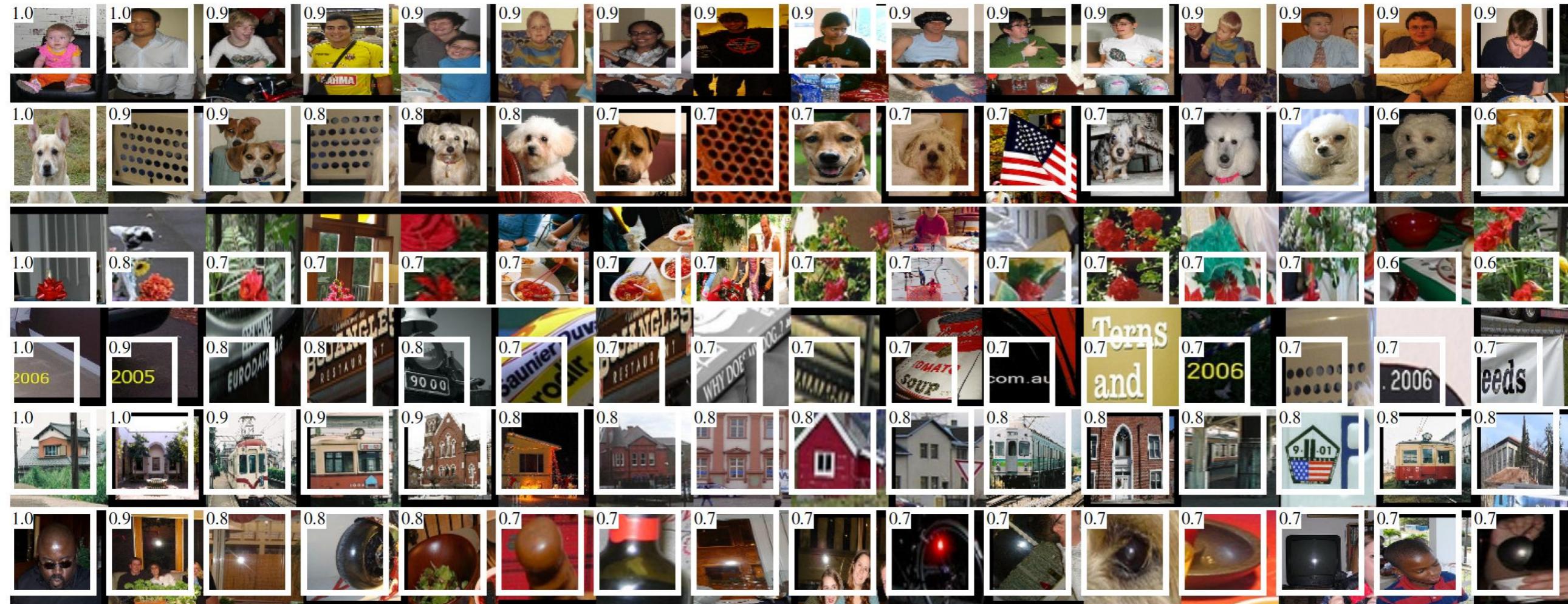
Средние слои: на каких изображениях максимальные значения активаций



<https://arxiv.org/pdf/1412.6806.pdf>

Анализ активации нейронов

Средние слои: на каких изображениях максимальные значения активаций



Rich feature hierarchies for accurate object detection and semantic segmentation – Girshick, et al - 2013

Чувствительность к удалению (Occlusion sensitivity)

Чтобы оценить, какие пиксели отвечают за отнесению к классу

Закрывать последовательно часть изображения – 2D-гистограмма вероятности
принадлежности к заданному классу при закрытии с центром в ij -м пикселе

Matthew D Zeiler, Rob Fergus «Visualizing and Understanding Convolutional Networks»

<https://arxiv.org/pdf/1311.2901.pdf>

Чувствительность к удалению (Occlusion sensitivity)

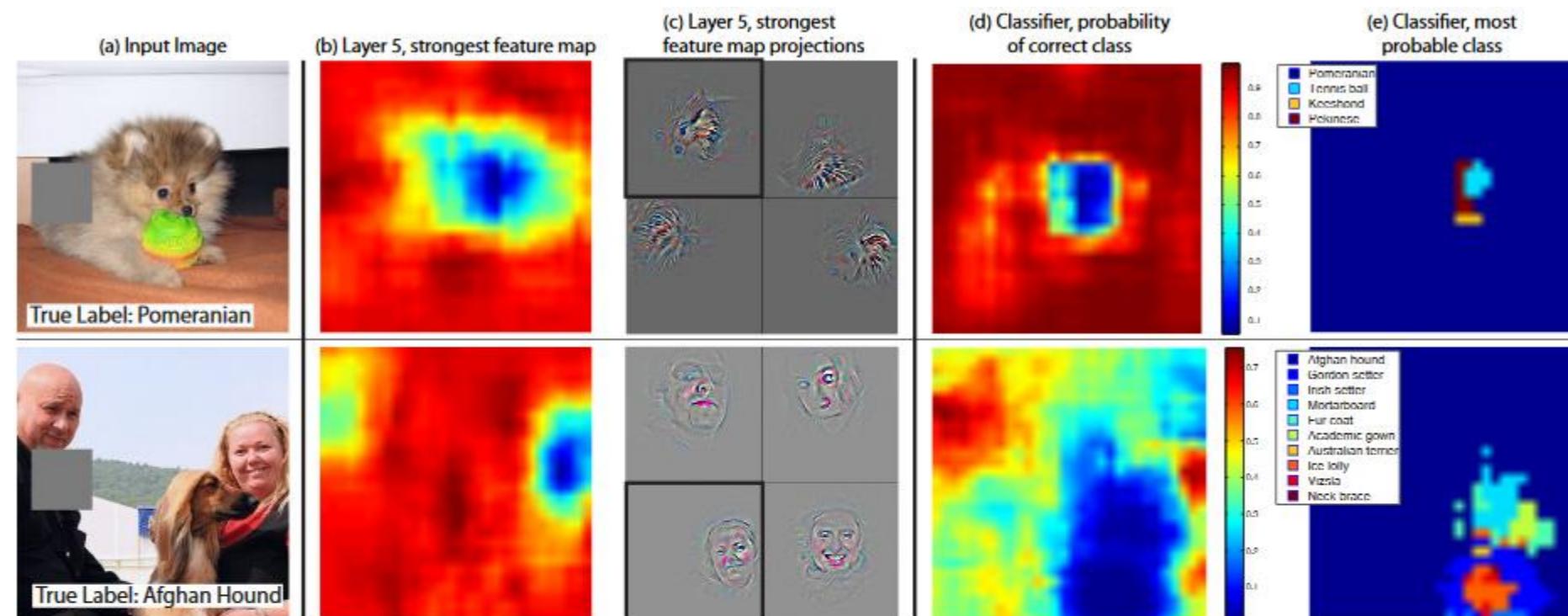


Figure 7. Three test examples where we systematically cover up different portions of the scene with a gray square (1st column) and see how the top (layer 5) feature maps ((b) & (c)) and classifier output ((d) & (e)) changes. (b): for each position of the gray scale, we record the total activation in one layer 5 feature map (the one with the strongest response in the unoccluded image). (c): a visualization of this feature map projected down into the input image (black square), along with visualizations of this map from other images. The first row example shows the strongest feature to be the dog's face. When this is covered-up the activity in the feature map decreases (blue area in (b)). (d): a map of correct class probability, as a function of the position of the gray square. E.g. when the dog's face is obscured, the probability for "pomeranian" drops significantly. (e): the most probable label as a function of occluder position. E.g. in the 1st row, for most locations it is "pomeranian", but if the dog's face is obscured but not the ball, then it predicts "tennis ball". In the 2nd example, text on the car is the strongest feature in layer 5, but the classifier is most sensitive to the wheel. The 3rd example contains multiple objects. The strongest feature in layer 5 picks out the faces, but the classifier is sensitive to the dog (blue region in (d)), since it uses multiple feature maps.

«Saliency maps» – градиенты (их модули) по входу



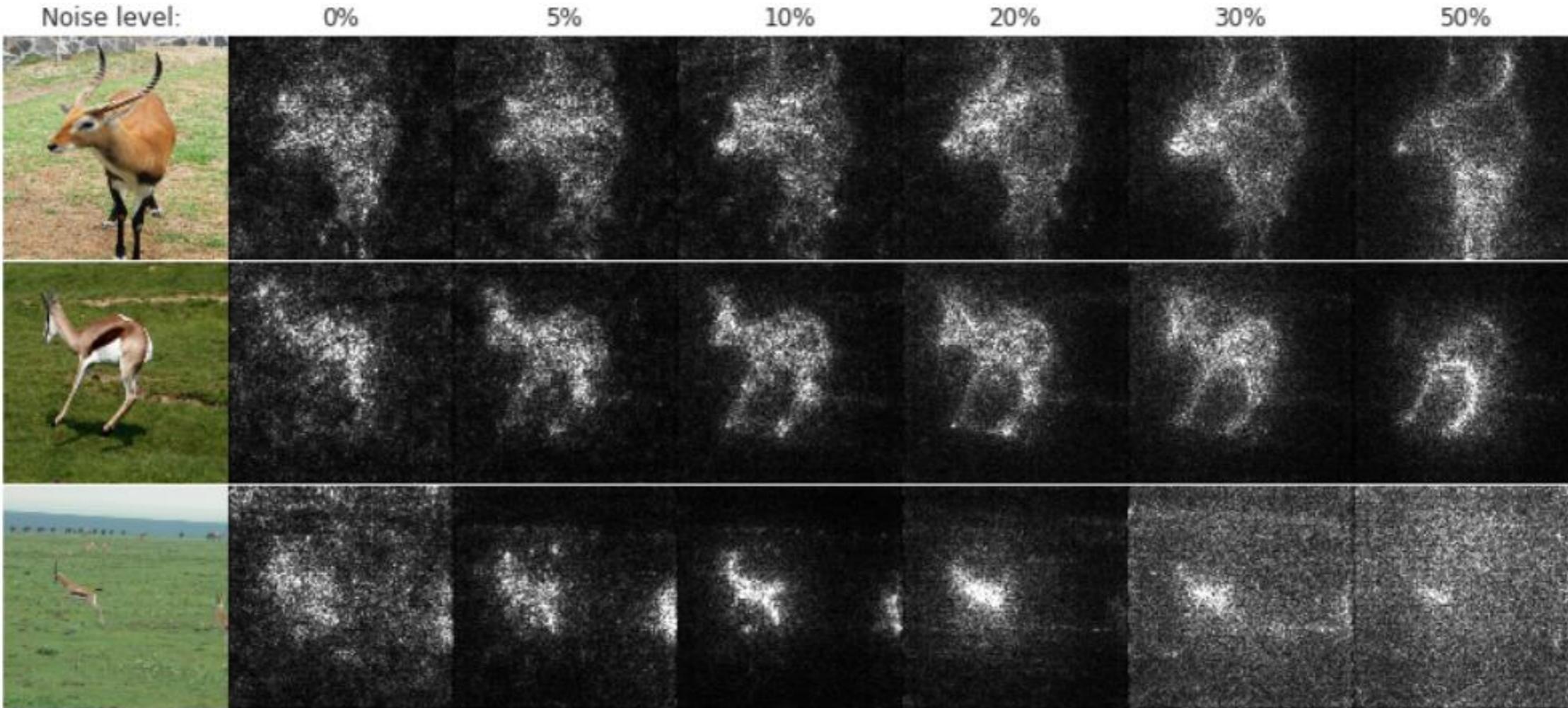
точнее здесь $\max(|R'|, |G'|, |B'|)$

Karen Simonyan et al «Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps» 2014

<https://arxiv.org/pdf/1312.6034.pdf>

«Saliency maps» – градиенты по входу

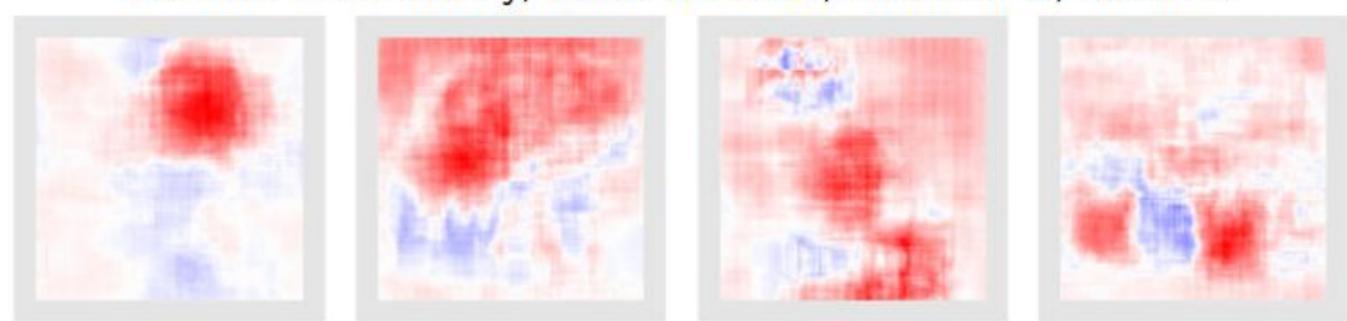
с помощью шума можно получать более адекватные иллюстрации



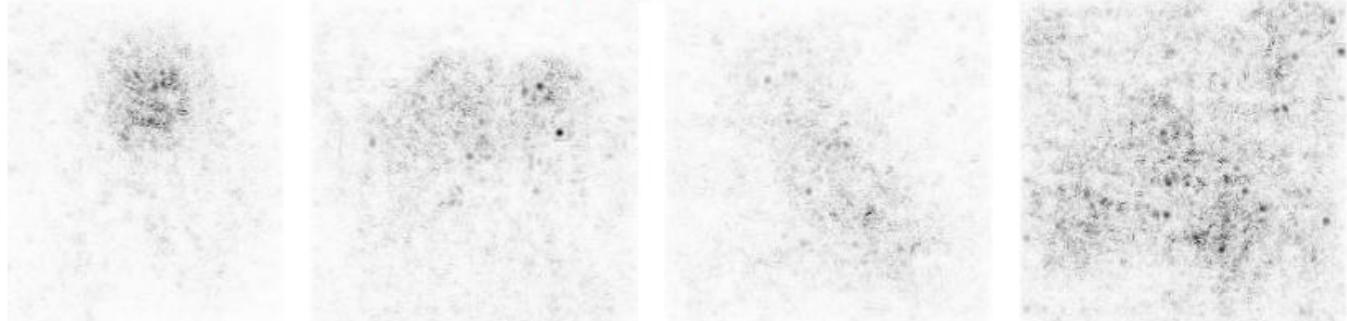
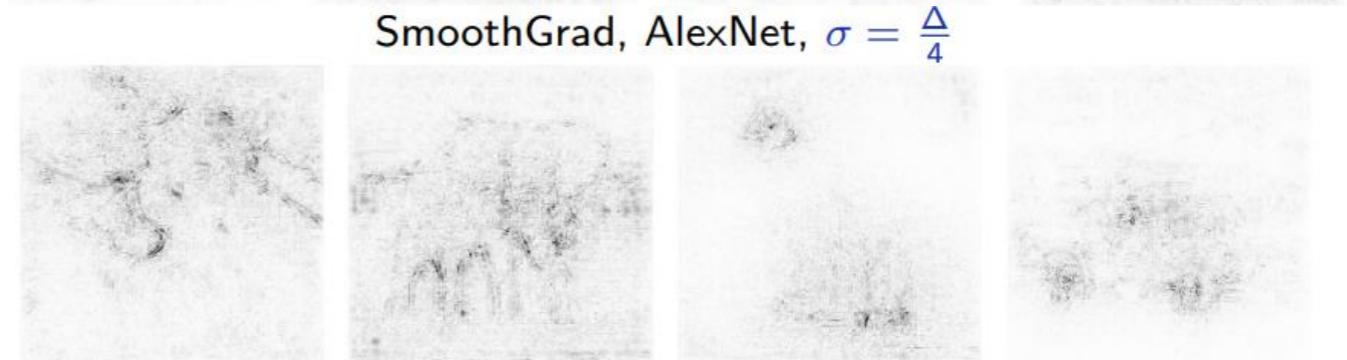
Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, Martin Wattenberg «SmoothGrad: removing noise by adding noise» //
<https://arxiv.org/abs/1706.03825>

Сравнение

Original images

Occlusion mask 32×32 Occlusion sensitivity, mask 32×32 , stride of 2, AlexNet

Gradient, AlexNet

SmoothGrad, AlexNet, $\sigma = \frac{\Delta}{4}$ 

Анализ отдельных нейронов: Class Model Visualisation

Сгенерировать изображения, максимизирующие активацию выделенного нейрона
(методом обратного распространения ошибки)

