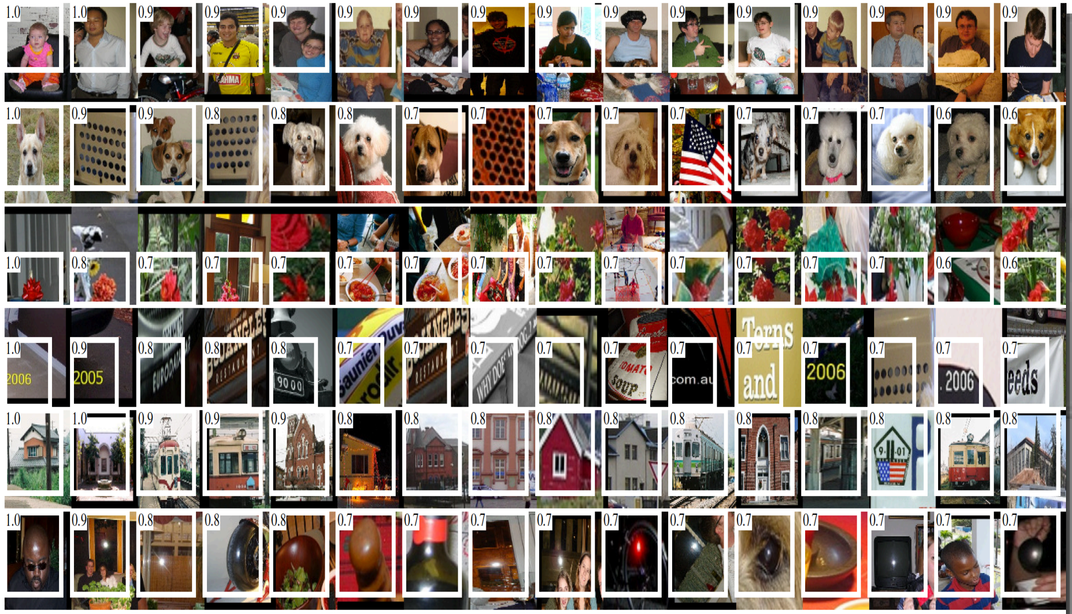
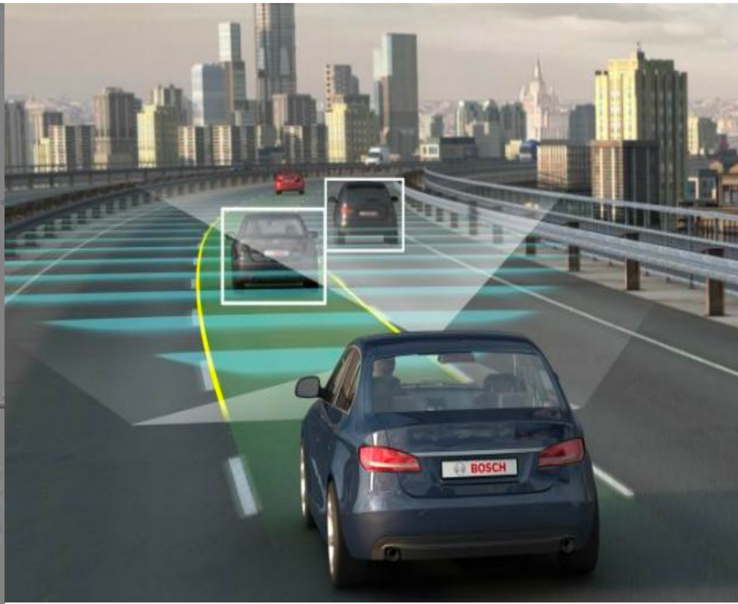
Visualization and Adversarial Examples

Jyoti Aneja, Ralf Gunter Correa Carvalho, Jiahui Yu

CS-598LAZ





Today's Talk

Visualization

1. What is Visualization?
2. Visualize patches that maximally activate neurons
3. Visualize the weights
4. Gradient based approaches
5. Optimization based approach

Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

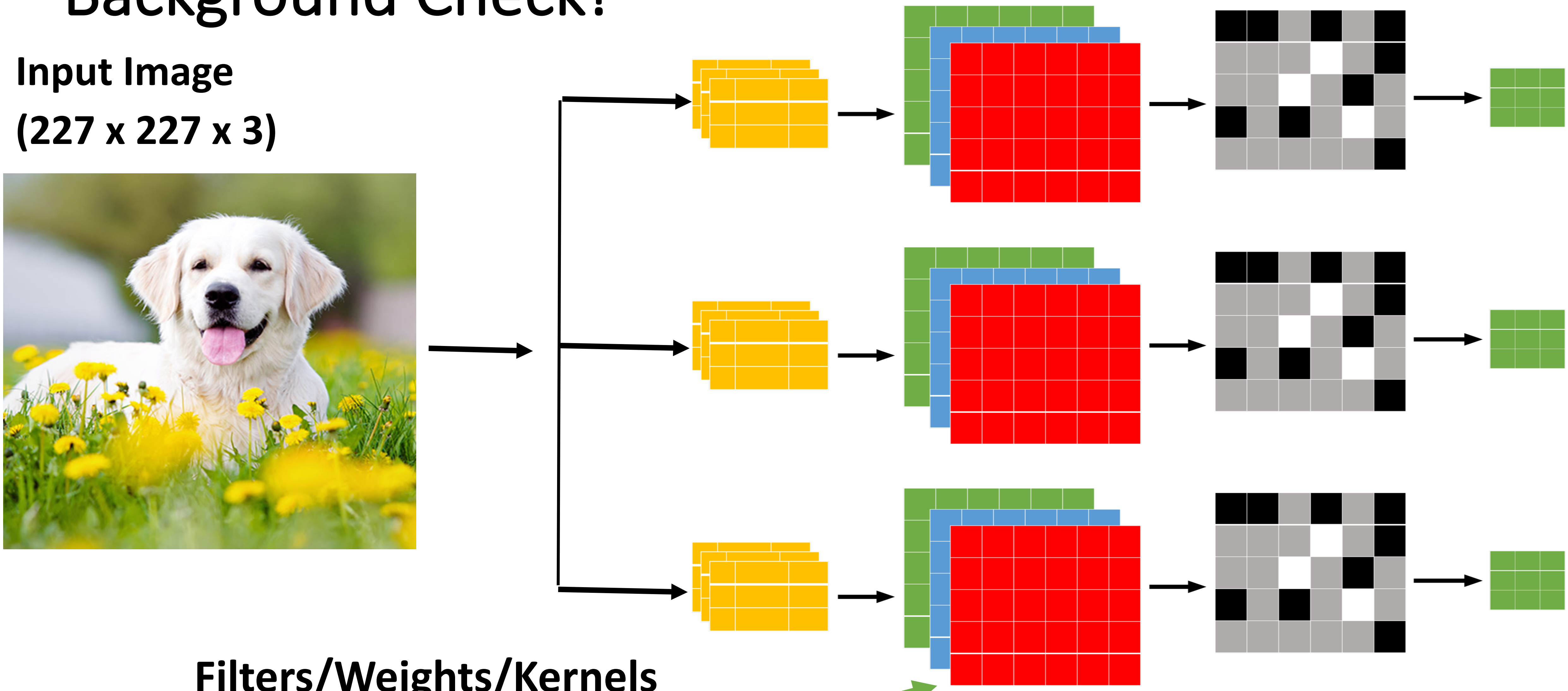
Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach

What is visualization?

Mapping between a neuron in a layer to the features in the image.

Background Check!



What is visualization?

Mapping between a neuron in a layer to the features in the original image.

Backpropagation : How does the loss change with weights?

Visualization : How does the activation of a particular neuron change when we change a part in the image?

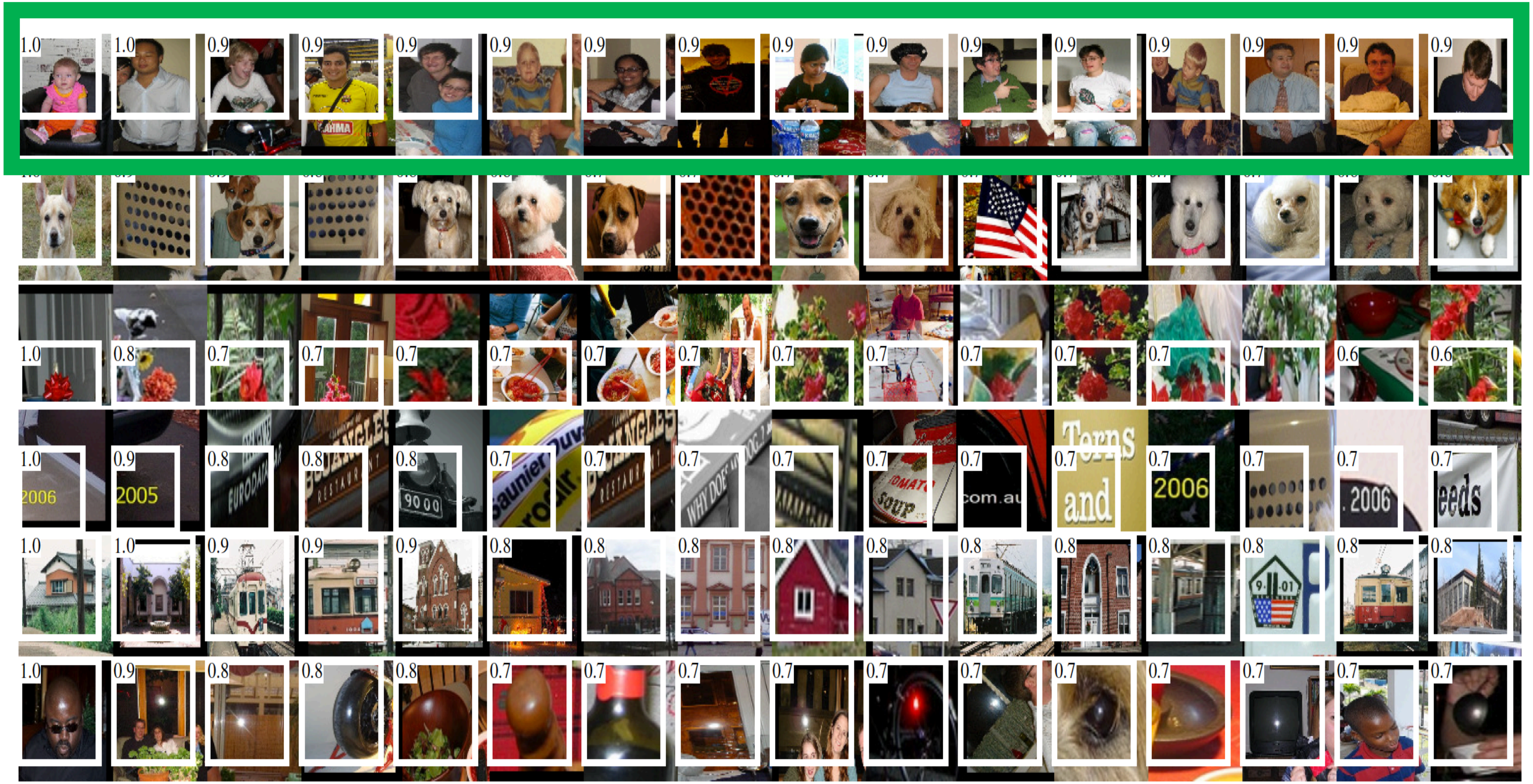
Why visualization?