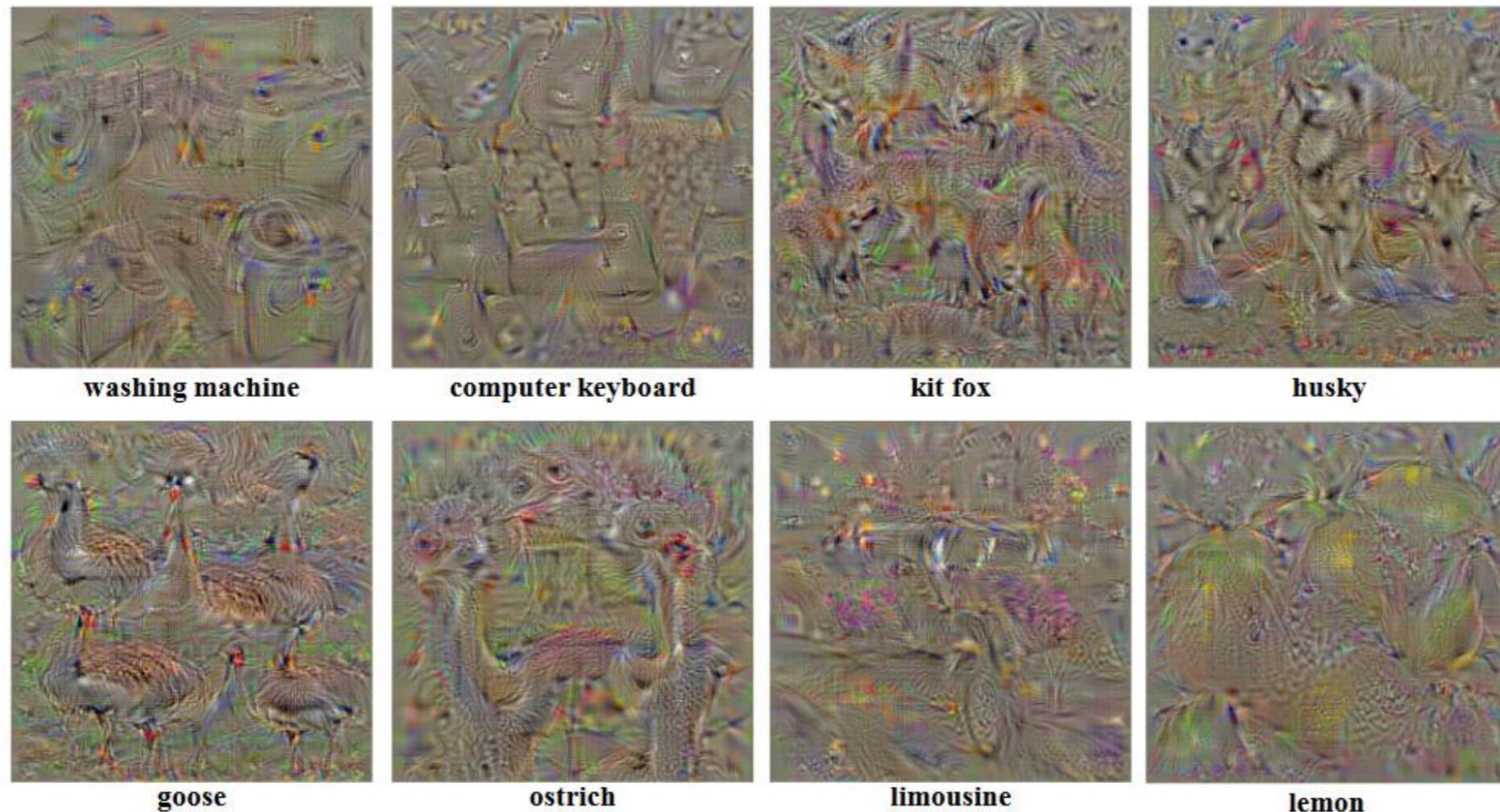


Figure 1: Numerically computed images, illustrating the class appearance models, learnt by a ConvNet, trained on ILSVRC-2013.
Note how different aspects of class appearance are captured in a single image.

Karen Simonyan et al «Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps» 2014

<https://arxiv.org/pdf/1312.6034.pdf>

Анализ отдельных нейронов / слоёв

ищем изображения, максимизирующие определённые активации

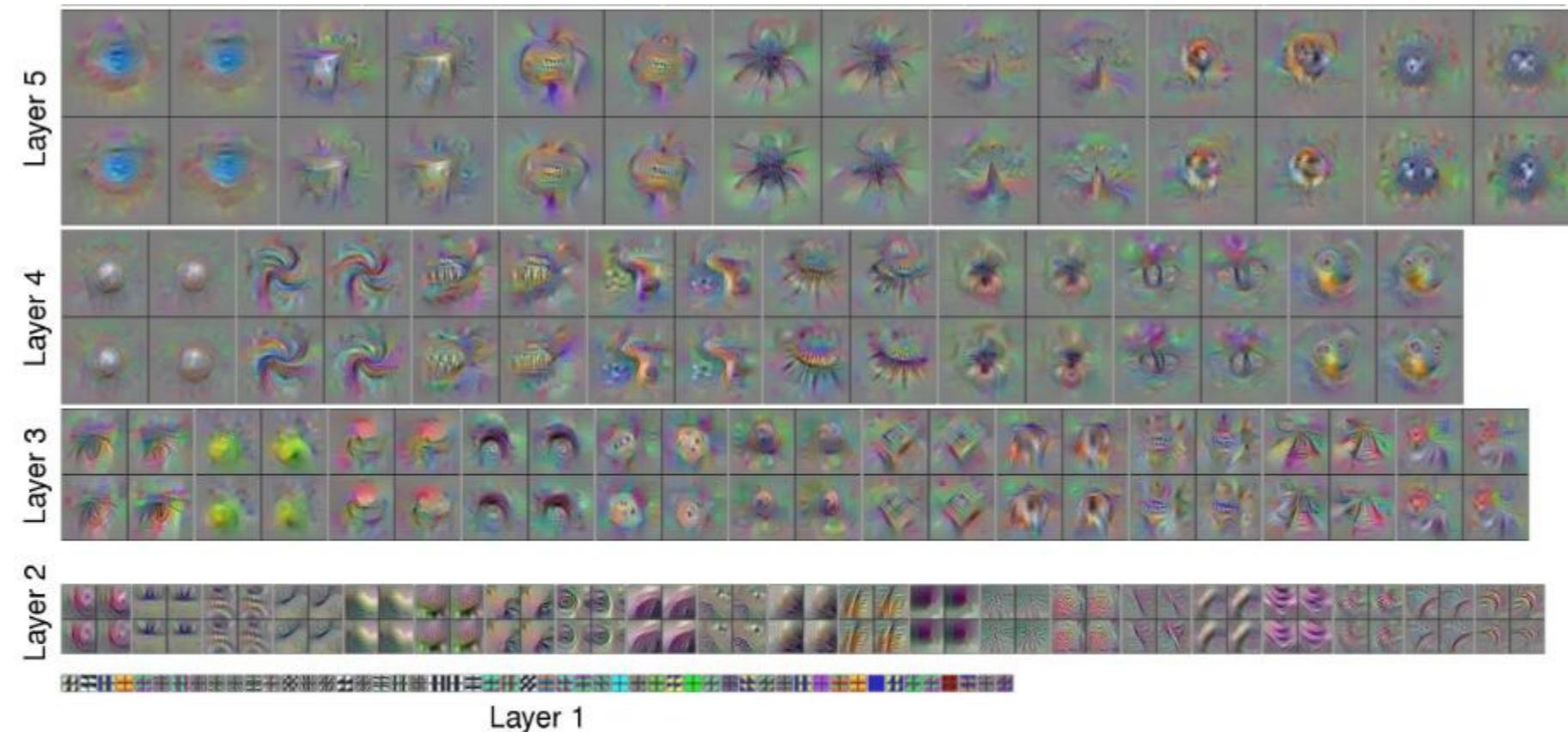
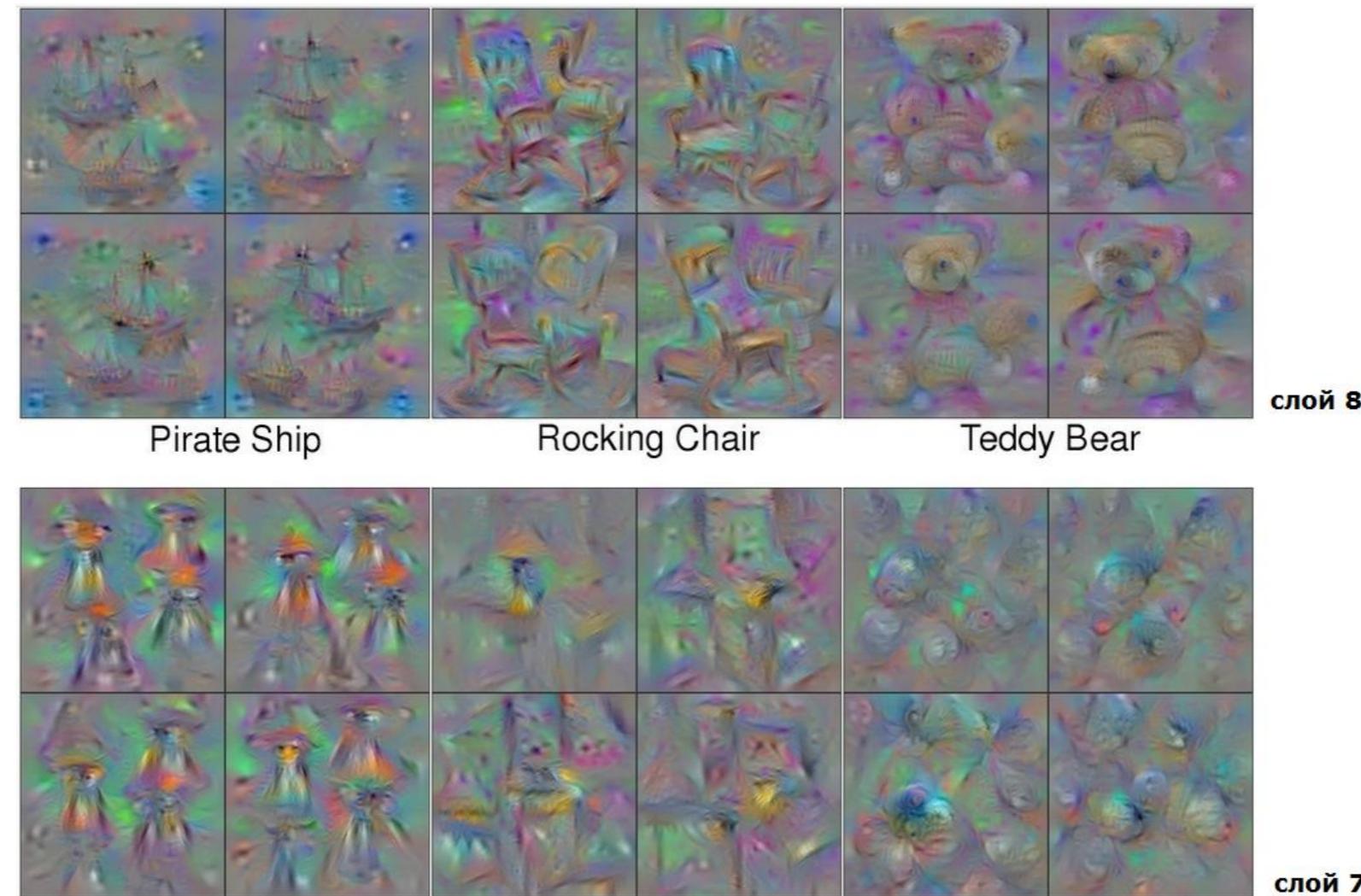


Figure 5. Visualization of example features of eight layers of a deep, convolutional neural network. The images reflect the true sizes of the features at different layers. In each layer, we show visualizations from 4 random gradient descent runs for each channel. While these images are hand picked to showcase the diversity and interpretability of the visualizations, one image for each filter of all five convolutional layers is shown in Figure S1 in supplementary information. One can recognize important features of objects at different scales, such as edges, corners, wheels, eyes, shoulders, faces, handles, bottles, etc. The visualizations show the increase in complexity and variation on higher layers, comprised of simpler components from lower layers. The variation of patterns increases with increasing layer number, indicating that increasingly invariant representations are learned. In particular, the jump from Layer 5 (the last convolution layer) to Layer 6 (the first fully-connected layer) brings about a large increase in variation. Best viewed electronically, zoomed in.

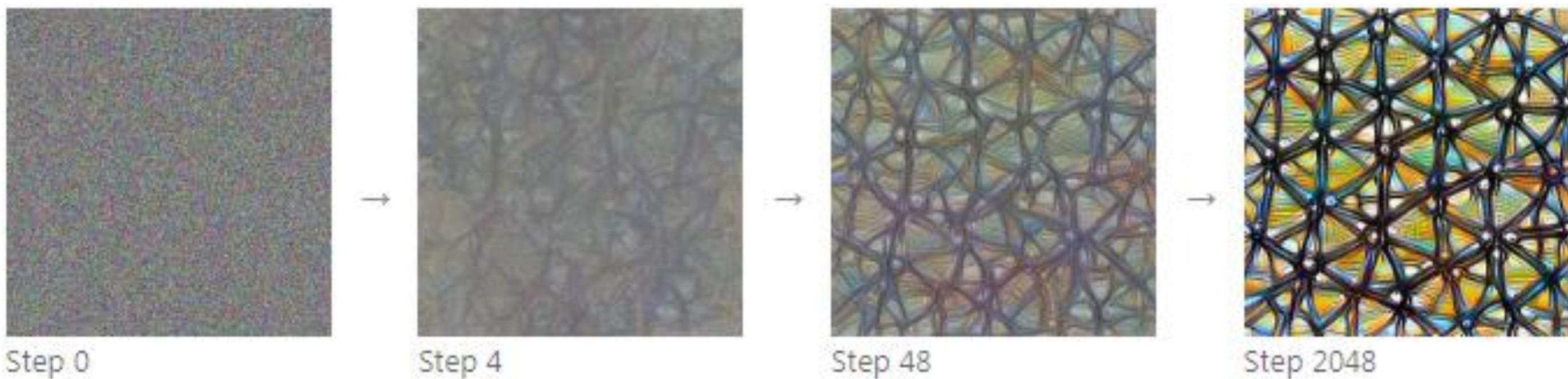
Анализ отдельных нейронов / слоёв



тут ещё + регуляризация

Анализ отдельных нейронов / слоёв

когда с помощью обратного распространения смотрим на чём активируется нейрон..



<https://distill.pub/2017/feature-visualization/>

Анализ отдельных нейронов / слоёв

Смогли сделать так:



Step 1

Step 32

Step 128

Step 256

Step 2048

Обычно получается так:



Step 1

Step 32

Step 128

Step 256

Step 2048

Анализ отдельных нейронов / слоёв

как сделать красиво... грамотная регуляризация

Борьба с высокочастотным шумом

- **наказывать разницу соседних пикселей**
- **размытие изображения после k шагов**
 - **bilateral filter**

Устойчивость к преобразованиям

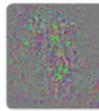
- **можно преобразовывать картинку перед градиентным шагом**
Learned priors – генерировать похожую на реальную картинку
 - **м.б. + GAN**

- **градиентный спуск в другом – декоррелируемом пространстве (в базисе Фурье)**

<https://distill.pub/2017/feature-visualization/>

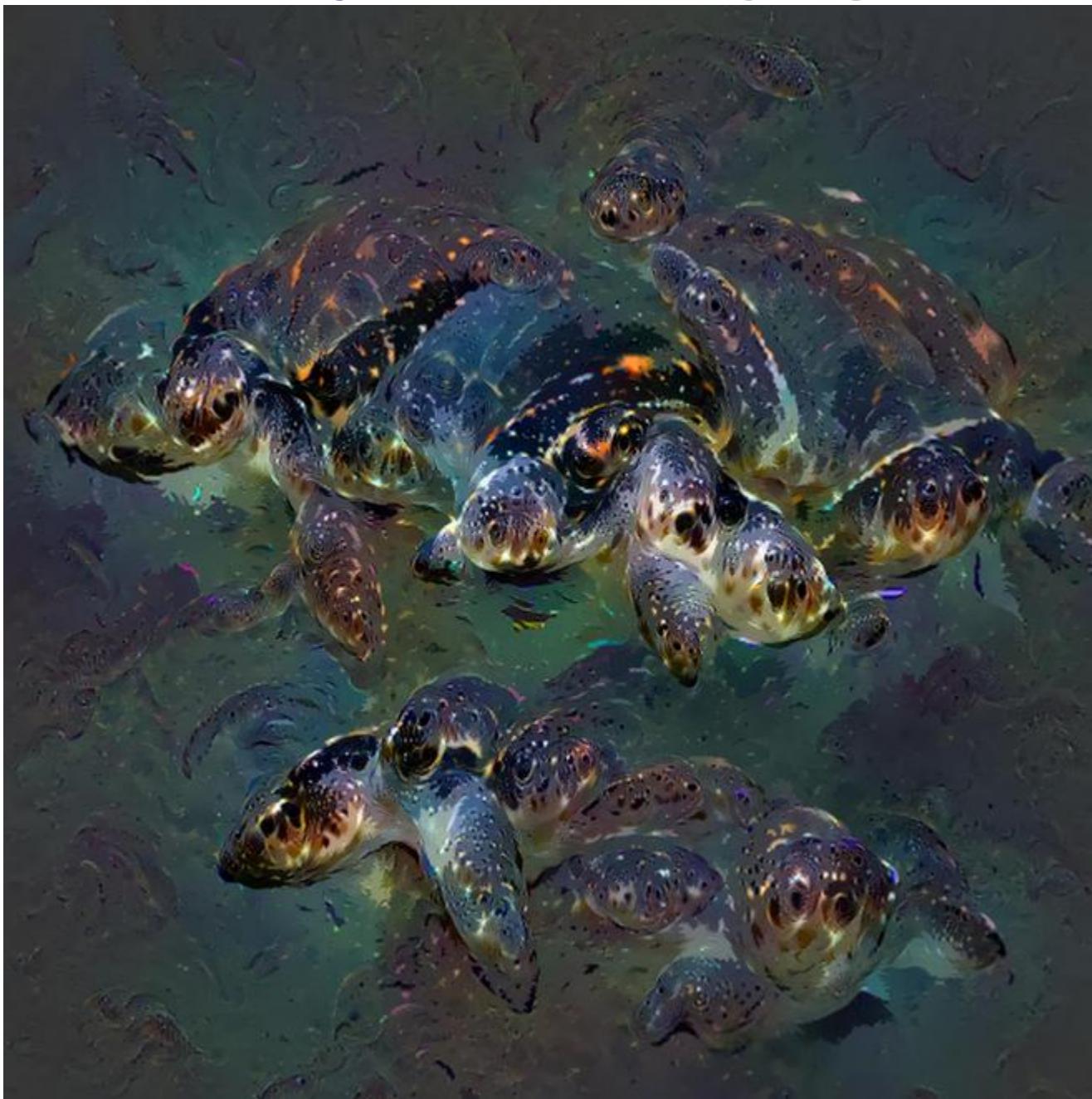
Анализ отдельных нейронов / слоёв

развитие улучшений изображений:

	Unregularized Penalization	Frequency Robustness	Transformation Prior	Learned Prior	Dataset Examples						
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Mordvintsev, et al., 2015 [4] Introduced jitter & multi-scale. Explored GMM priors for classes.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Øygard, et al., 2015 [15] Introduces gradient blurring. (Also uses jitter.)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Tyka, et al., 2016 [16] Regularizes with bilateral filters. (Also uses jitter.)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Mordvintsev, et al., 2016 [17] Normalizes gradient frequencies. (Also uses jitter.)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
							Nguyen, et al., 2016 [18] Paramaterizes images with GAN generator.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
							Nguyen, et al., 2016 [10] Uses denoising autoencoder prior to make a generative model.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

<https://distill.pub/2017/feature-visualization/>

Нейроискусство <https://mtyka.github.io/deepdream/2016/02/05/bilateral-class-vis.html>



За что отвечают отдельные нейроны, каналы, слои

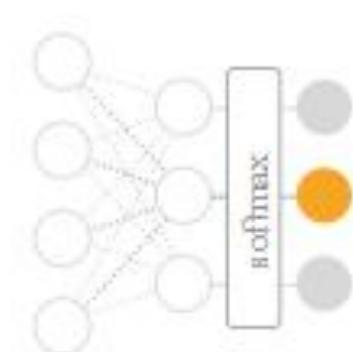
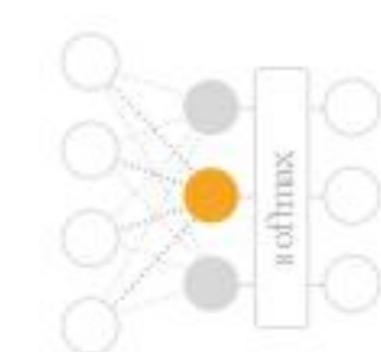
Different **optimization objectives** show what different parts of a network are looking for.

n layer index

x,y spatial position

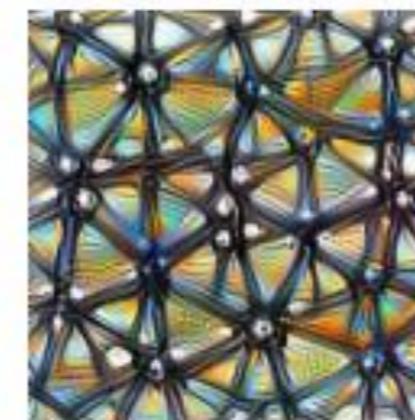
z channel index

k class index



Neuron

$\text{layer}_n[x, y, z]$



Channel

$\text{layer}_n[:, :, :, z]$



Layer/DeepDream

$\text{layer}_n[:, :, :, :]^2$



Class Logits

$\text{pre_softmax}[k]$



Class Probability

$\text{softmax}[k]$

Class Logits – оптимизация до softmax-а → более красивые картинки

Как соотносится визуализация с действительностью

Dataset Examples show us what neurons respond to in practice



Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes?
mixed4a, Unit 6

Animal faces—or snouts?
mixed4a, Unit 240

Clouds—or fluffiness?
mixed4a, Unit 453

Buildings—or sky?
mixed4a, Unit 492

Позитивные и негативные примеры



Negative optimized



Minimum activation examples



Slightly negative activation examples



Slightly positive activation examples



Maximum activation examples



Positive optimized

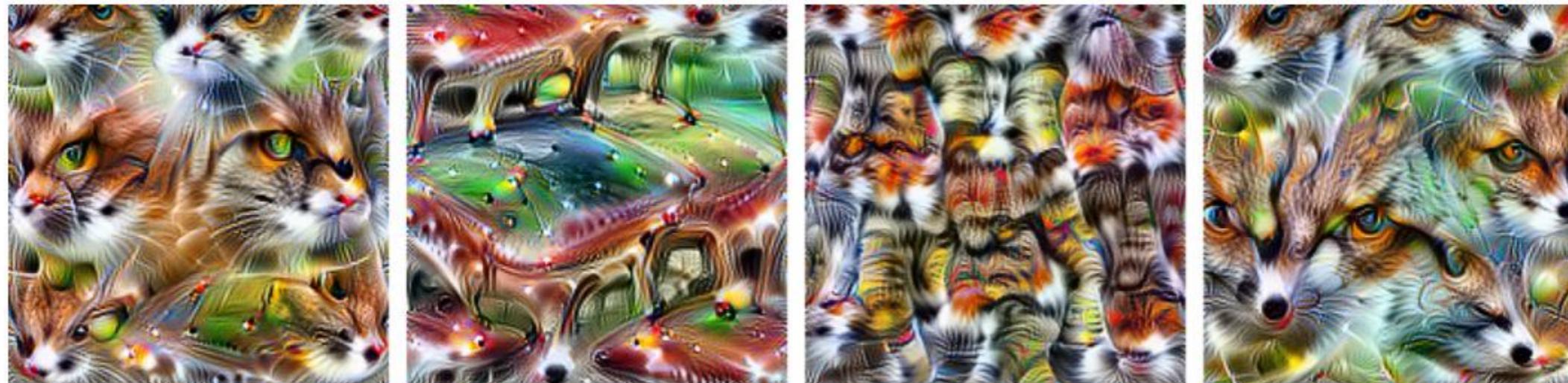
Необходимость большого числа экспериментов с рандомизацией



Dataset examples

Optimization with diversity. Layer mixed4a, Unit 143

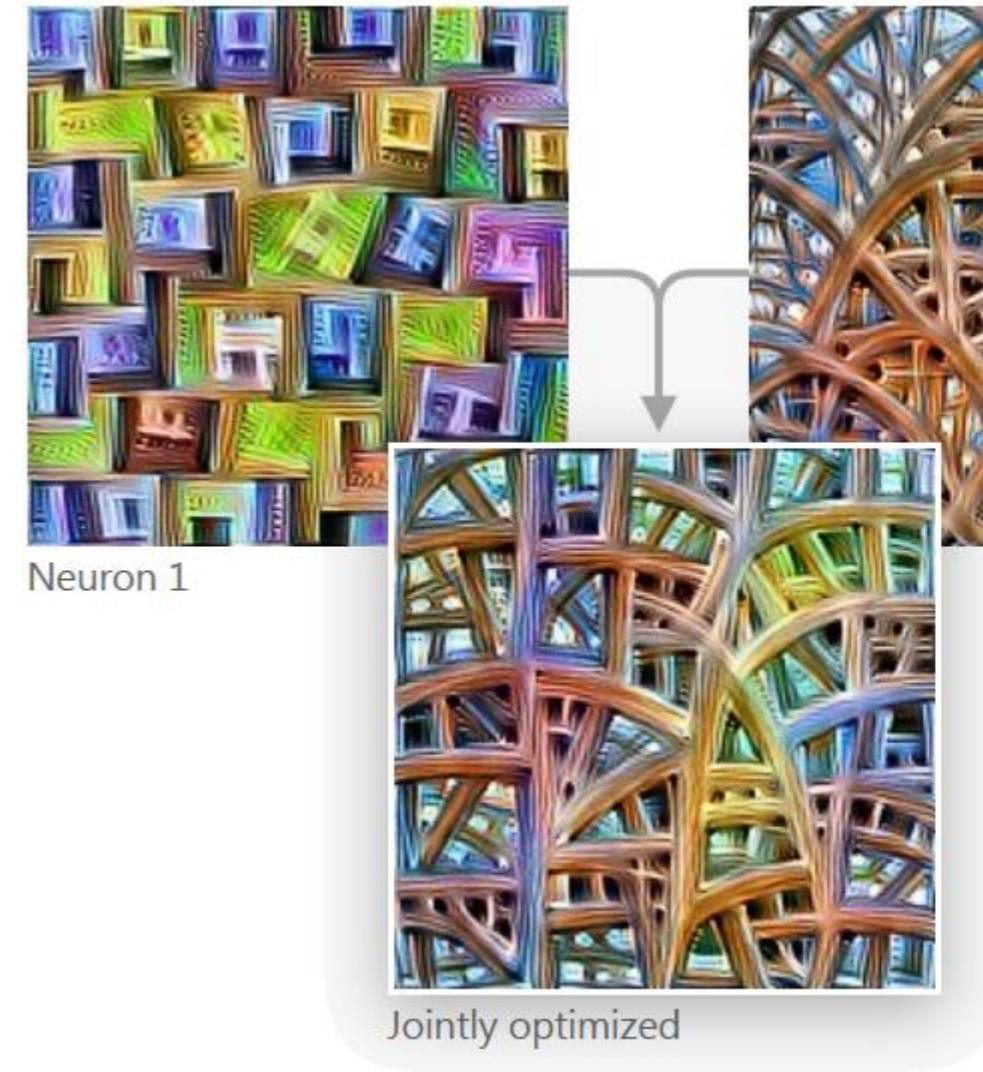
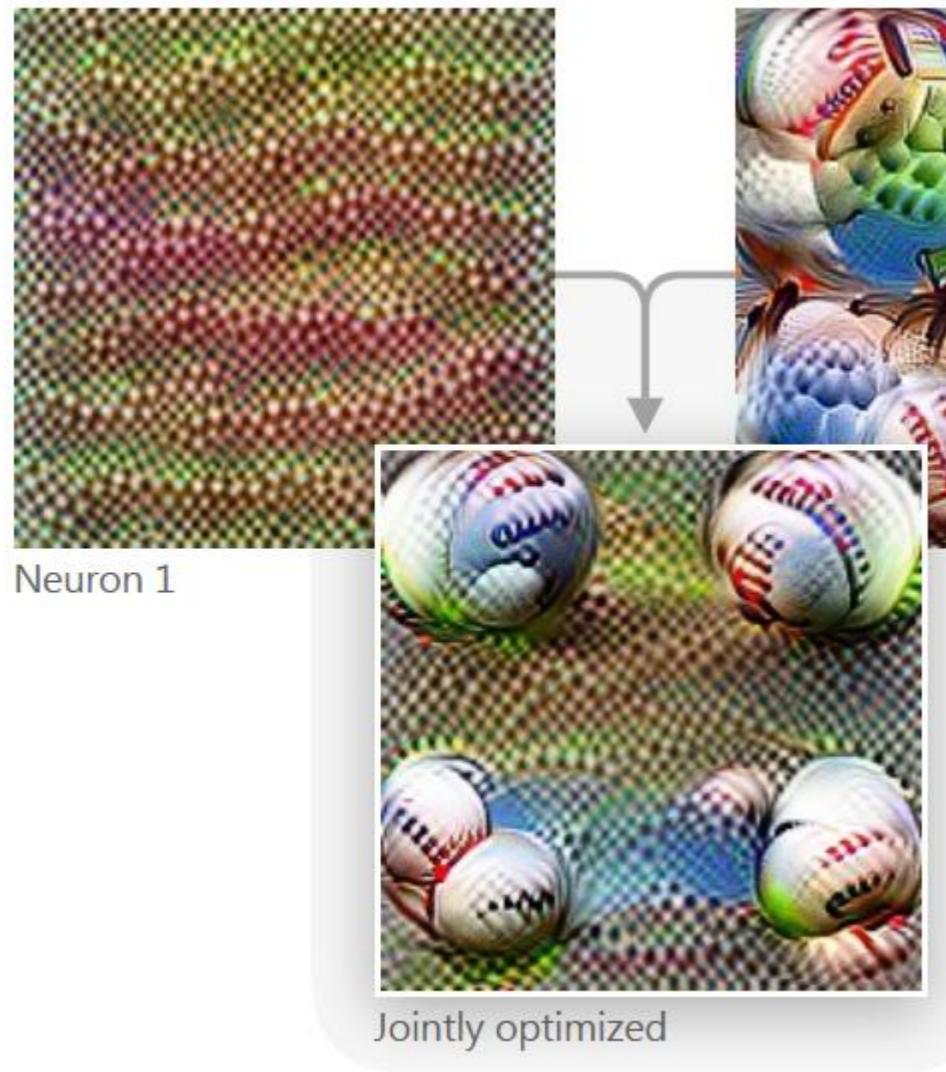
По одному эксперименту кажется, что «голова собаки» / машины



Dataset examples

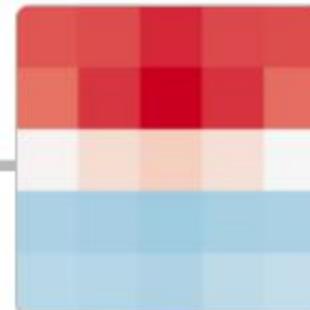
Optimization with diversity show cats, foxes, but also cars. Layer mixed4e, Unit 55

Можно оптимизировать сразу несколько нейронов



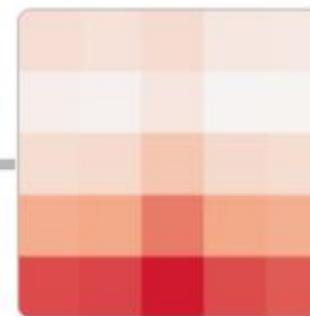
Визуализация: что можно обнаружить

Windows (4b:237)
excite the car detector
at the top and inhibit
at the bottom.

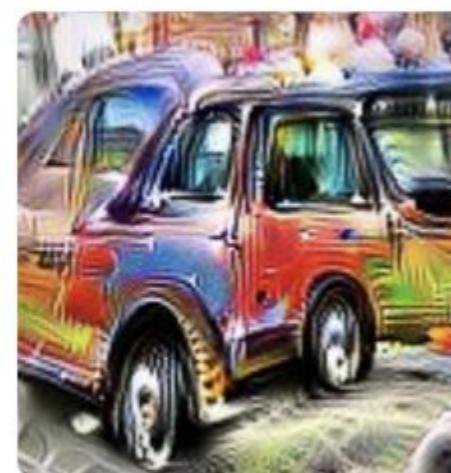
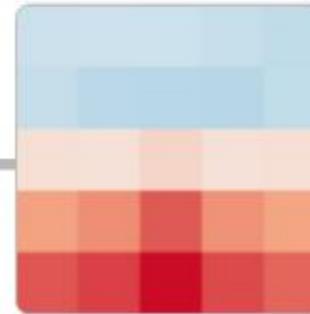


- positive (excitation)
- negative (inhibition)

Car Body (4b:491)
excites the car
detector, especially at
the bottom.



Wheels (4b:373) excite
the car detector at the
bottom and inhibit at
the top.



A **car detector** (4c:447)
is assembled from
earlier units.

нейрон детектирующий автомобиль и ясную логику его работы (окно + кузов + колёса)
но дальше он используется странно... при детектировании собаки

Визуализация: исследование нейронов детектирование кривых



Each neuron's ideal curve, created with feature visualization, which uses optimization to find superstimuli.