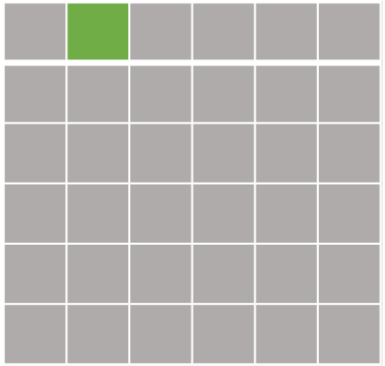
- Understand how and why neural networks work
- Observe the evolution of features during training
- Aid the development of better models (rather than just trial-and-error)
- Diagnose potential problems with the model

Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach

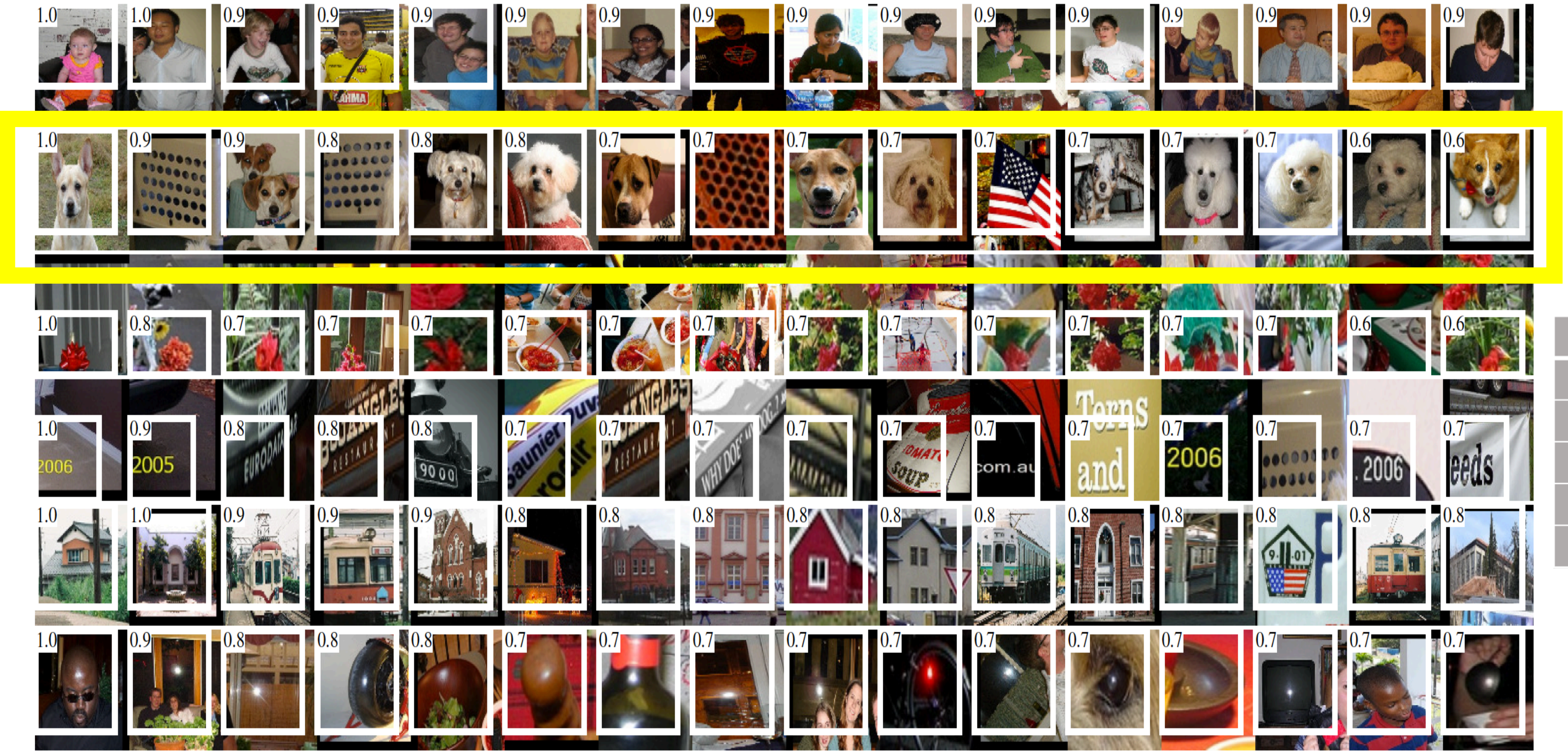
Visualize patches that maximally activate neurons

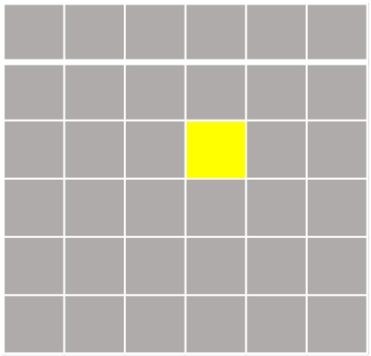




Rich feature hierarchies for accurate object detection and semantic segmentation – Girshick, et al - 2013

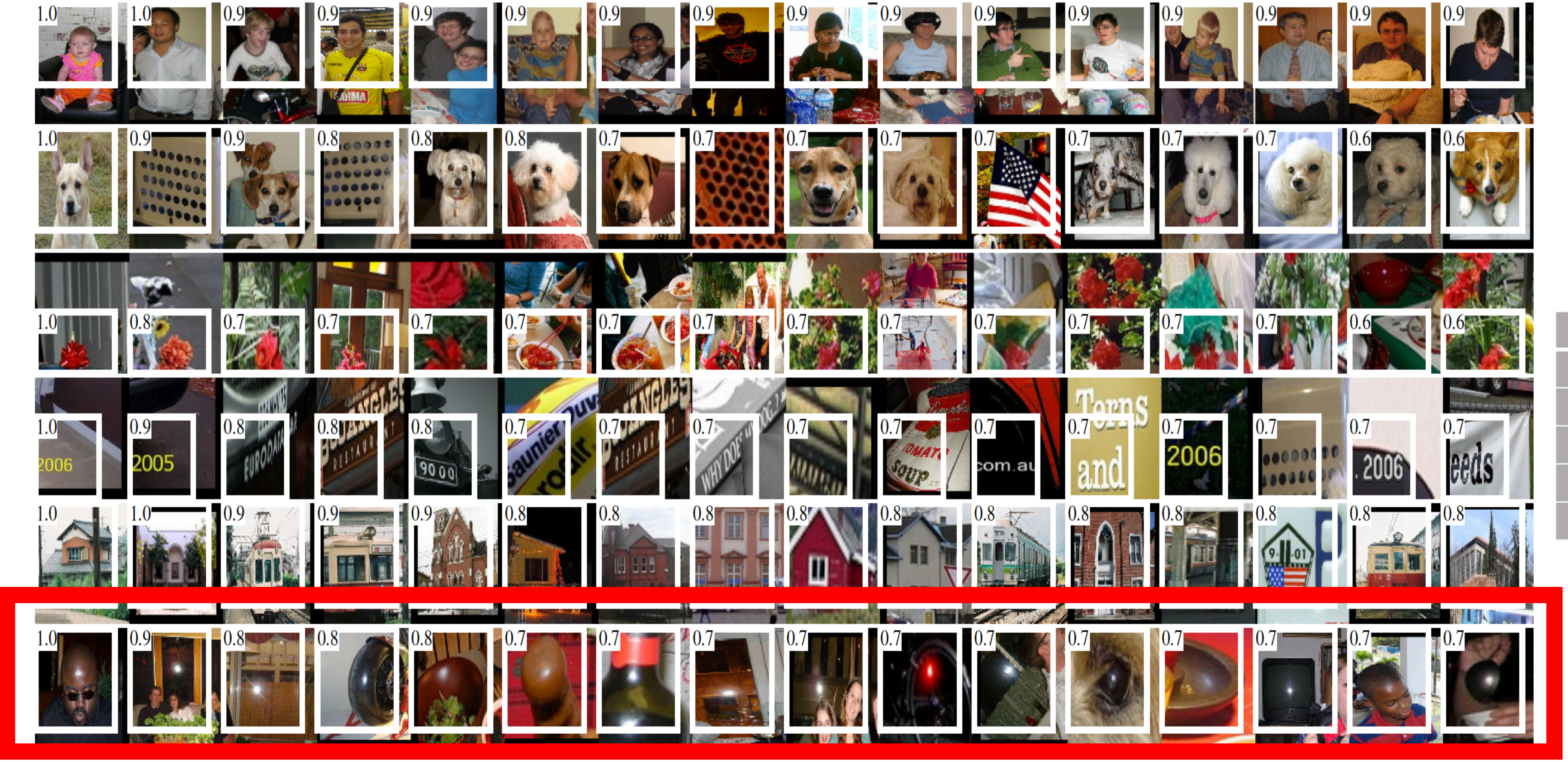
Visualize patches that maximally activate neurons

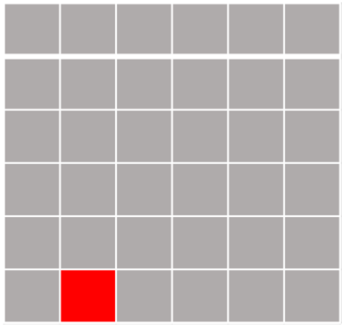




Rich feature hierarchies for accurate object detection and semantic segmentation – Girshick, et al - 2013