можно рассмотреть конкретный нейрон на искусственном датасете



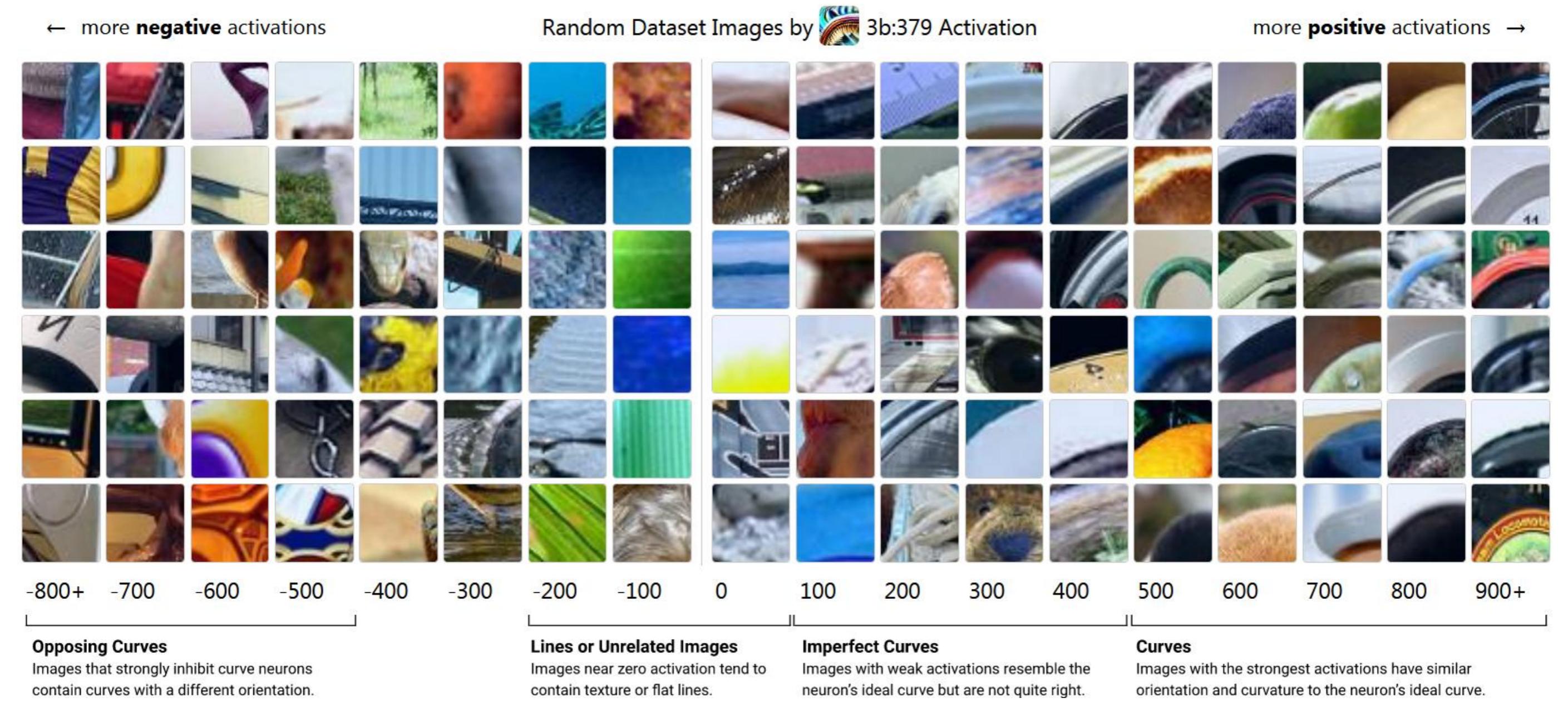
3b:379 Activations by Orientation



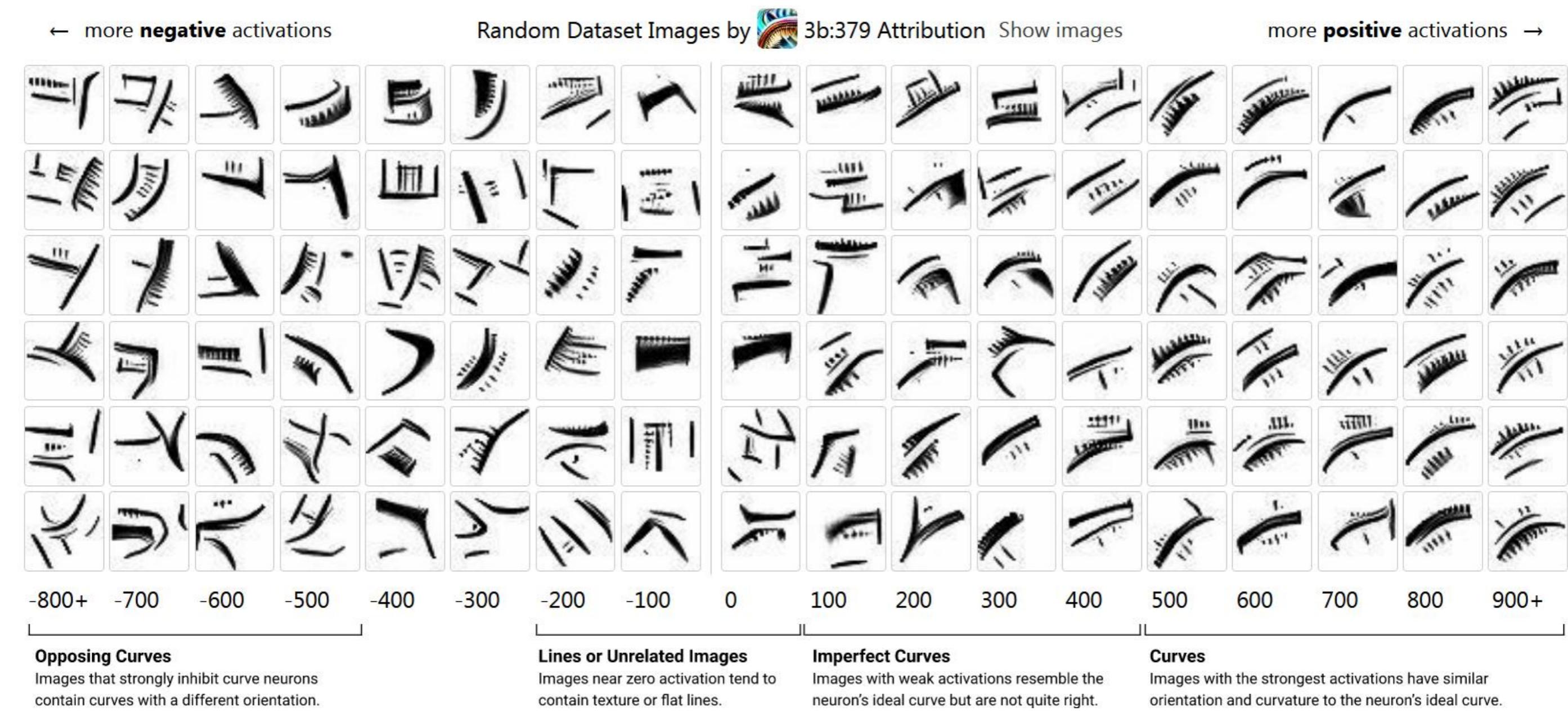
Later in this article we'll look in depth at activations to synthetic curve images.

<https://distill.pub/2020/circuits/curve-detectors/>

Визуализация: исследование нейронов



Визуализация: исследование нейронов



Визуализация: семантические словари

{активация нейрона: его визуализация}



<https://distill.pub/2018/building-blocks/>

Современные методы: FullGrad

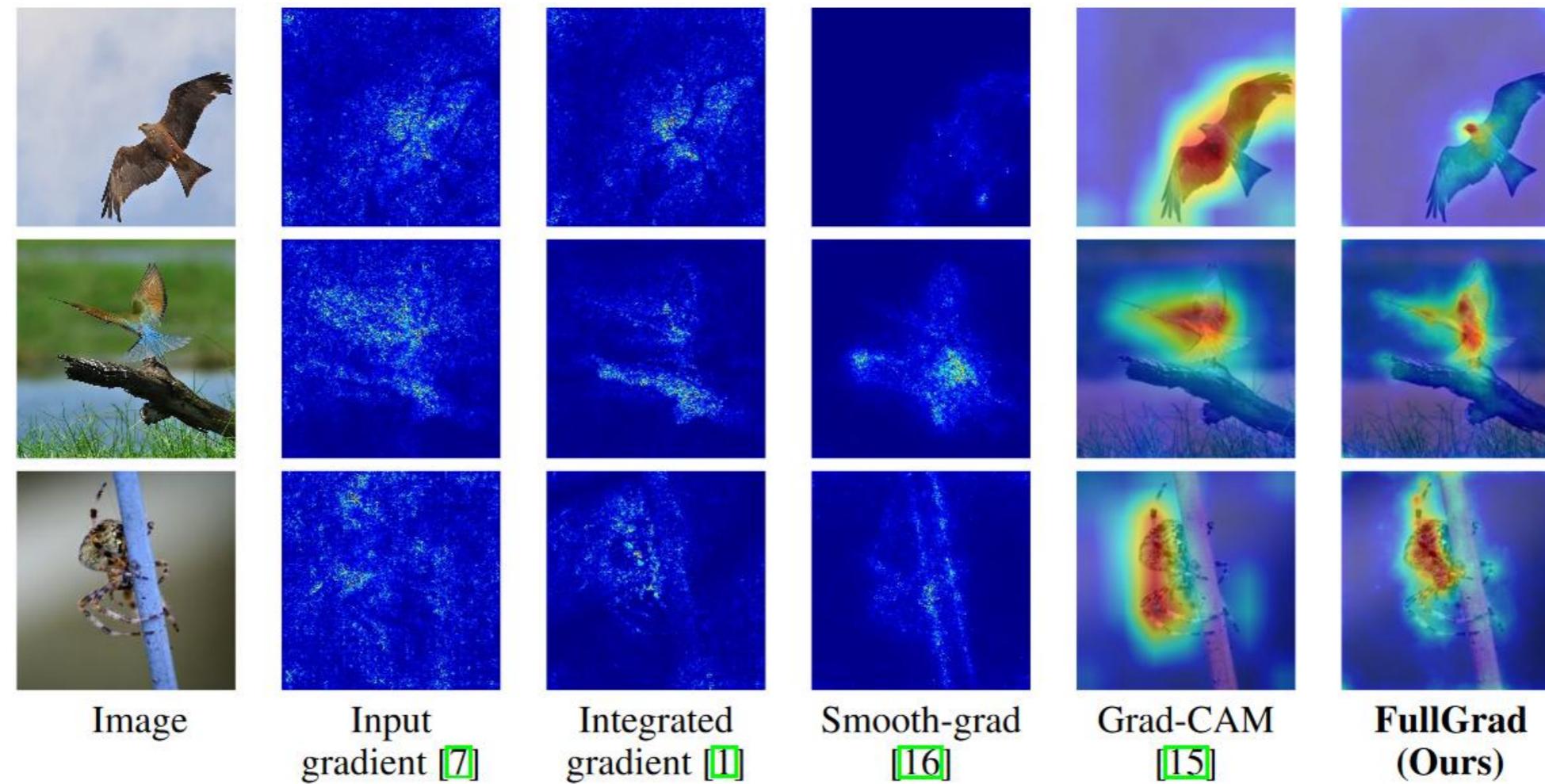
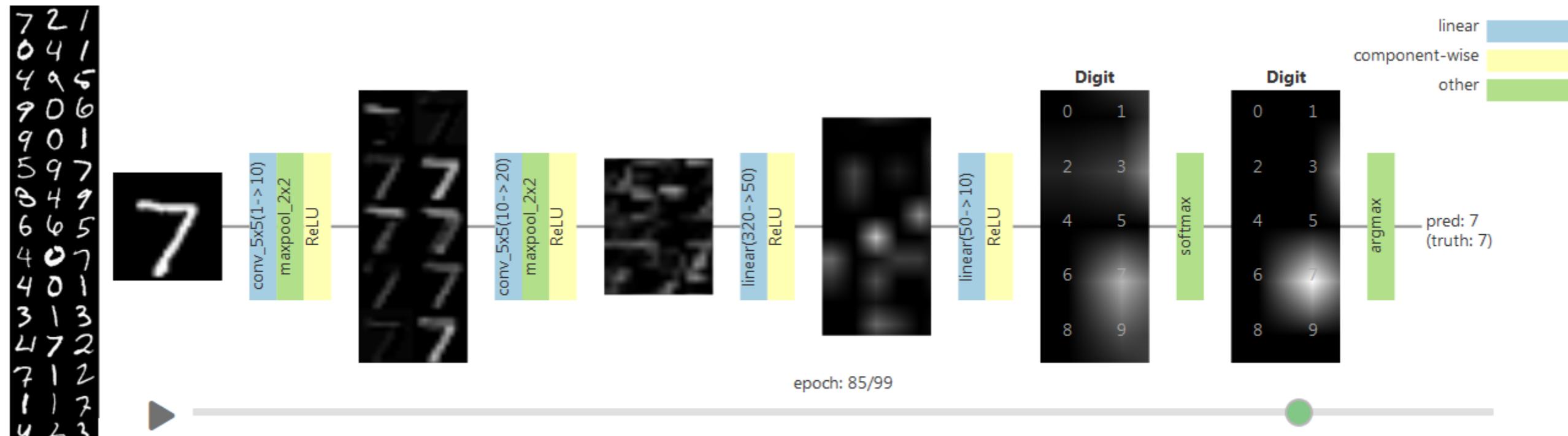


Figure 3: Comparison of different neural network saliency methods. Integrated-gradients [1] and smooth-grad [16] produce noisy object boundaries, while grad-CAM [15] indicates important regions without adhering to boundaries. FullGrad combine both desirable attributes by highlighting salient regions while being tightly confined within objects. For more results, please see supplementary material.

Suraj Srinivas, François Fleuret «Full-Gradient Representation for Neural Network Visualization» <https://arxiv.org/pdf/1905.00780.pdf>

Современные методы



Neural network opened. The colored blocks are building-block functions (i.e. neural network layers), the gray-scale heatmaps are either the input image or intermediate activation vectors after some layers.

<https://distill.pub/2020/grand-tour/>

«Разумность нейросетей»

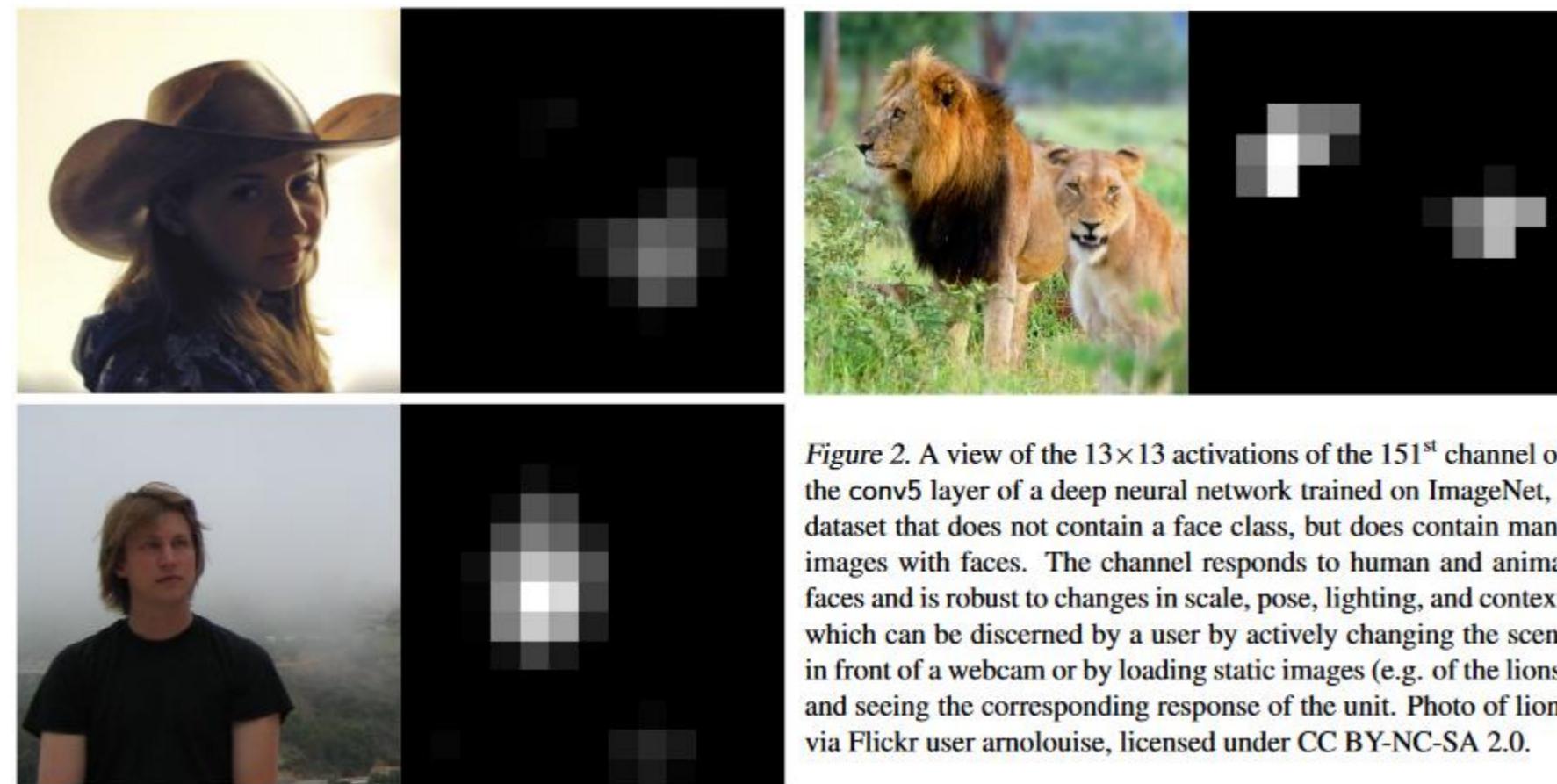


Figure 2. A view of the 13×13 activations of the 151st channel on the conv5 layer of a deep neural network trained on ImageNet, a dataset that does not contain a face class, but does contain many images with faces. The channel responds to human and animal faces and is robust to changes in scale, pose, lighting, and context, which can be discerned by a user by actively changing the scene in front of a webcam or by loading static images (e.g. of the lions) and seeing the corresponding response of the unit. Photo of lions via Flickr user arnolouise, licensed under CC BY-NC-SA 2.0.

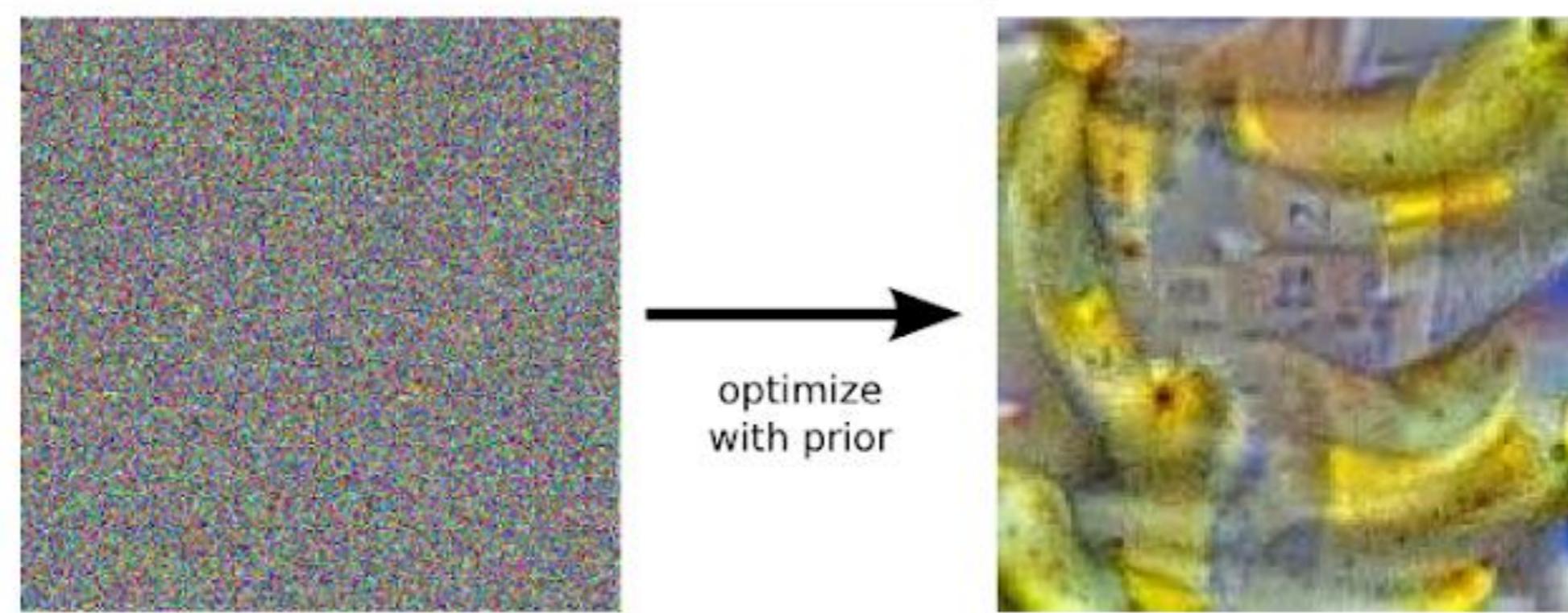
нет класса «голова», но она находится одним из каналов

Jason Yosinski «Understanding Neural Networks Through Deep Visualization» //
<https://arxiv.org/pdf/1506.06579.pdf>

Генерация изображений

**Натренировали НС-классификатор
Дали на вход случайный шум и стали градиентно менять изображение:
увеличить уверенность, что это «банан»**

Есть хак: соседние пиксели коррелированы



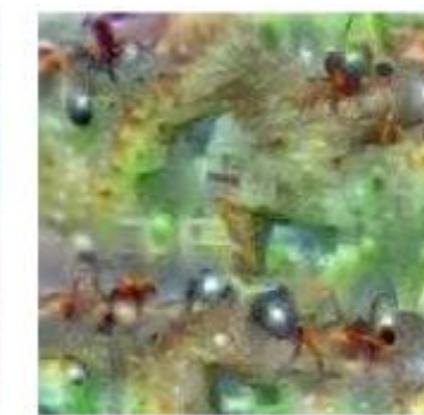
Генерация изображений



Hartebeest



Measuring Cup



Ant



Starfish



Anemone Fish



Banana



Parachute



Screw

**Можно максимизировать не вероятность класса,
а активацию какого-то нейрона**

Генерация изображений



Можем превращать изображения в изображения другого класса!

Генерация изображений

Зачем: чтобы понять, что выучились тому...



Класс «гантель» – всегда генерируется гантель с рукой!

<https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

Генерация изображений: восстановление из признаков

по стилю, ориентации (относительно камеры), цвету, яркости и т.п.

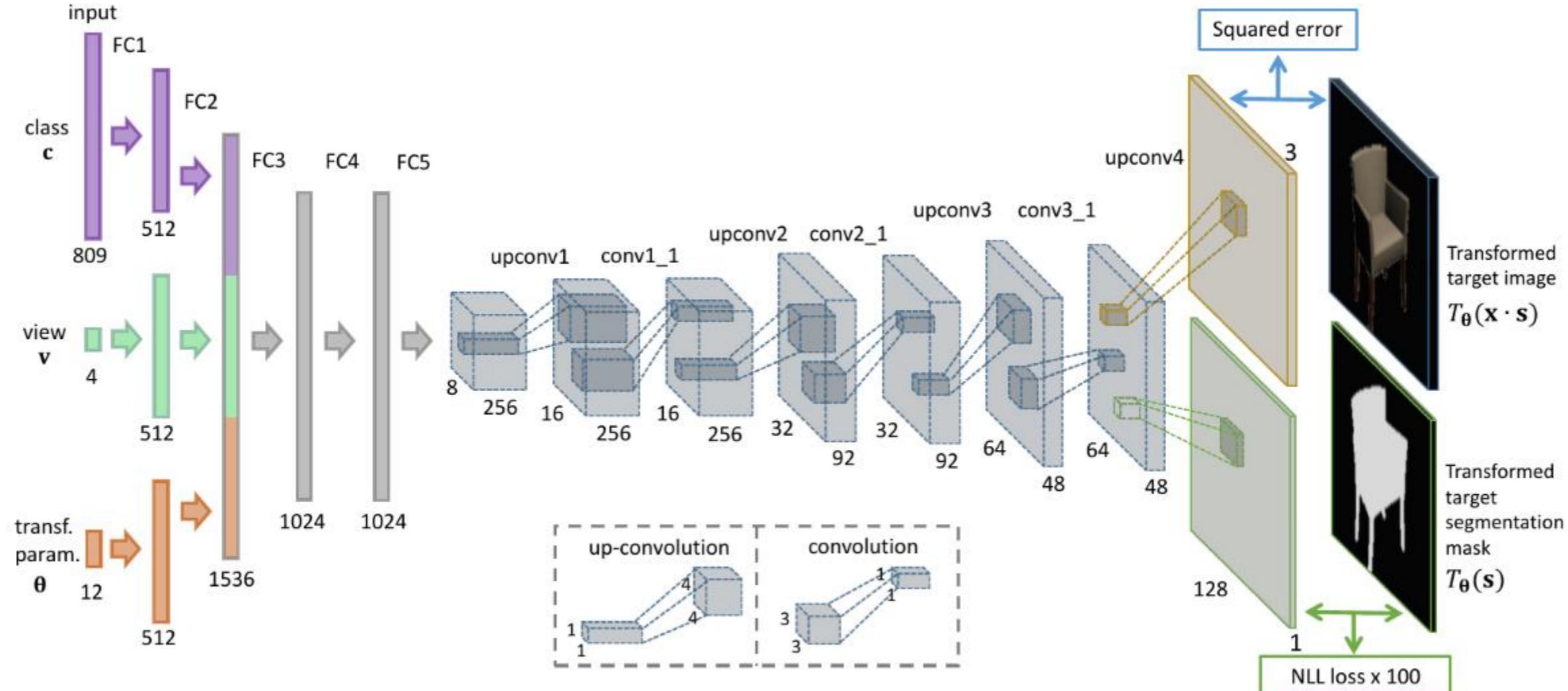
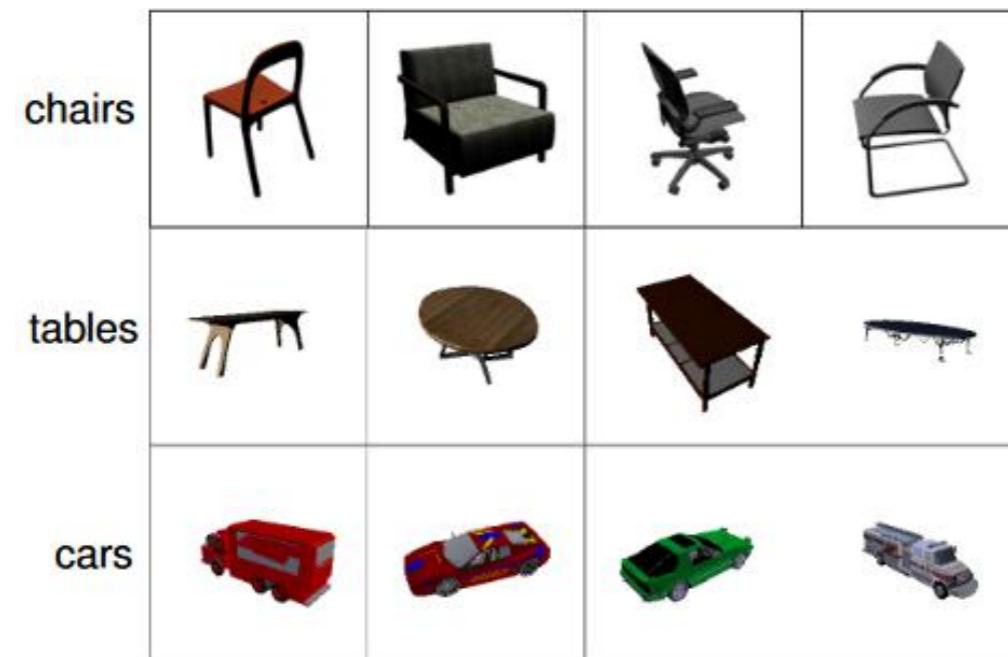


Fig. 1. Architecture of a 1-stream deep network (“1s-S-deep”) that generates 128×128 pixel images. Layer names are shown above: FC - fully connected, upconv - upsampling+convolution, conv - convolution.

«развернутая CNN»: параметры → 1024 признака → изображение и маска сегментации

Генерация изображений: восстановление из признаков



Выборка:

(ОНЕ-стиль, угол камеры, доп. параметры)



(изображение, маска сегментации)

**Маска для простоты
(можно забелить фон)**

**Деконволюционные слои (deconvolutional layers)
Деконволюция («обратная» операция к свёртке)**

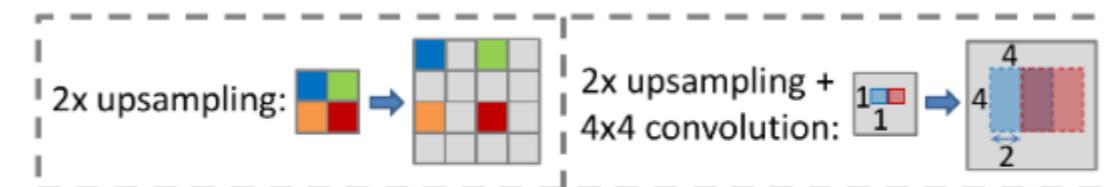


Fig. 2. Illustration of upsampling (left) and upsampling+convolution (right) as used in the generative network.

Dosovitskiy A. и др. «Learning to Generate Chairs, Tables and Cars with Convolutional Networks», 2017
 // <https://arxiv.org/pdf/1411.5928.pdf>

Генерация изображений – преобразование эталонов

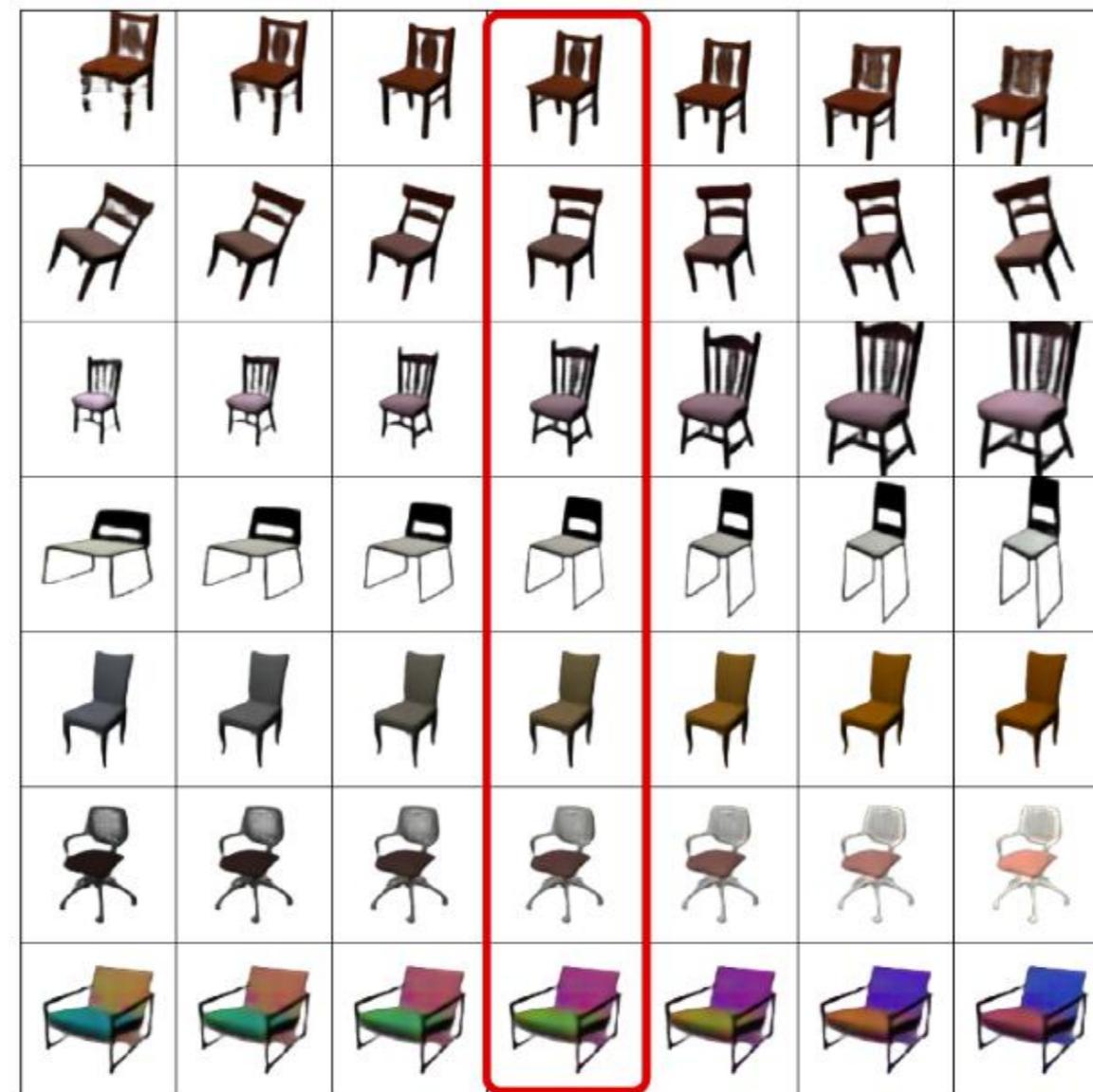


Fig. 7. Generation of chair images while activating various transformations. Each row shows one transformation: translation, rotation, zoom, stretch, saturation, brightness, color. The middle column shows the reconstruction without any transformation.

Генерация изображений – линейная комбинация стилей



Fig. 12. Examples of morphing different chairs, one morphing per row. Leftmost and rightmost chairs in each row are present in the training set, all intermediate ones are “invented” by the network. Rows are ordered by decreasing subjective quality of the morphing, from top to bottom.