Visualize patches that maximally activate neurons



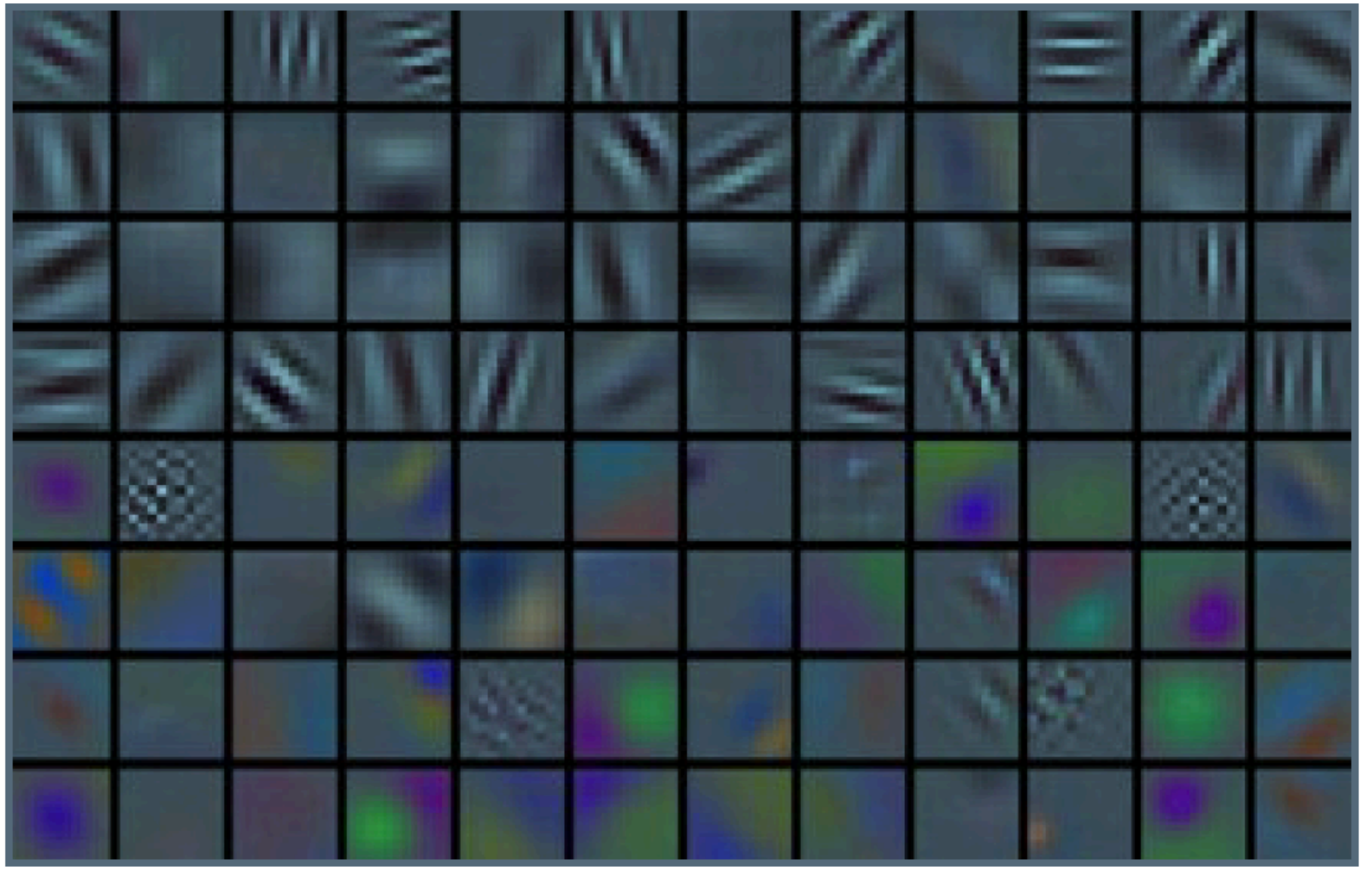


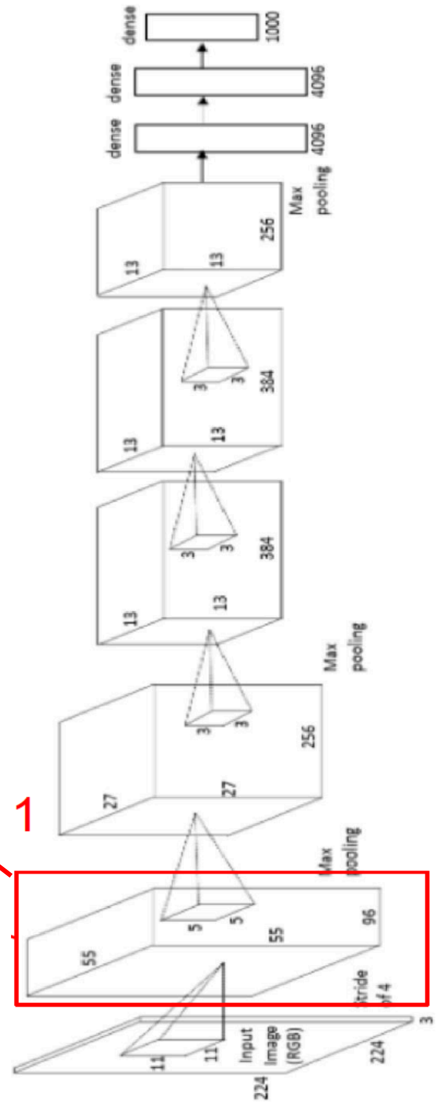
Rich feature hierarchies for accurate object detection and semantic segmentation – Girshick, et al - 2013

Outline - Visualization

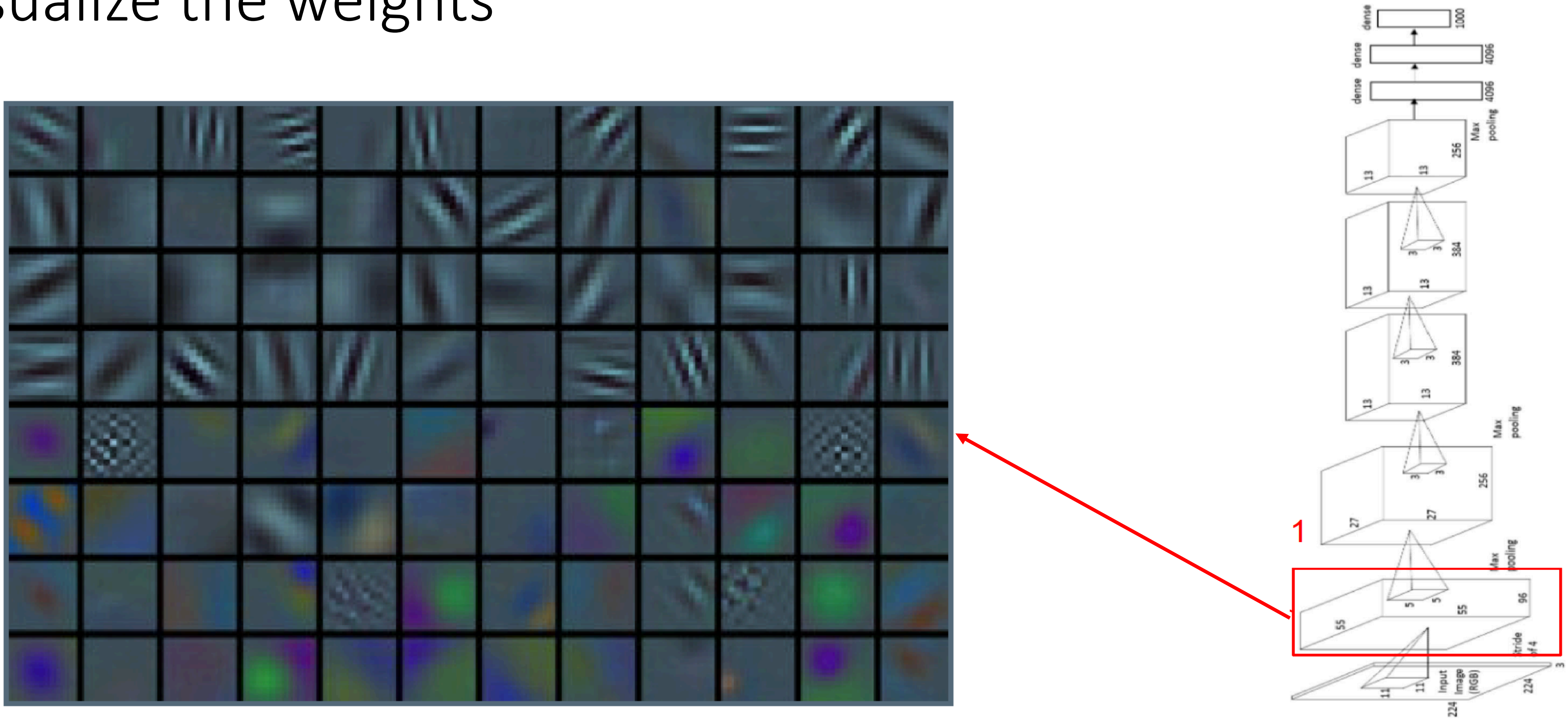
- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach

Visualize the weights





Visualize the weights



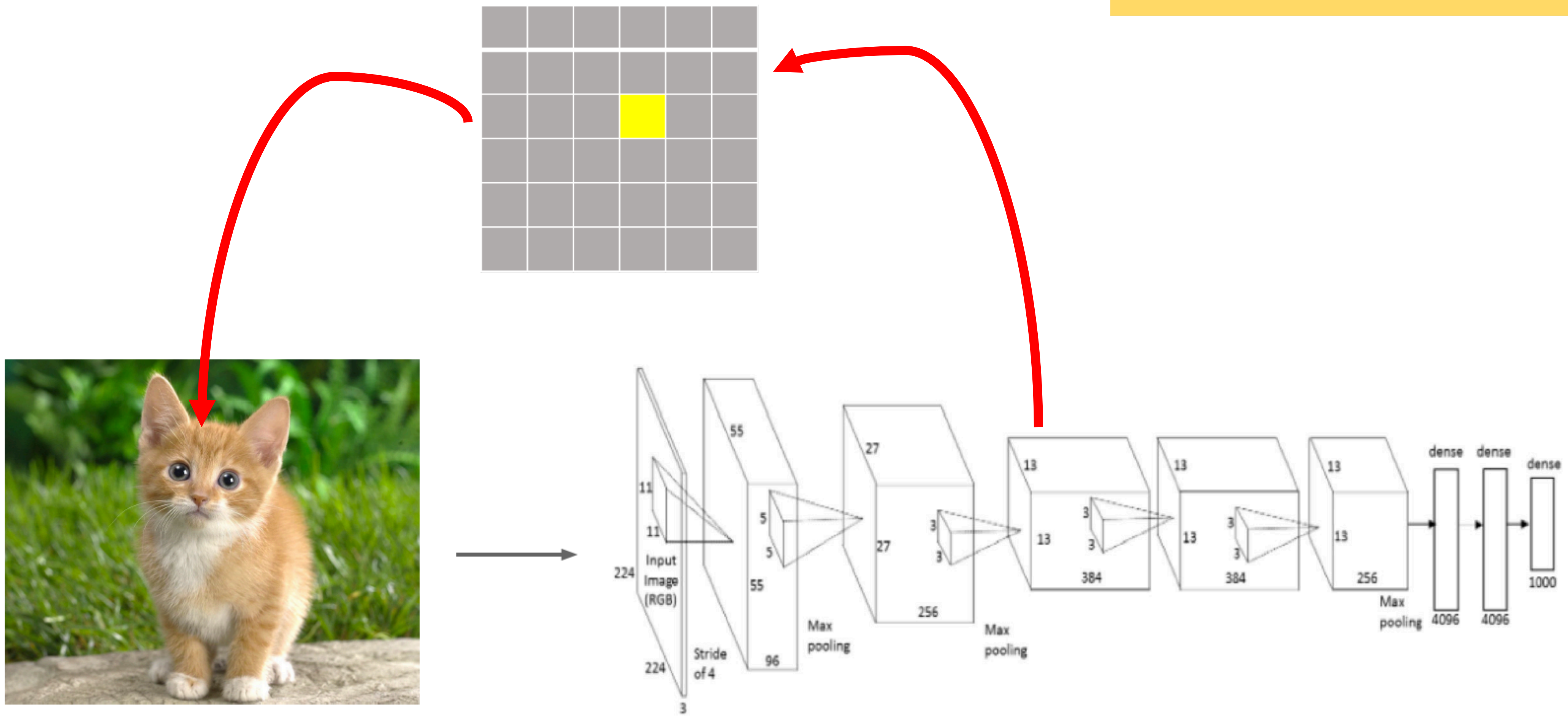
Only possible for the first layer ☹

Outline - Visualization

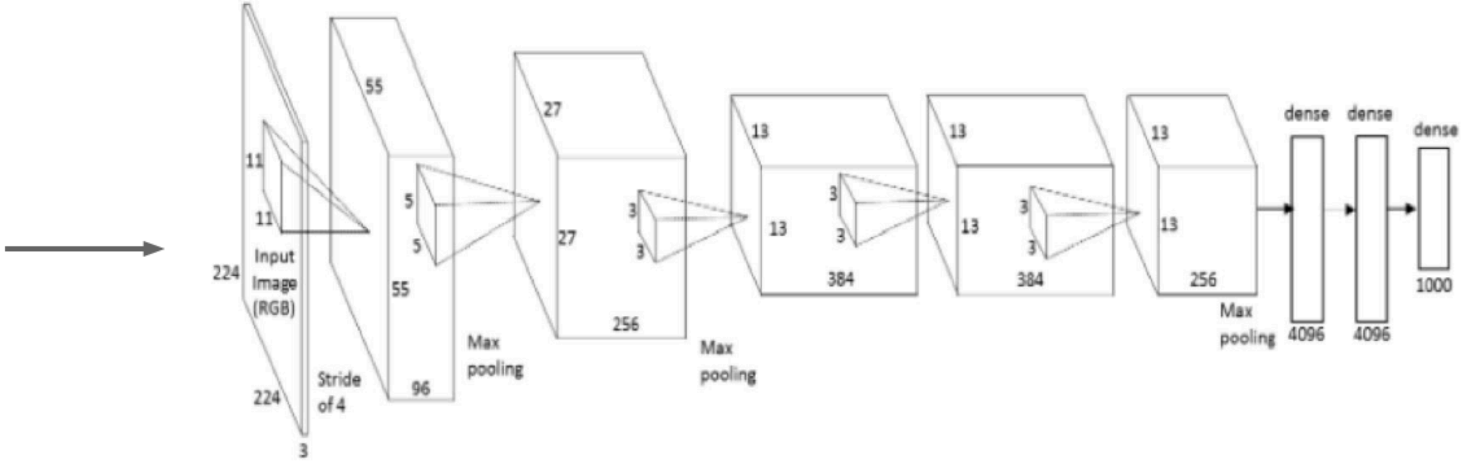
- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach

Gradient based approaches

Q : How can we compute the gradient of an arbitrary neuron w.r.t the image?

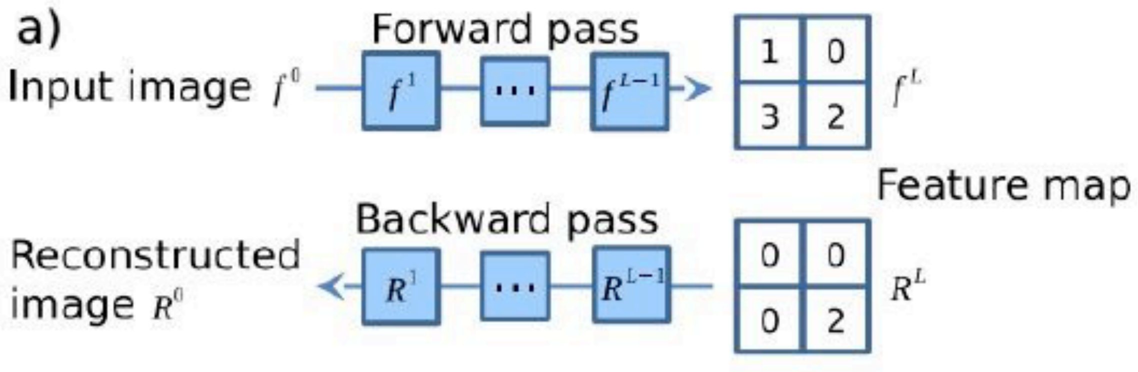


Gradient based approaches



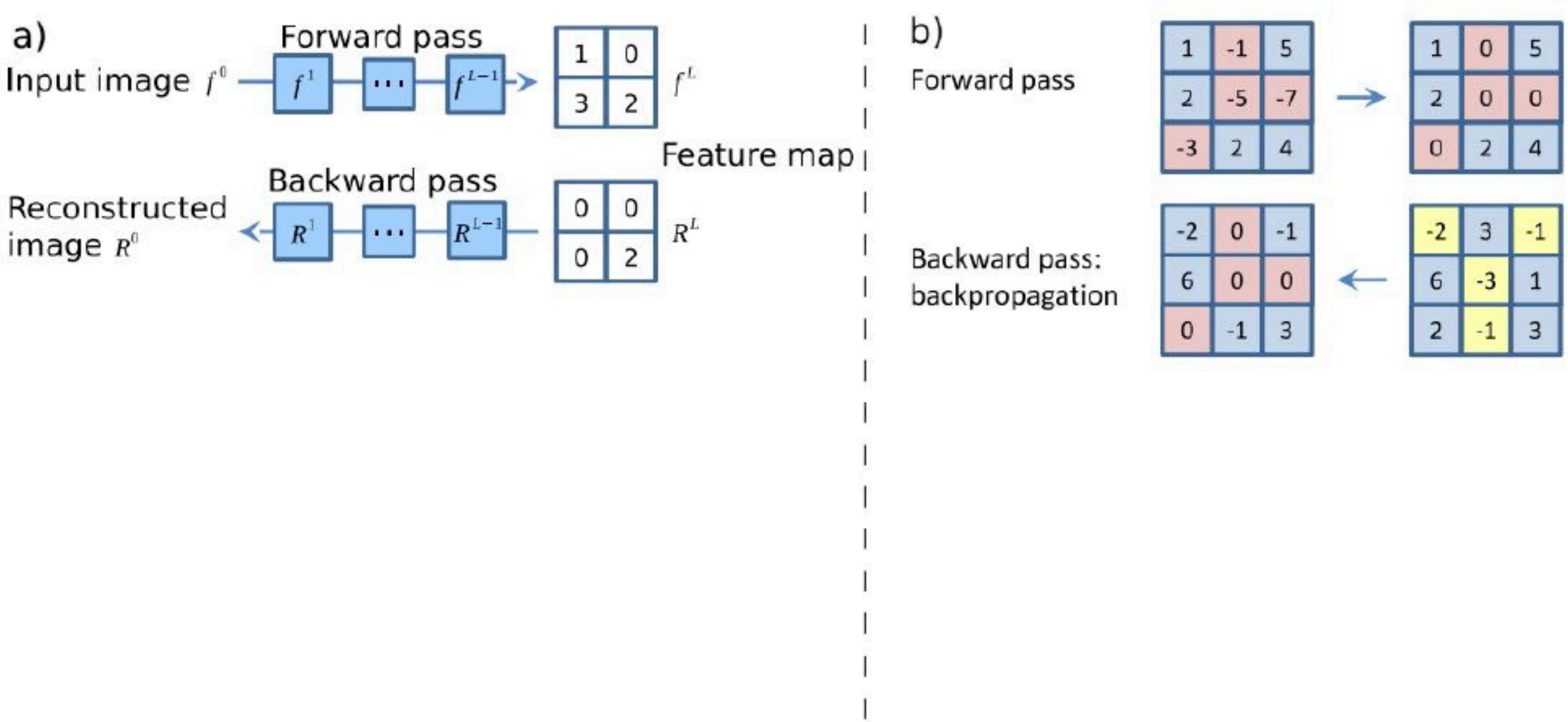
1. Input the image into the net
2. Pick a layer, set the gradient there to be all 0 except for one 1 for some neuron of interest
3. “Map it” back to the image

Gradient based approaches - "Map back"



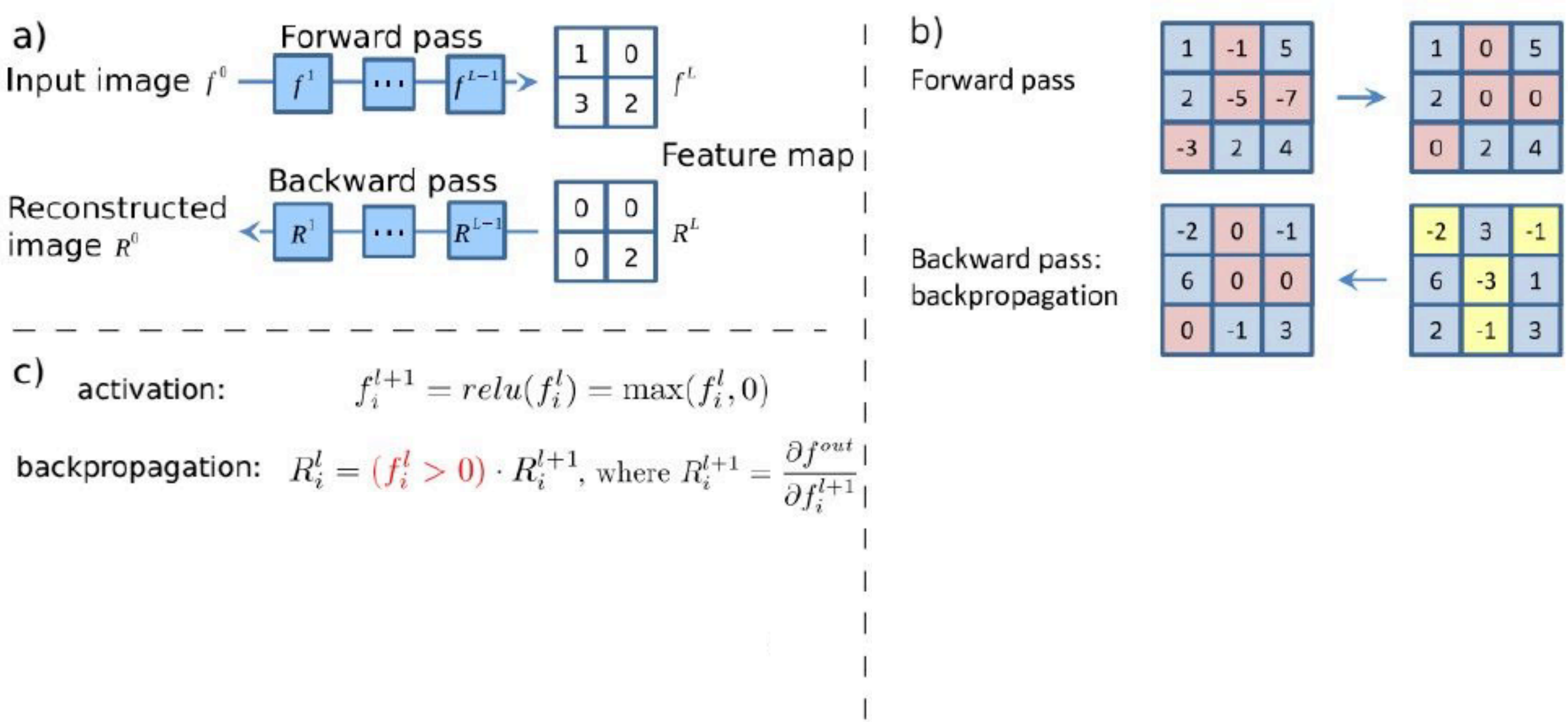
Striving for Simplicity: The all convolutional net - Springenberg, et al. - 2015