Генерация изображений – Арифметика над признаками

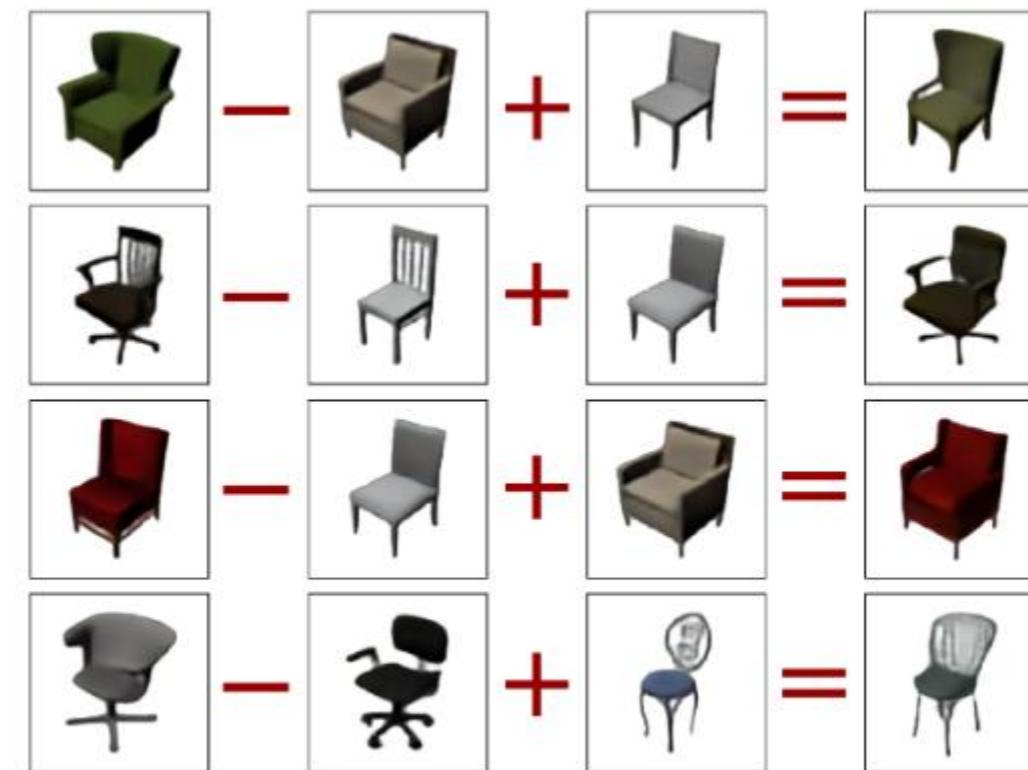
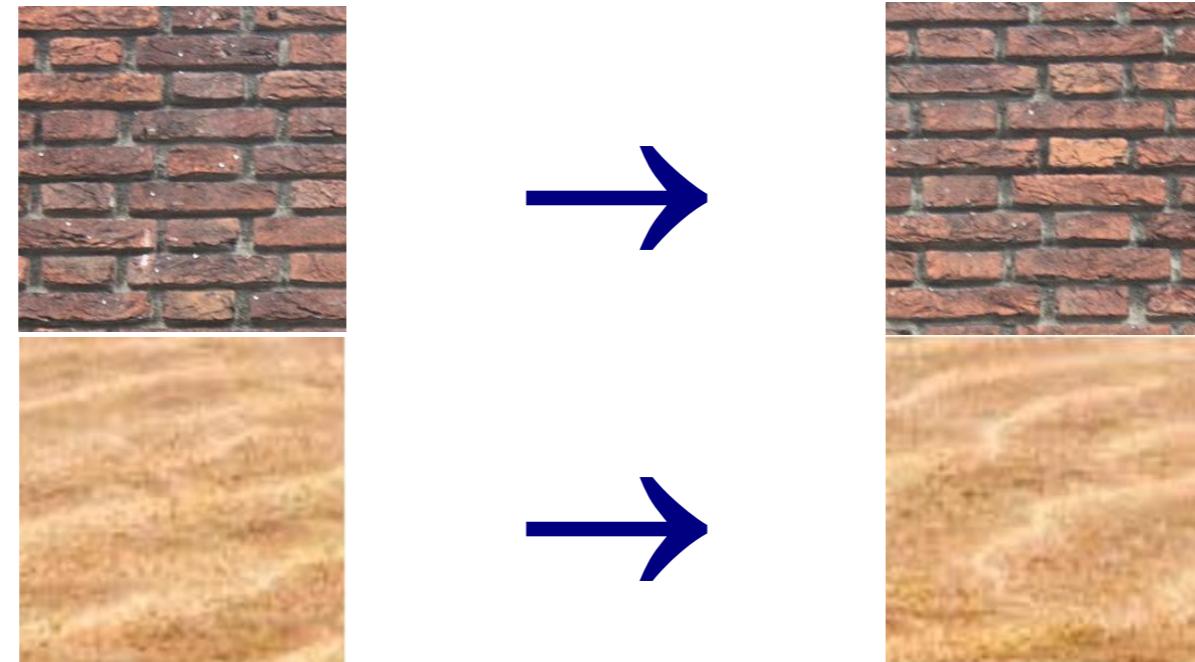


Fig. 16. Feature arithmetics: simple operations in the feature space lead to interpretable changes in the image space.

Генерация текстур



дано: простое однородное изображение

надо сгенерировать «похожее»

тоже однородное, не отличающееся по содержанию,

но не точно такое же (нет попиксельного сходства)

почему тут одна текстура?

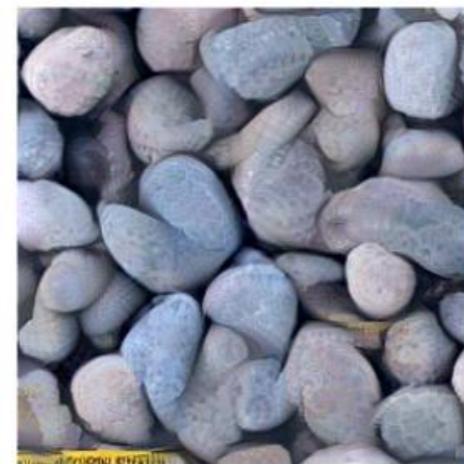
сохранена локальная структура, но не сохранена глобальная

Генерация текстур

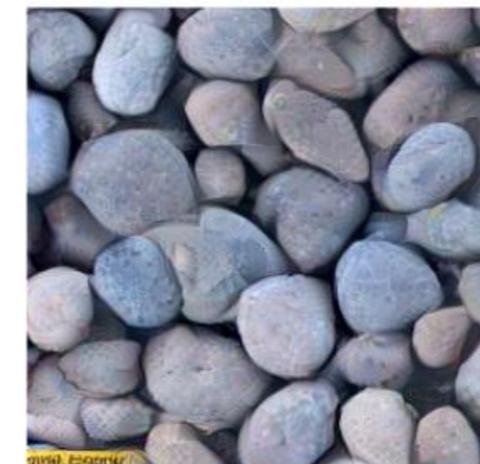
~1k parameters



~10k parameters



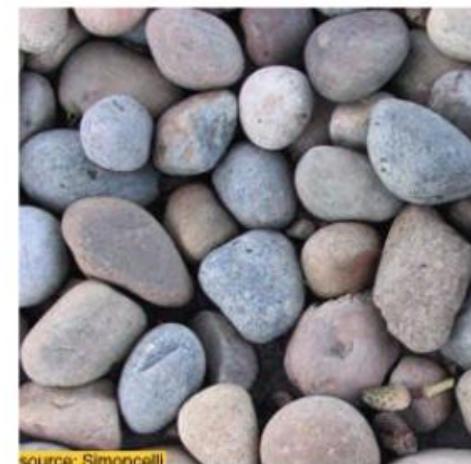
~177k parameters



~852k parameters



original



эксперимент по изменению числа параметров в модели

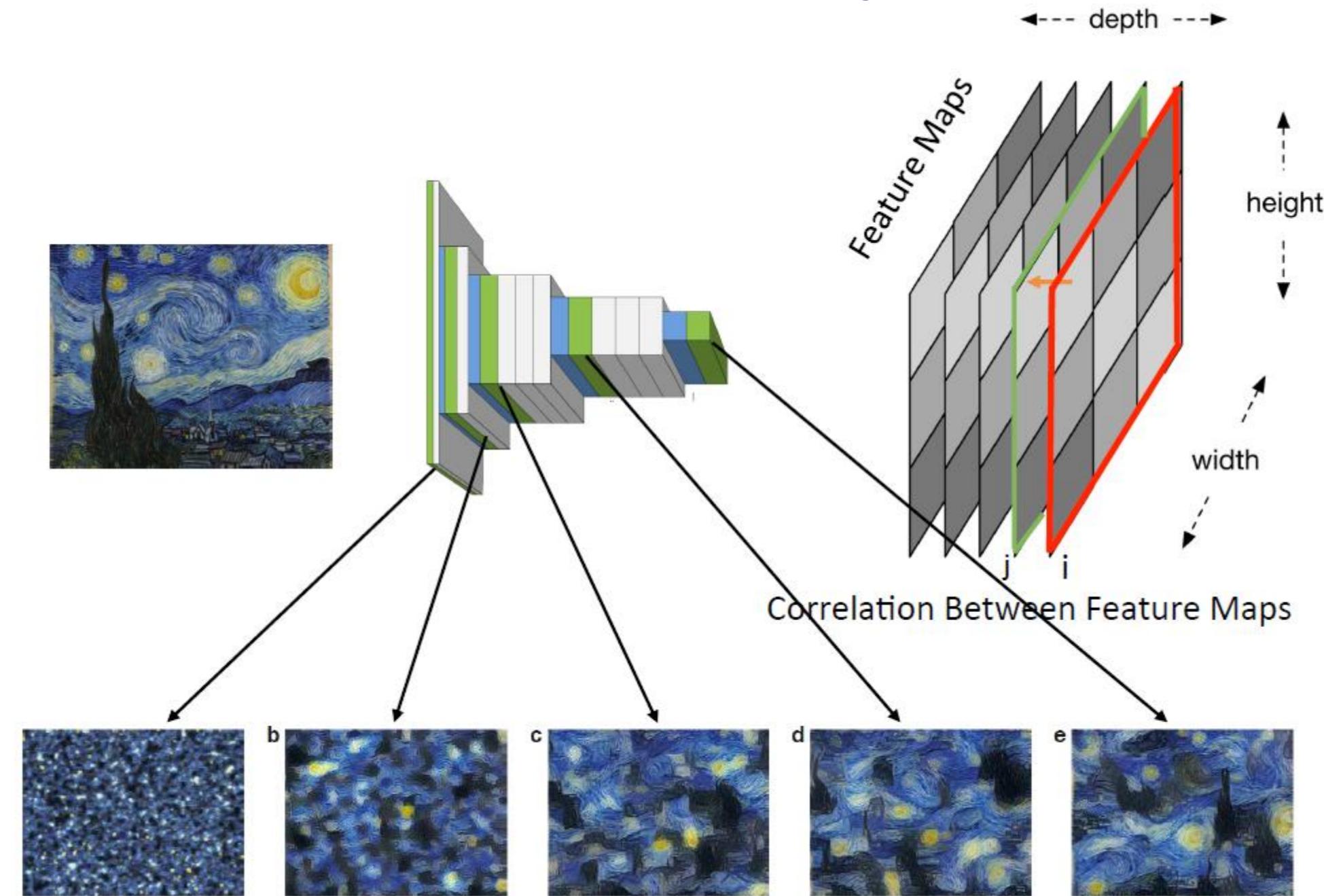
Что такое стиль / текстура?

Не должен зависеть от координат...

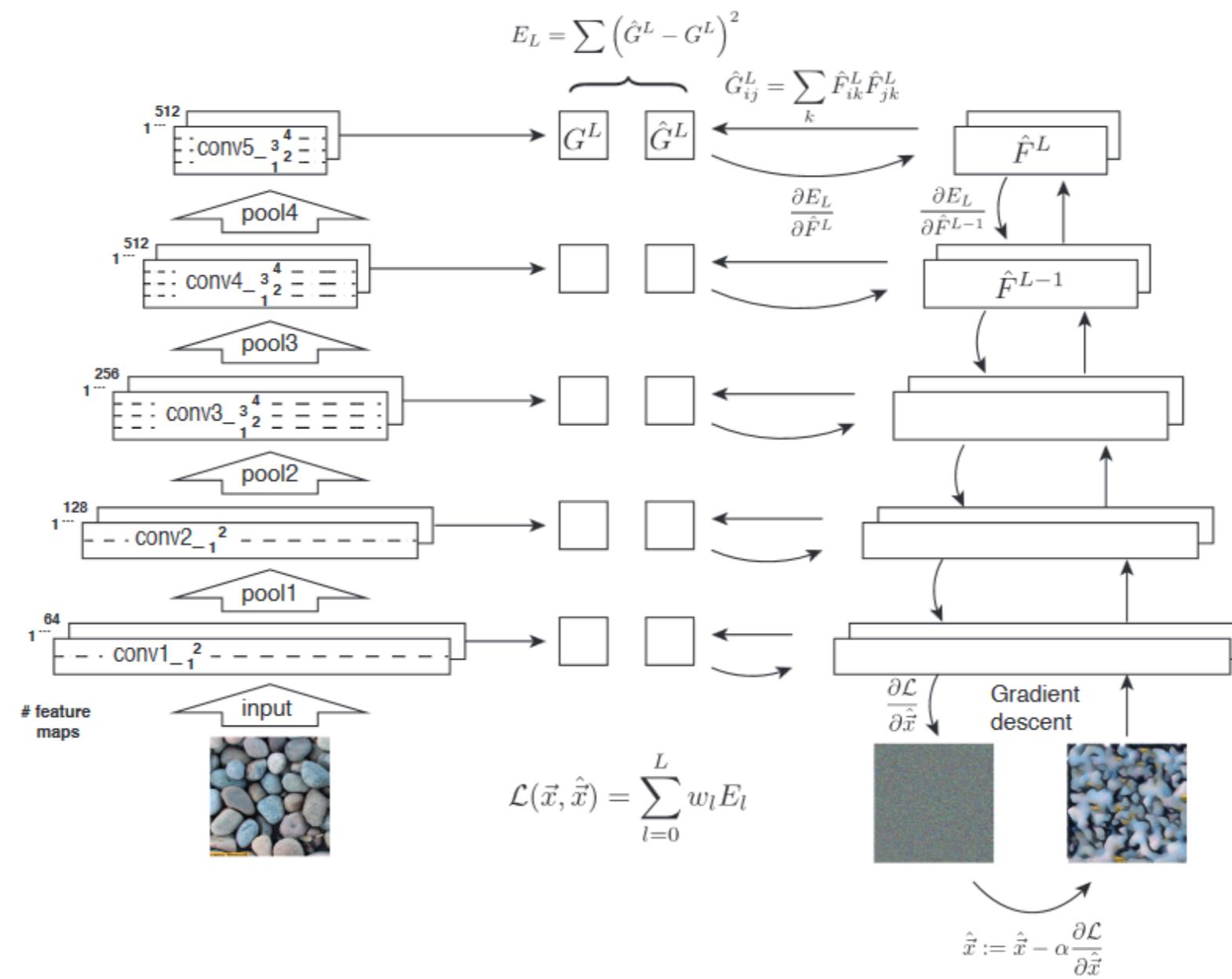
- 1) усредним признаки по пространственным измерениям;)**
- 2) сложнее – посчитаем ковариации (больше информации)**

Leon A. Gatys «Texture Synthesis Using Convolutional Neural Networks» <https://arxiv.org/pdf/1505.07376.pdf>

Что такое стиль / текстура?