Gradient based approaches - “Map back”



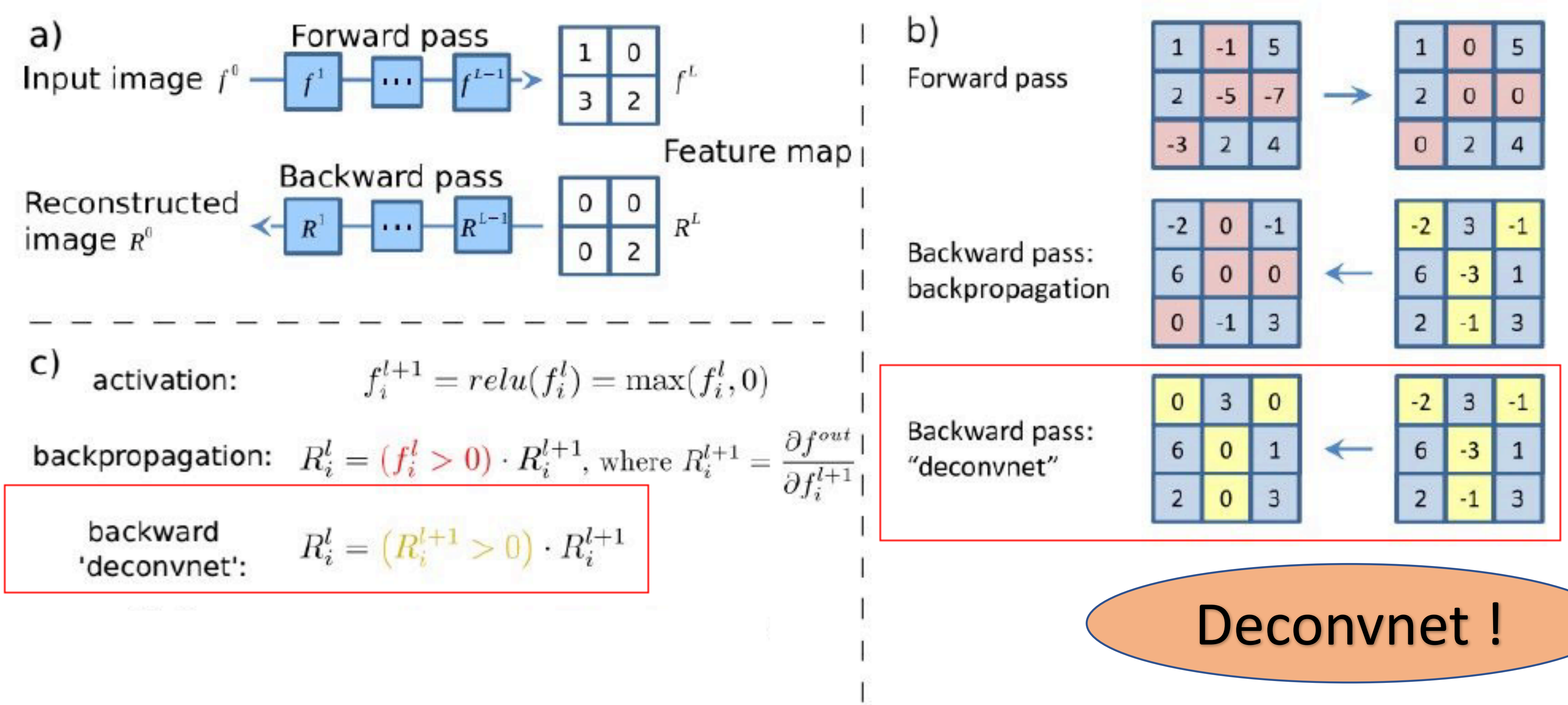
Striving for Simplicity: The all convolutional net - Springenberg, et al. - 2015

Gradient based approaches - “Map back”



Striving for Simplicity: The all convolutional net - Springenberg, et al. - 2015

Gradient based approaches - “Map back”



Deconvnet !

Striving for Simplicity: The all convolutional net - Springenberg, et al. - 2015

Visualizing the neurons along the way to the top

Choose a target neuron

↓

Input the images 1 by 1

↓

Select the top 9 images that have the highest activation for that neuron

↓

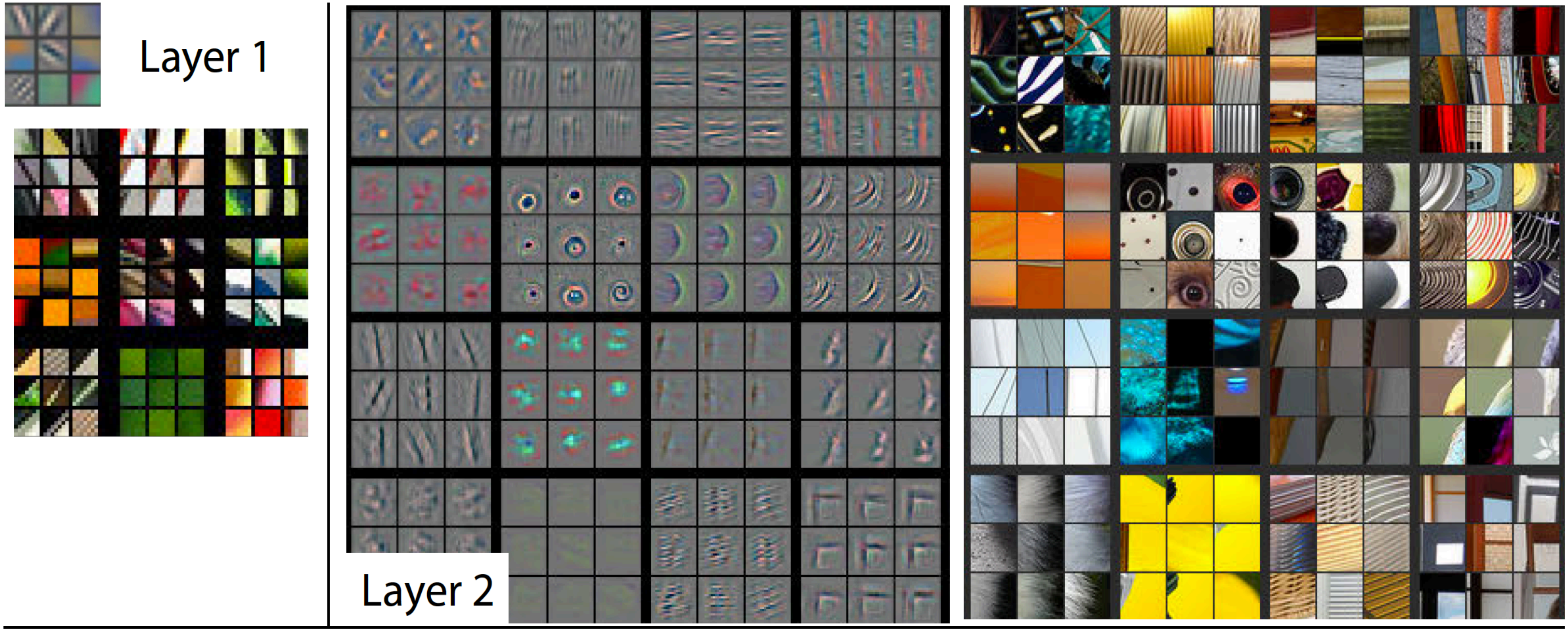
Cluster those images together

↓

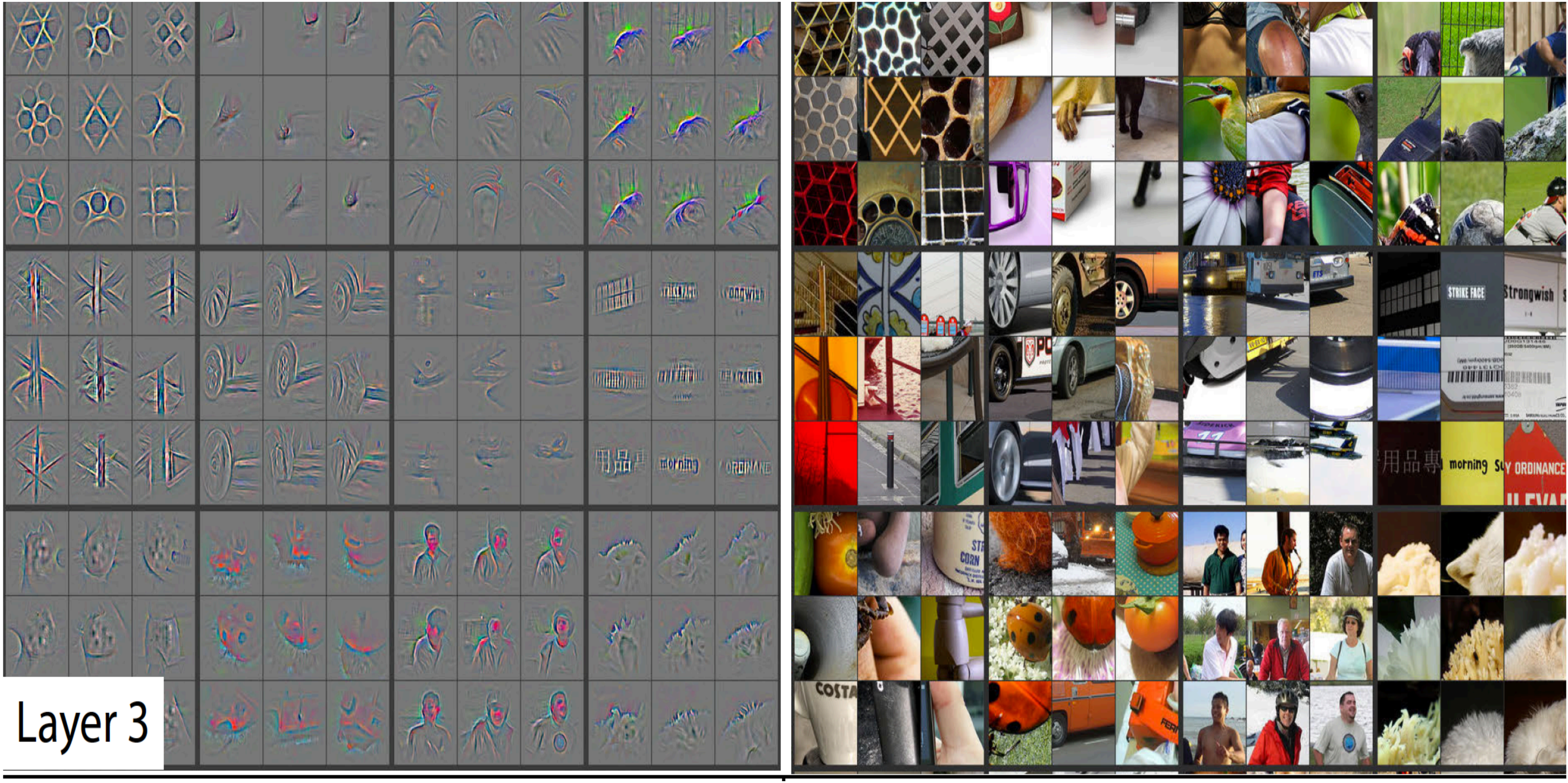
Map back from that neuron and create a "back-pass map"

Matthew D. Zeiler, Rob Fergus [Visualizing and Understanding Convolutional Networks](), ECCV 2014

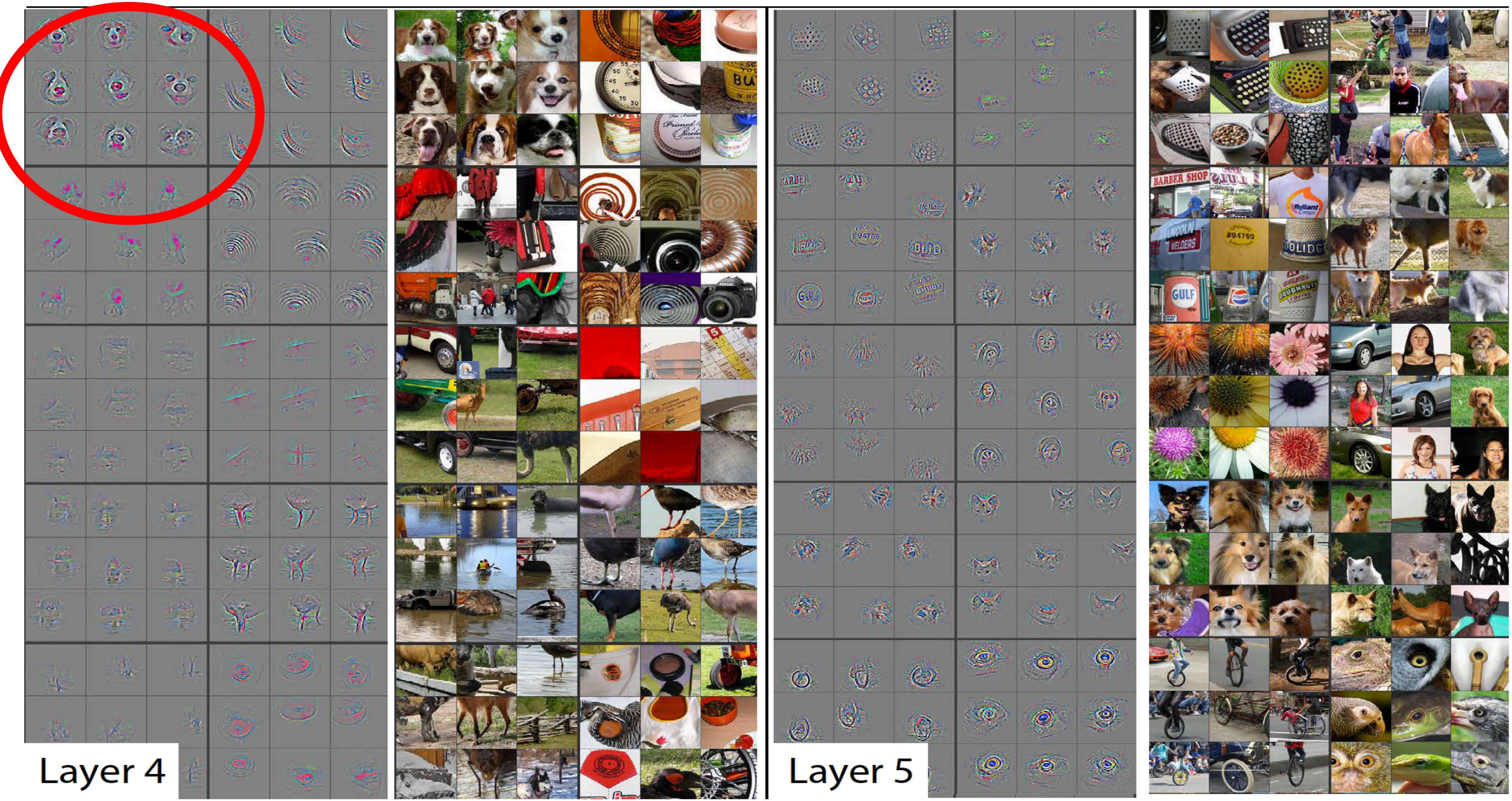
Visualizing the neurons along the way to the top



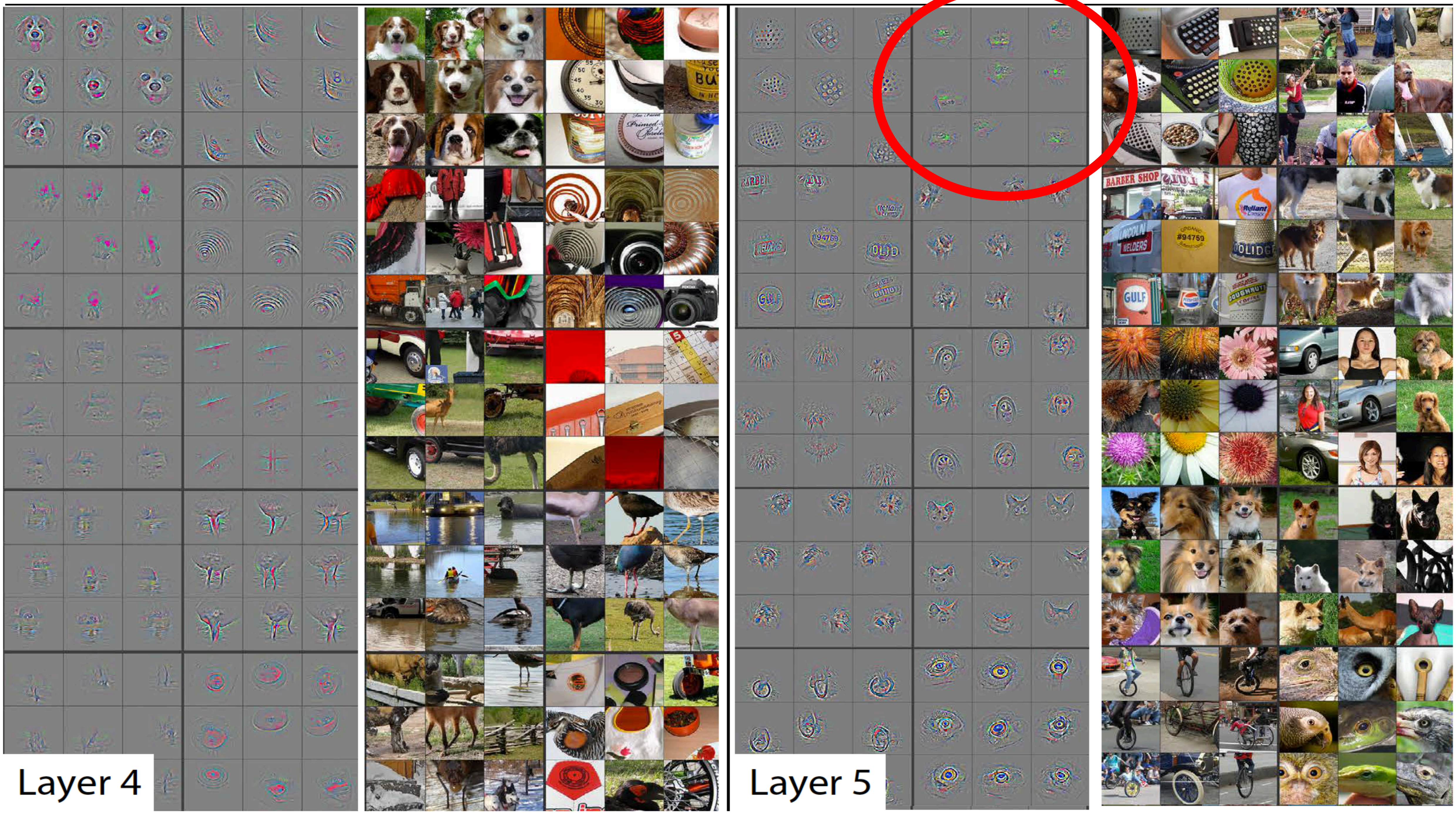
Matthew D. Zeiler, Rob Fergus [Visualizing and Understanding Convolutional Networks](), ECCV 2014

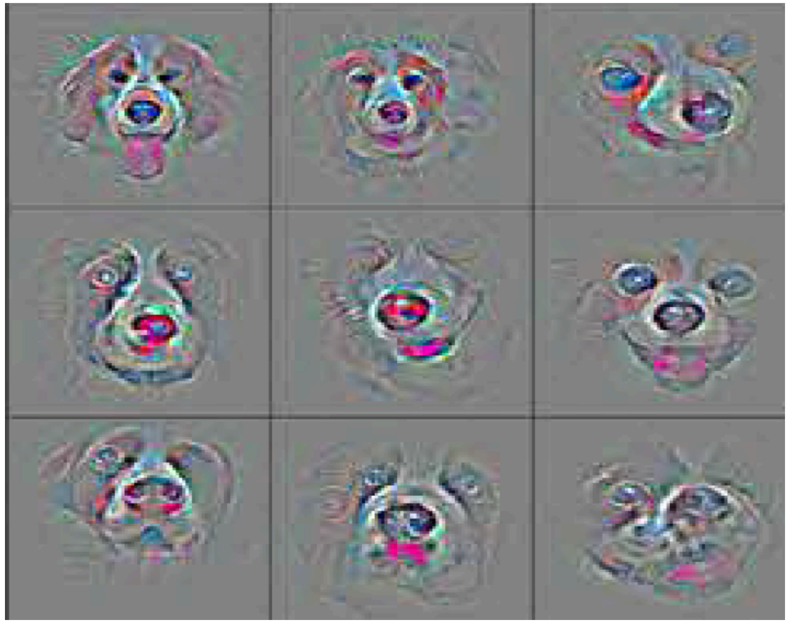


Matthew D. Zeiler, Rob Fergus [Visualizing and Understanding Convolutional Networks](), ECCV 2014