Генерация текстур



Генерация текстур

Идея:

Берём какую-нибудь CNN, например VGG-19 (её свёрточную часть)

На этой части изображение ~ тензор $w \times h \times k$

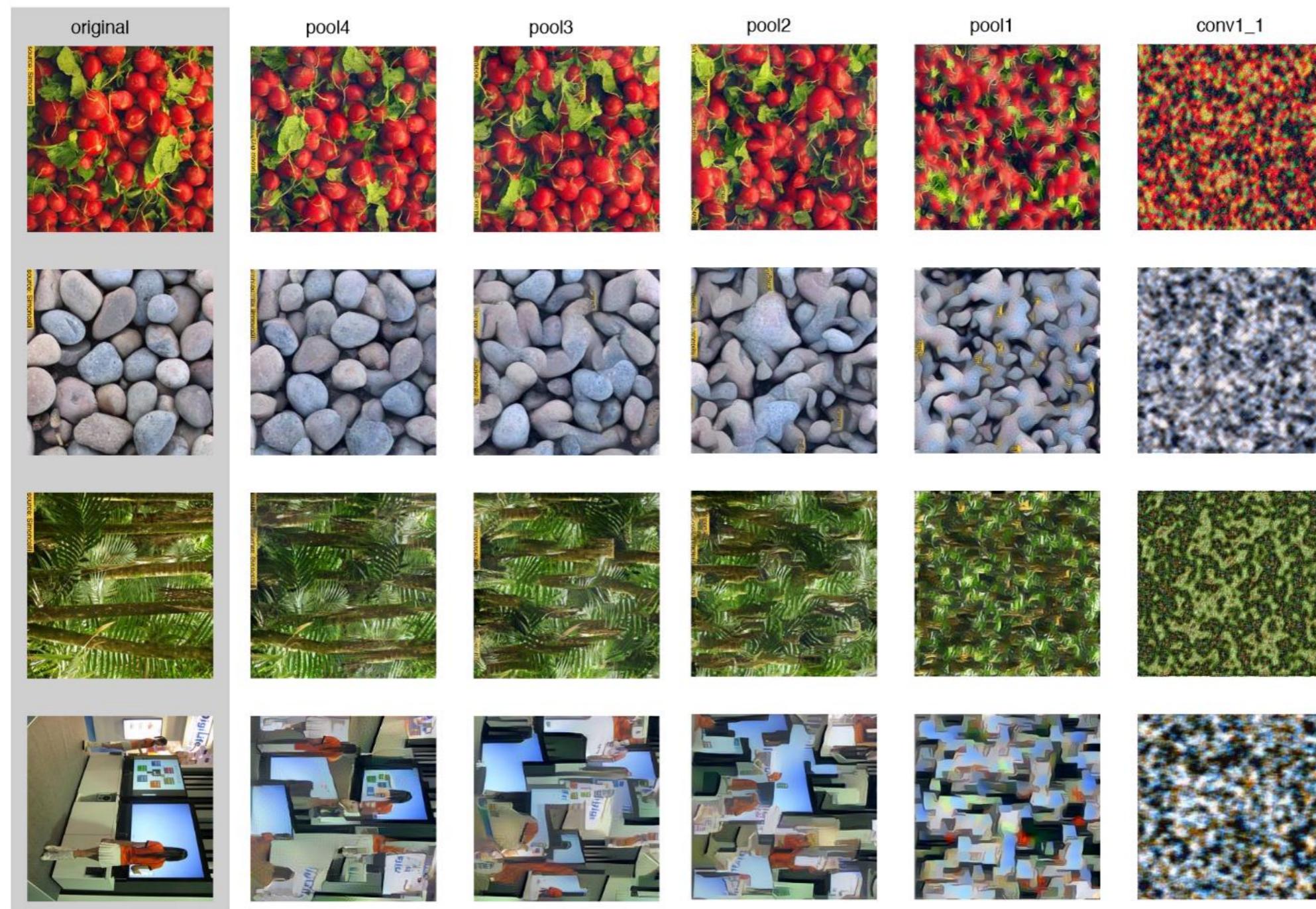
k – число каналов (м.б. 512)

Считаем матрицу Грама $k \times k$

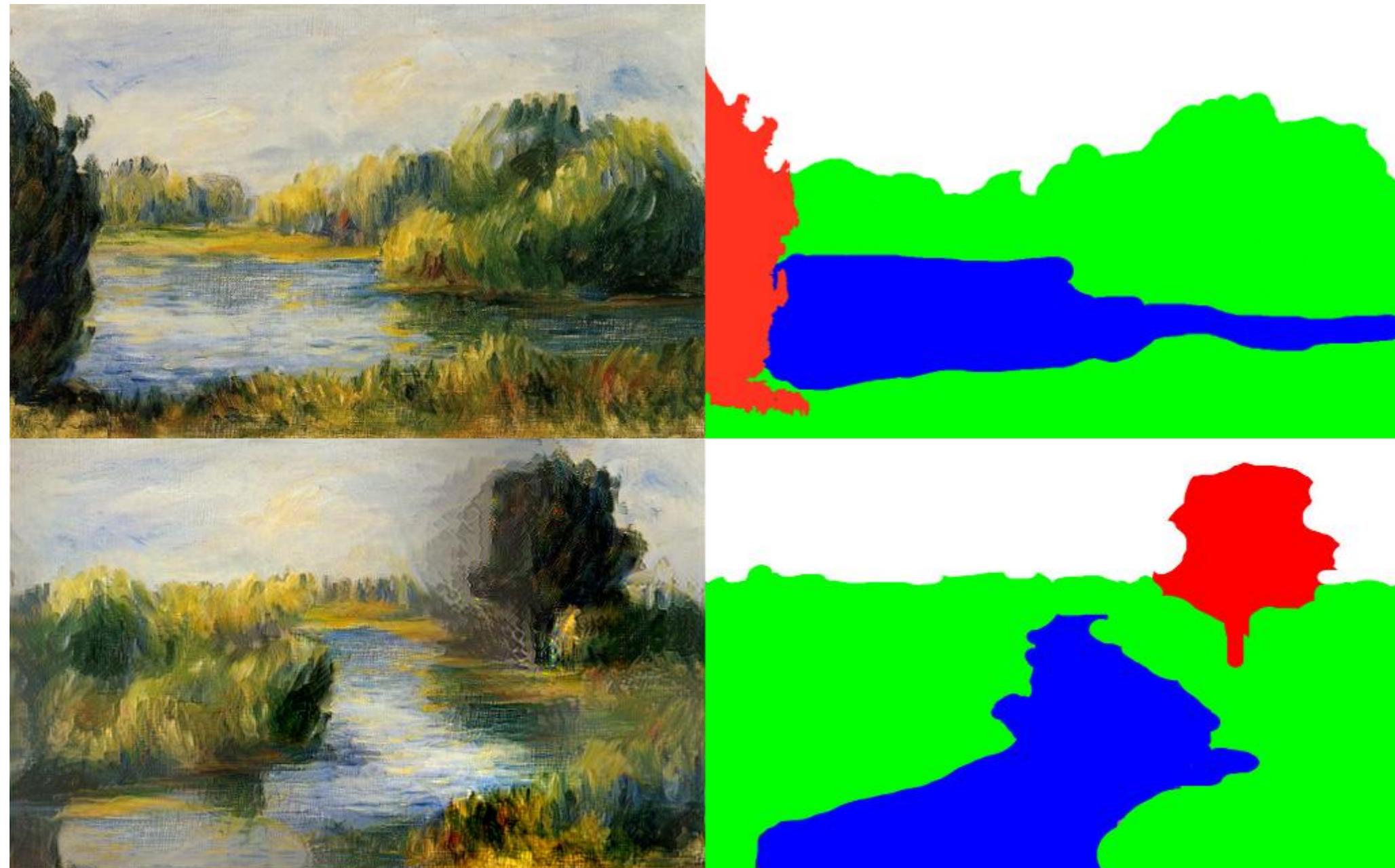
i, j -й элемент – корреляция wh -мерных векторизаций i -го канала и j -го

Вторая точно такая же сеть – на вход шум – тоже считаем матрицы Грама

Теперь обучаем вторую НС (обучаемые параметры – само изображение) так, чтобы её матрицы Грама были похожи на матрицы Грама первой НС



Генерация пейзажей



<https://github.com/DmitryUlyanov/fast-neural-doodle>

Генерация пейзажей

Вход: картина + сегментация + новая сегментация

**Для каждого сегмента получаем описание стиля (матрицы Грама разных слоёв) и
генерируем новое изображение**

Стилизация (перенос стиля)

**Идея: похожесть на фотографию + похожесть на стиль рисунка
= похожесть признакового описания + похожесть матриц Грама**

Leon A. Gatys и др. «A Neural Algorithm of Artistic Style», 2015 //
<https://arxiv.org/pdf/1508.06576.pdf>

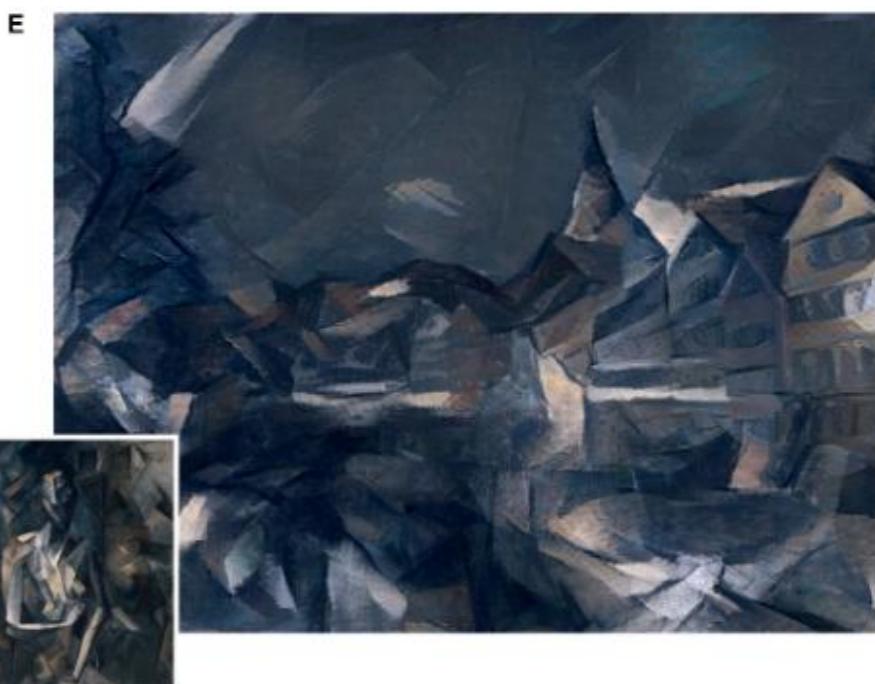
A



B



E



C



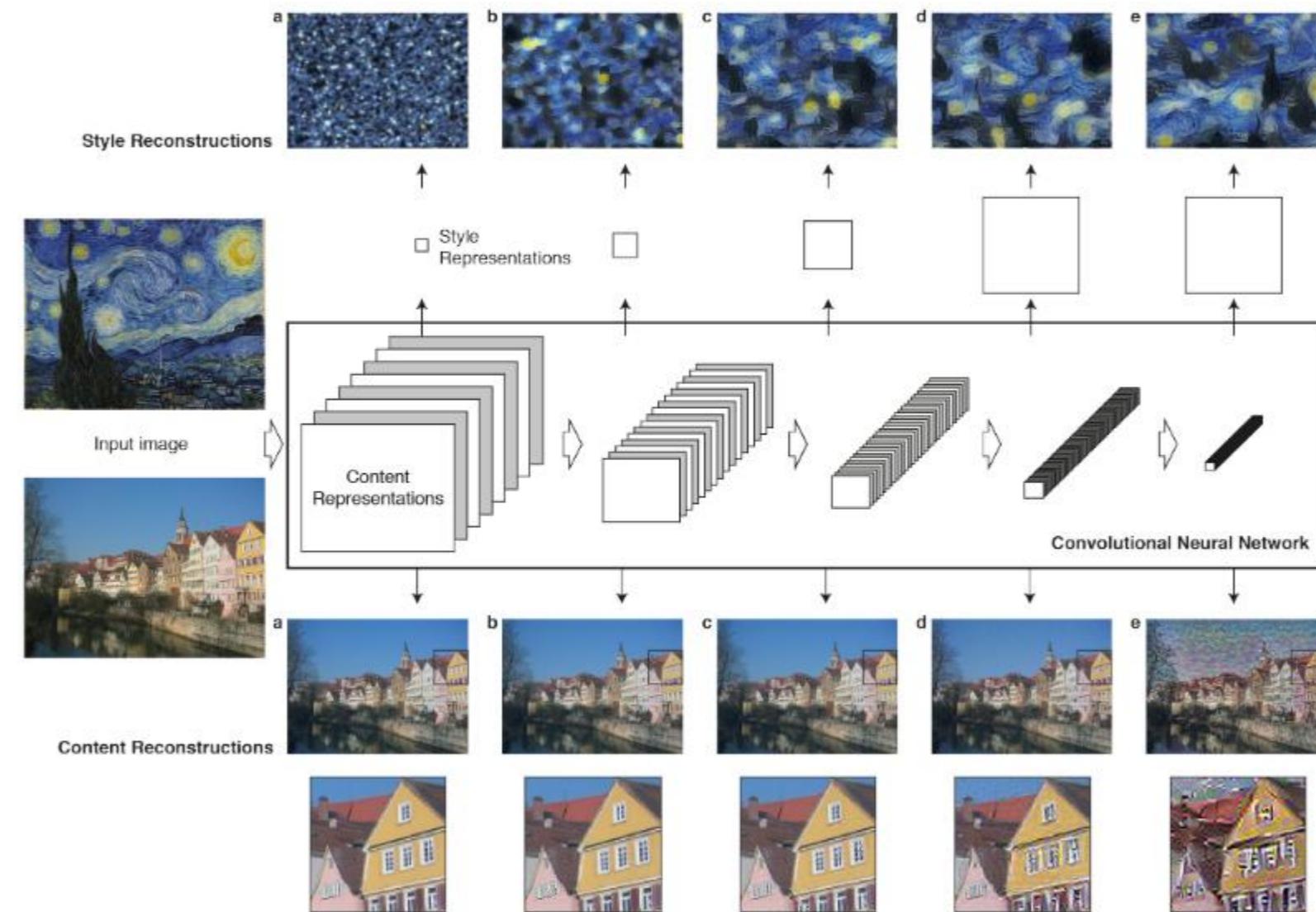
D



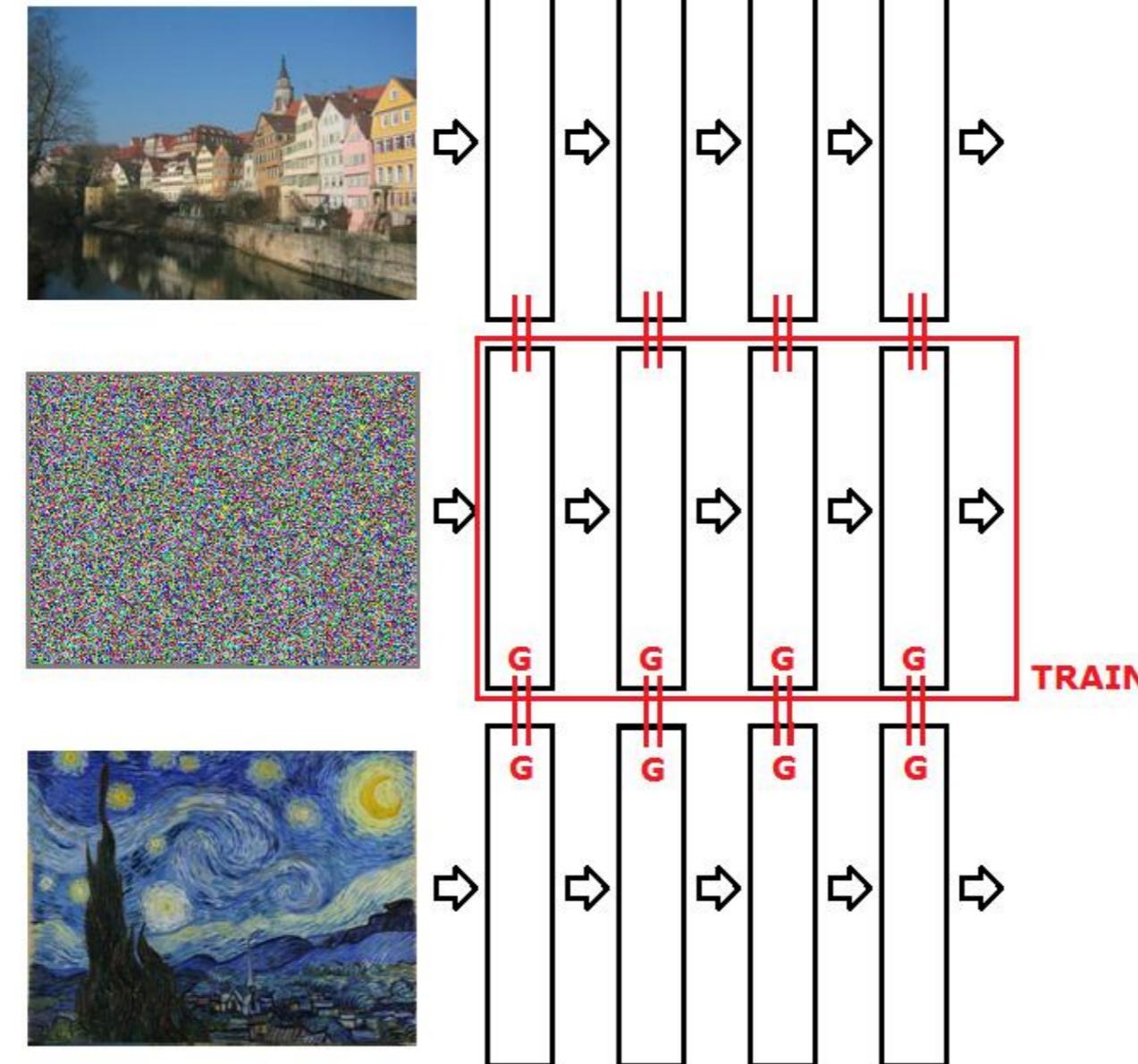
F



Стилизация (перенос стиля)