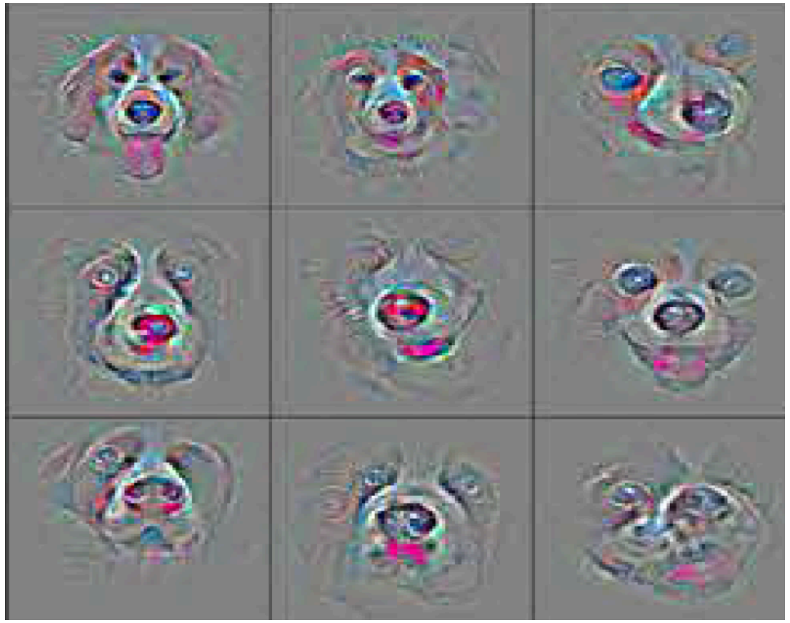


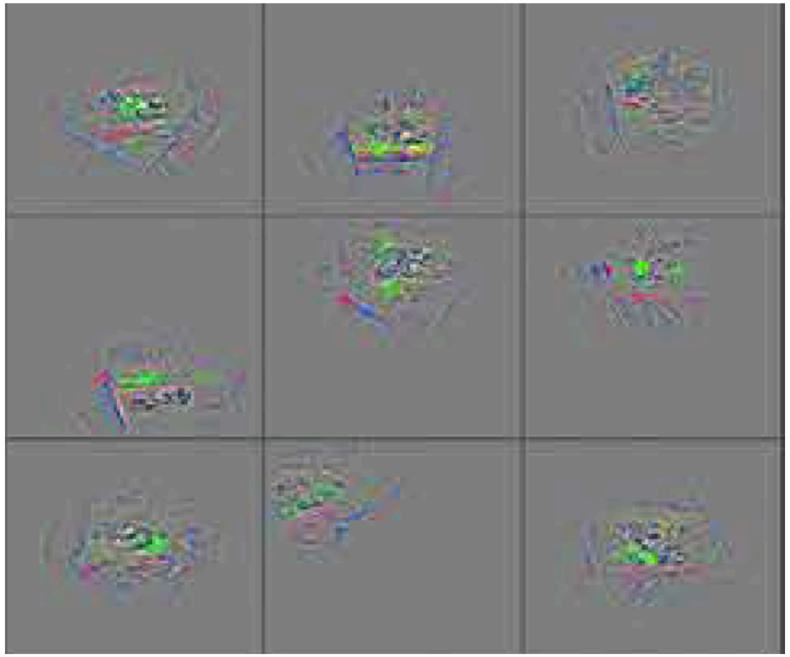
Matthew D. Zeiler, Rob Fergus [Visualizing and Understanding Convolutional Networks](), ECCV 2014

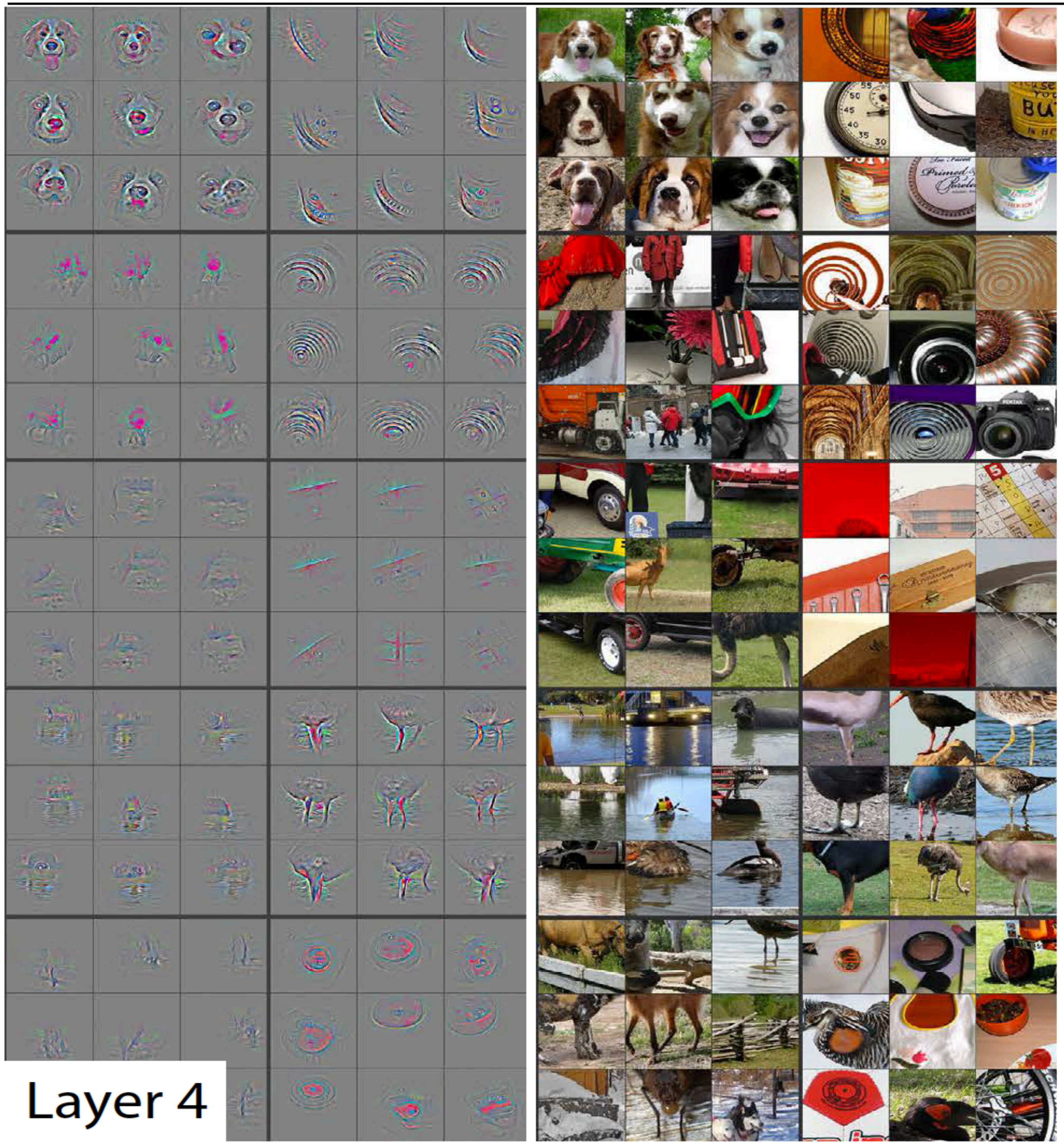

Layer 4
Layer 5

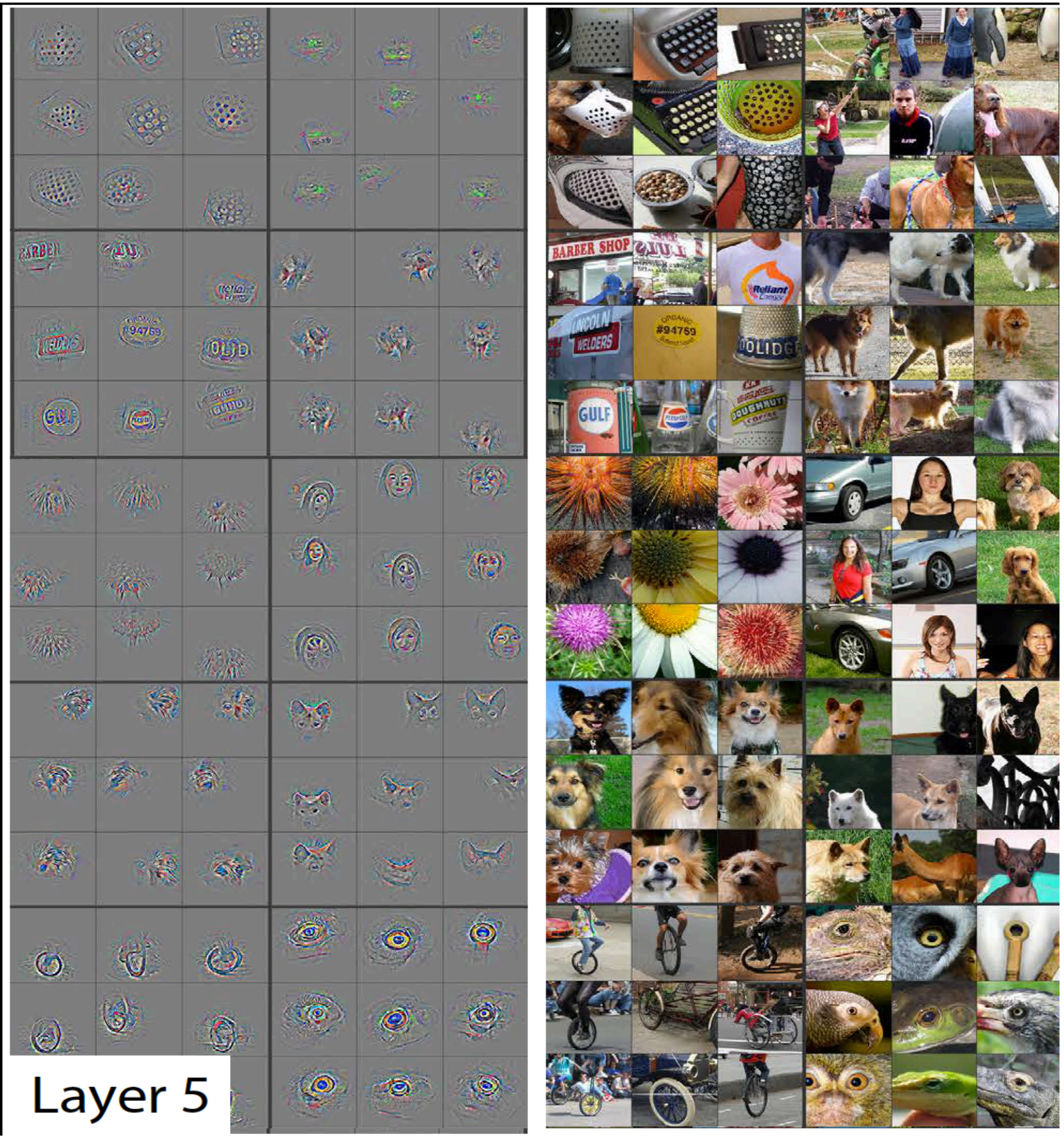


What features are being captured from these pictures?