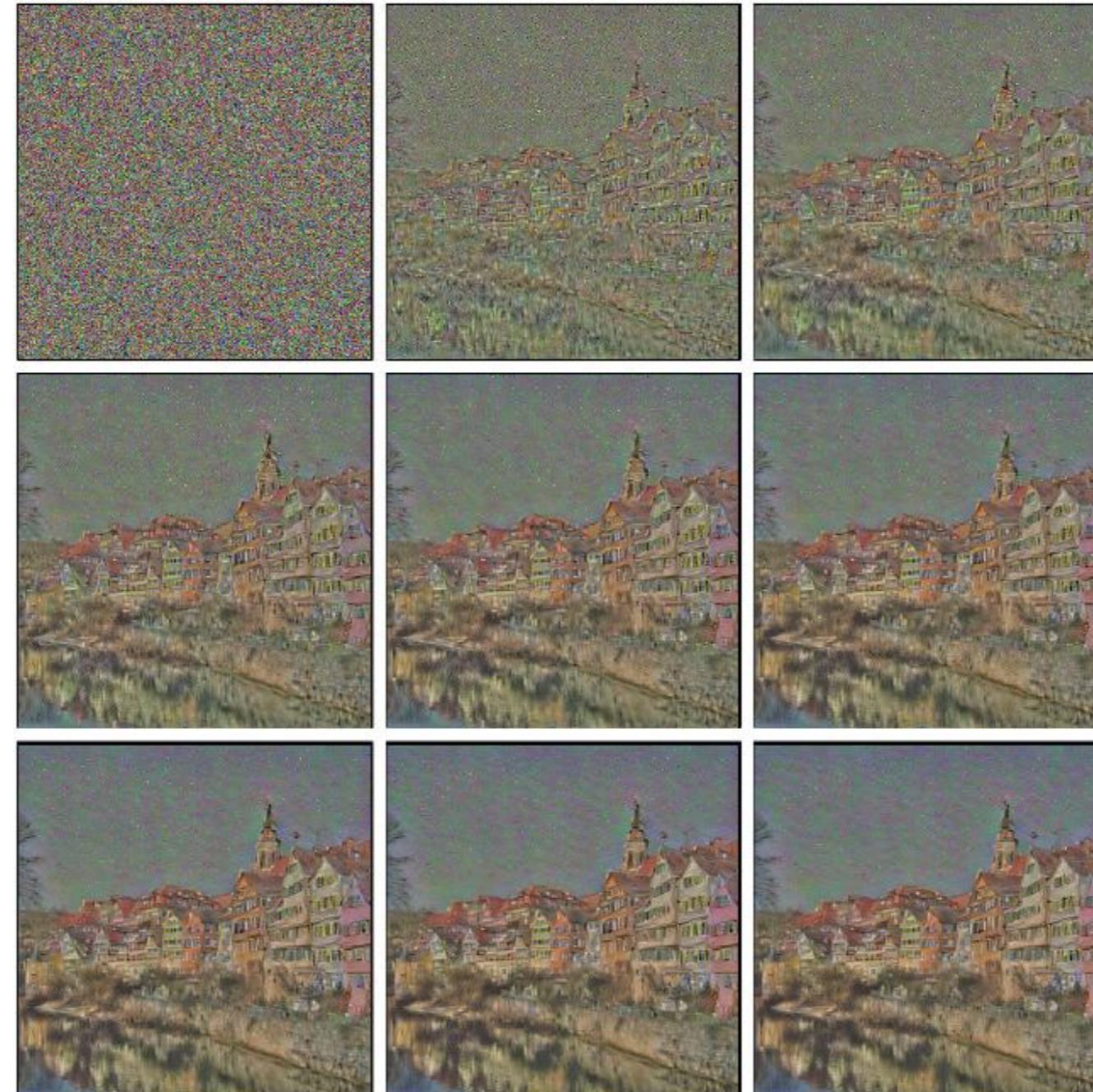


Стилизация (перенос стиля)

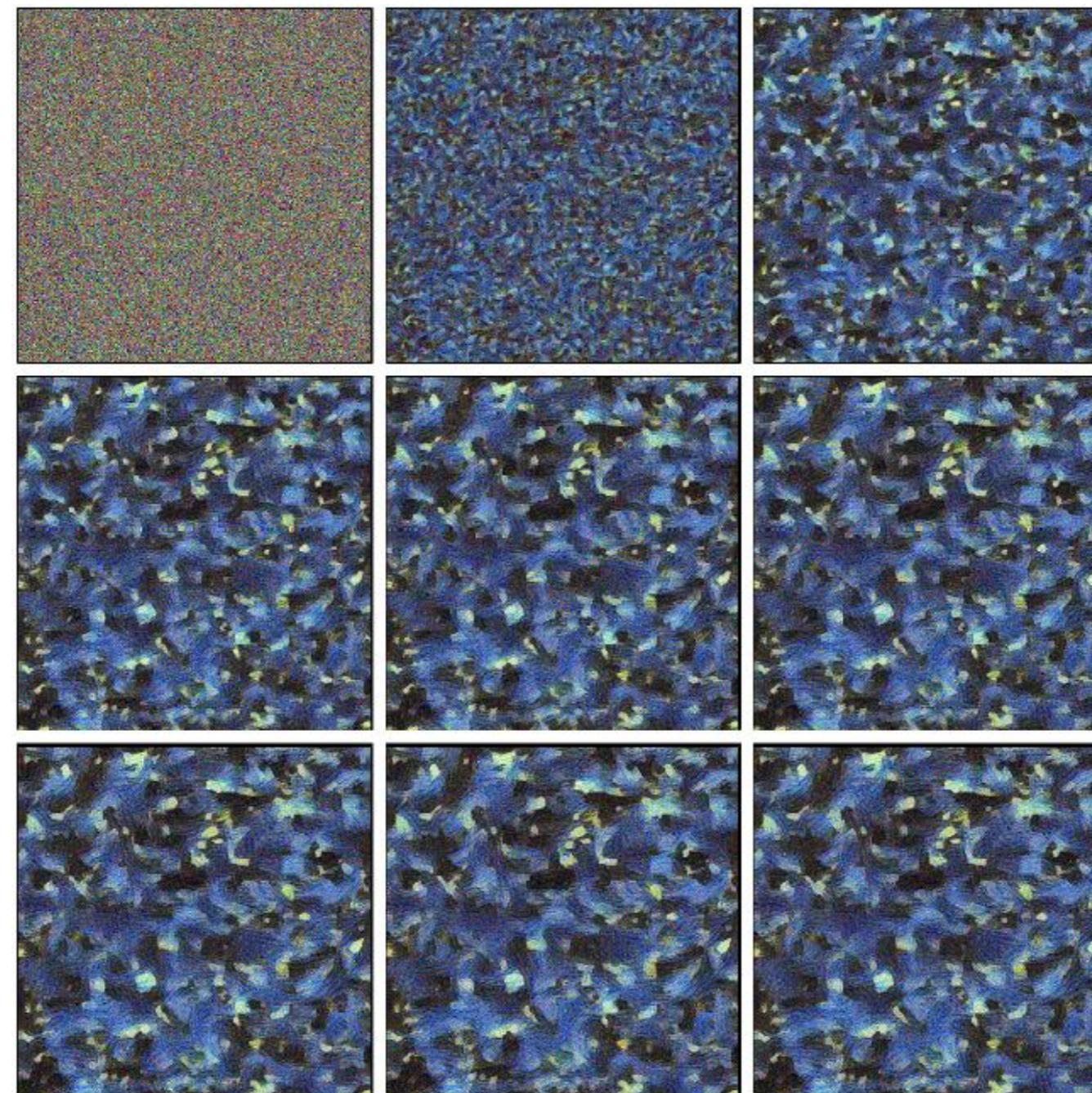


на самом деле с равенствами не совсем так...

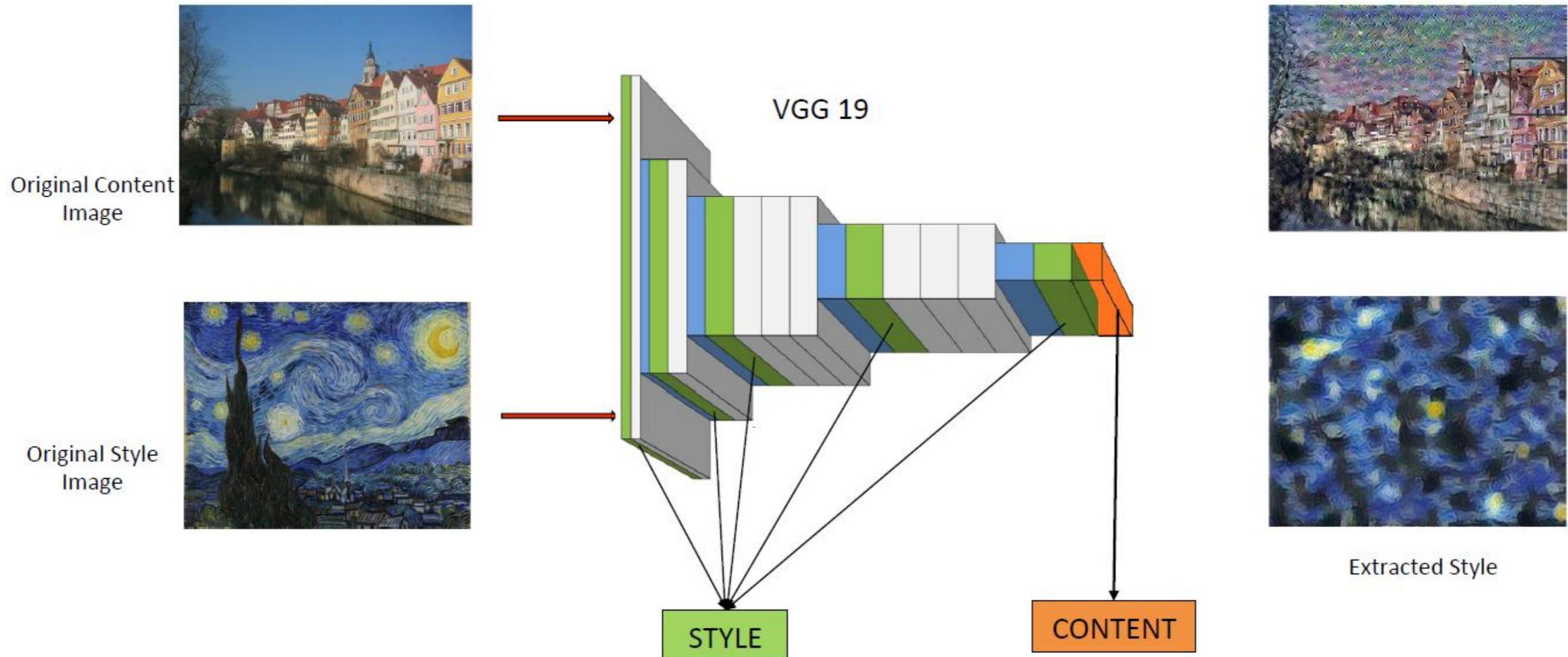
Оптимизация по содержанию



Оптимизация по стилю



Стилизация



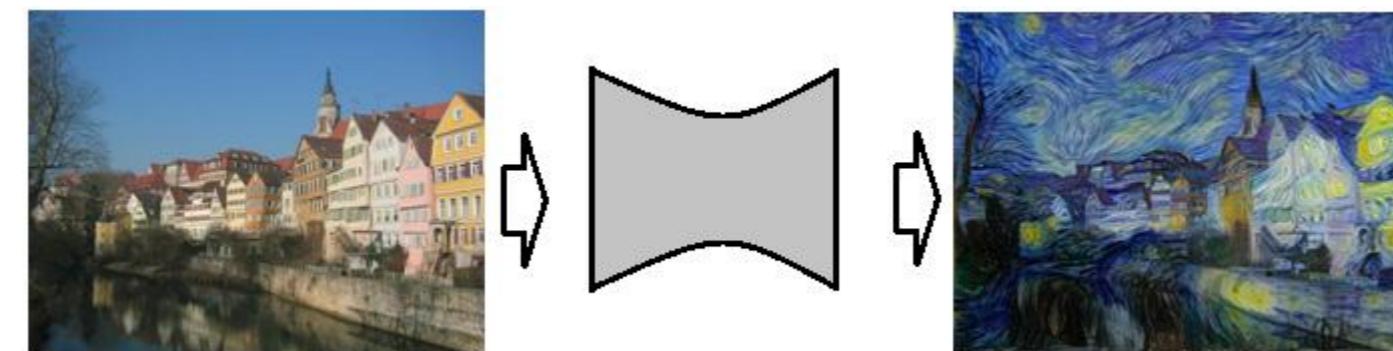
ошибка = л/к ошибок за контент и стиль



Figure 3: Detailed results for the style of the painting *Composition VII* by Wassily Kandinsky. The rows show the result of matching the style representation of increasing subsets of the CNN layers (see Methods). We find that the local image structures captured by the style representation increase in size and complexity when including style features from higher layers of the network. This can be explained by the increasing receptive field sizes and feature complexity along the network's processing hierarchy. The columns show different relative weightings between the content and style reconstruction. The number above each column indicates the ratio α/β between the emphasis on matching the content of the photograph and the style of the artwork (see Methods).

Быстрая стилизация

раньше ~5 минут



**пусть будет всего одна сеть,
но она умеет делать конкретную стилизацию!**

Кстати,

- **зашумлять изображения, чтобы была устойчива к шуму**

[Дмитрий Ульянов]

Дальше – GAN

Быстрая стилизация

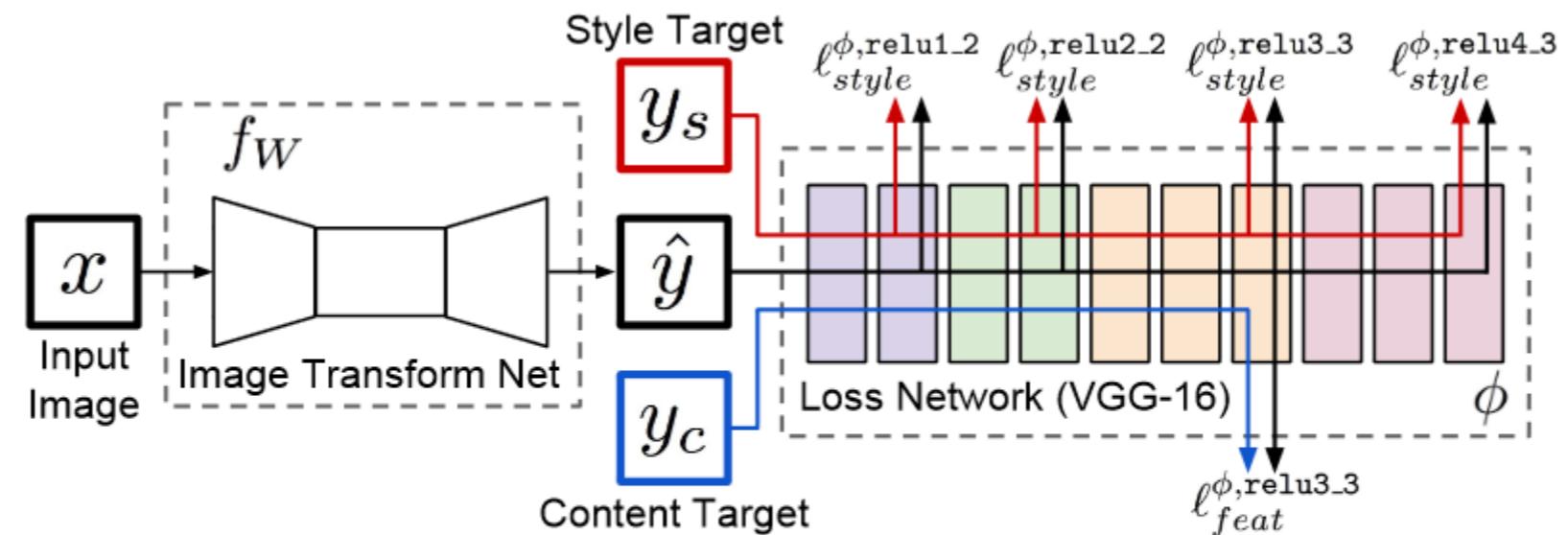


Fig. 2. System overview. We train an *image transformation network* to transform input images into output images. We use a *loss network* pretrained for image classification to define *perceptual loss functions* that measure perceptual differences in content and style between images. The loss network remains fixed during the training process.

Justin Johnson, Alexandre Alahi, Li Fei-Fei
«Perceptual Losses for Real-Time Style Transfer and Super-Resolution»
<https://arxiv.org/pdf/1603.08155.pdf>

Итог

**С помощью визуализации можно понять,
что и как делают НС
но нет идеального и универсального способа
хорошее средство контроля НС**

**В визуализации много интересных результатов
и много неожиданных**

**Взятие производных – мощный инструмент
можно брать и по входу**

**Одна из формализаций стиля ~ матрица Грама
но есть и другие (далее в GAN)**

**Интересно, как задача с текстурами привела
к целому направлению исследований**

Ссылки

Было много выше...

Интерфейс для визуализации и интерпретации

<https://distill.pub/2018/building-blocks/>

Лекция Дмитрия Ульянова

<https://www.youtube.com/watch?v=Y61Q9Y9pKIs>