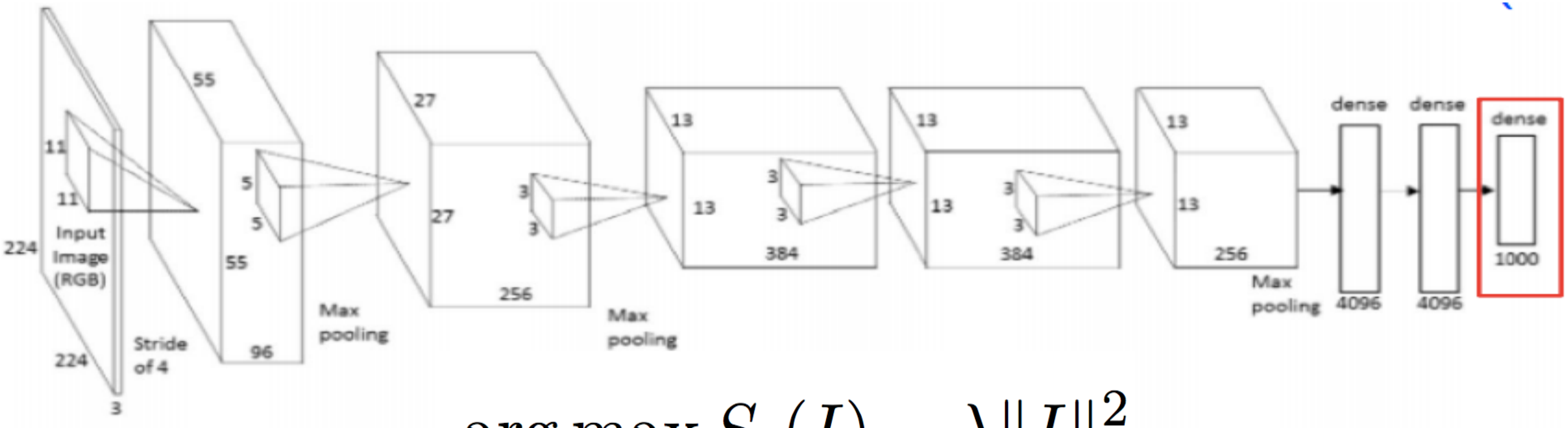
Matthew D. Zeiler, Rob Fergus [Visualizing and Understanding Convolutional Networks](), ECCV 2014

Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach

Optimization Approach

Can we find an image that increases some class score?

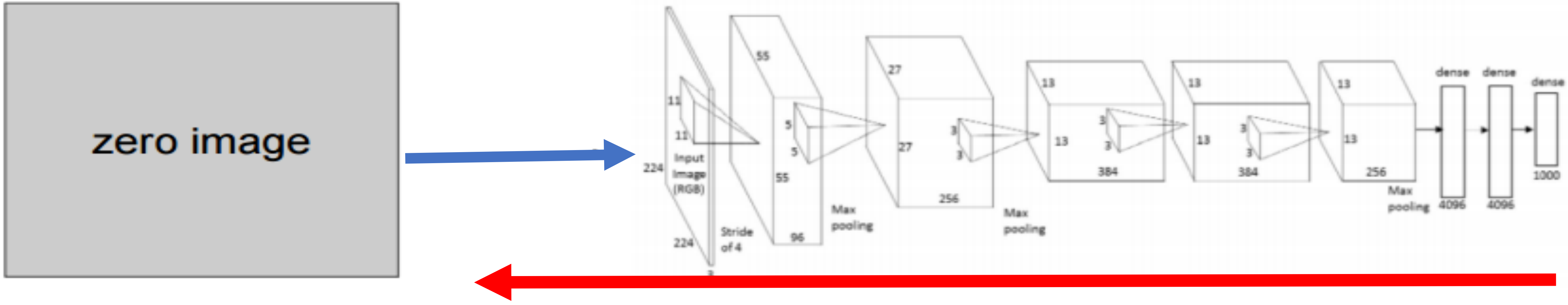


$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Score for class c before softmax

Regularization term

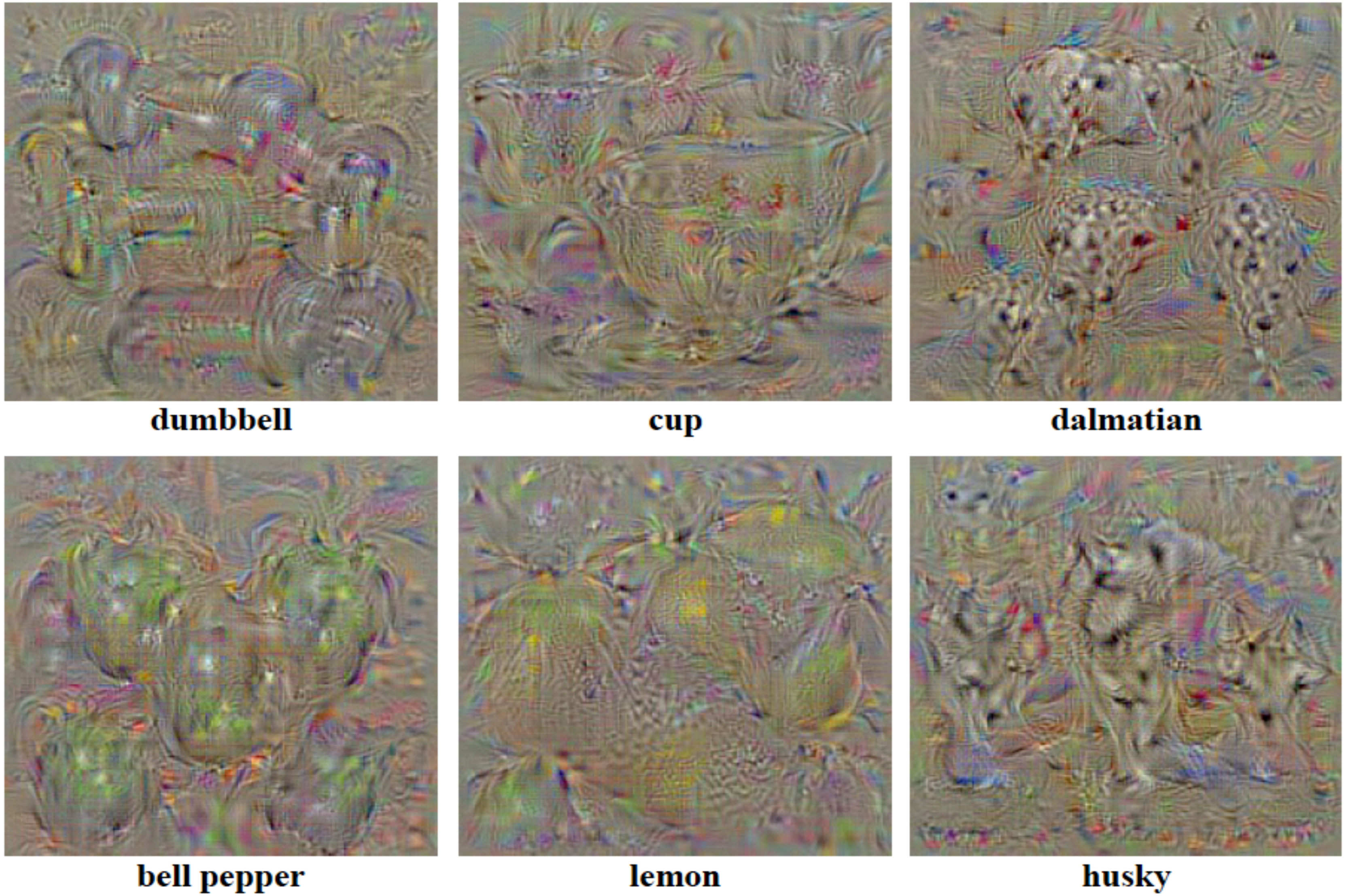
Optimization Approach - Algorithm



Start with zero image
Repeat:
  Feed image forward
  Set the gradient of the scores' vector to be [0,0,....1,....,0]
  Backward pass the gradients to the image
  Update image (add regularization to avoid large updates)

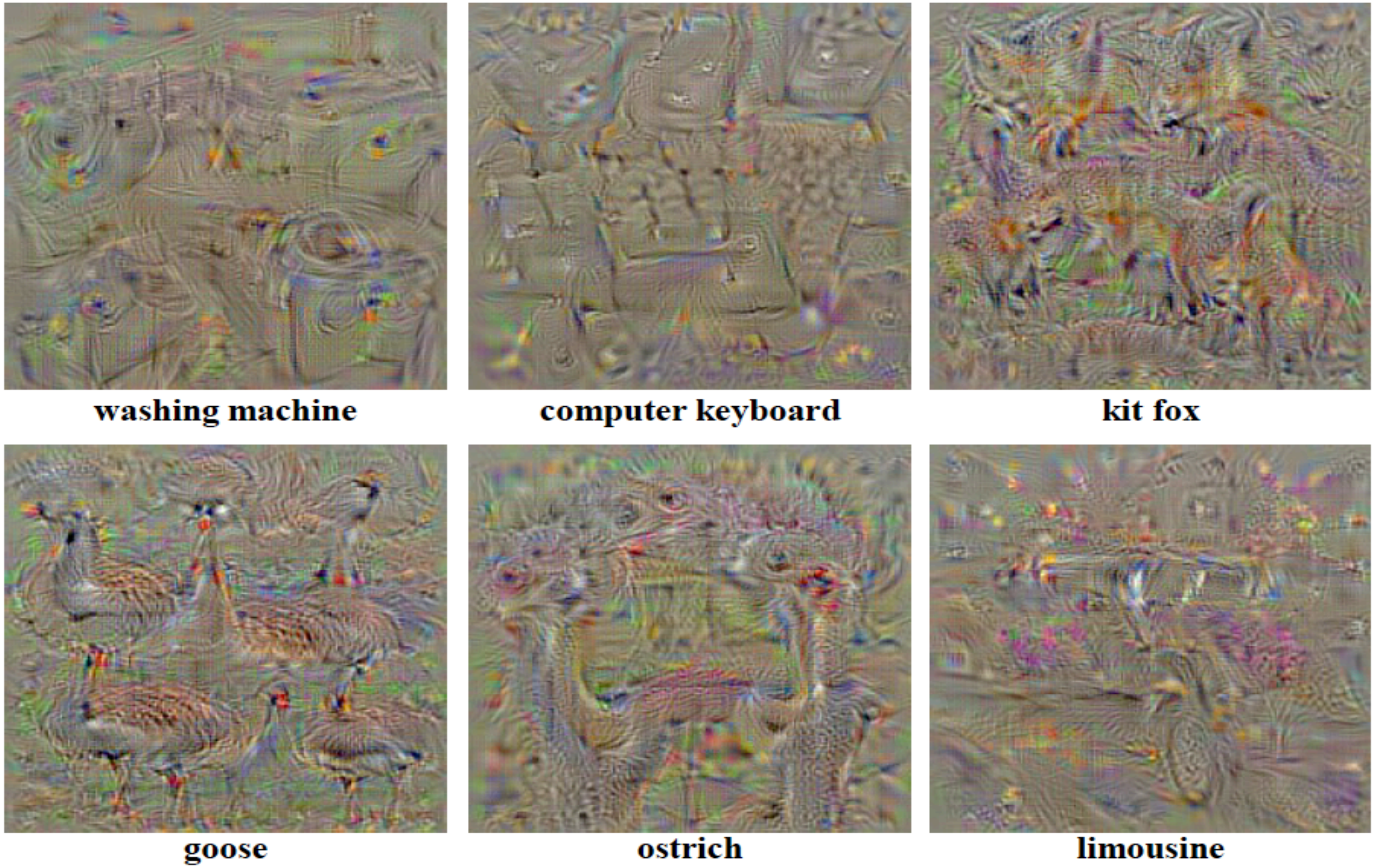
$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Optimization Approach - Examples



Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency MapsKaren Simonyan et al 2014

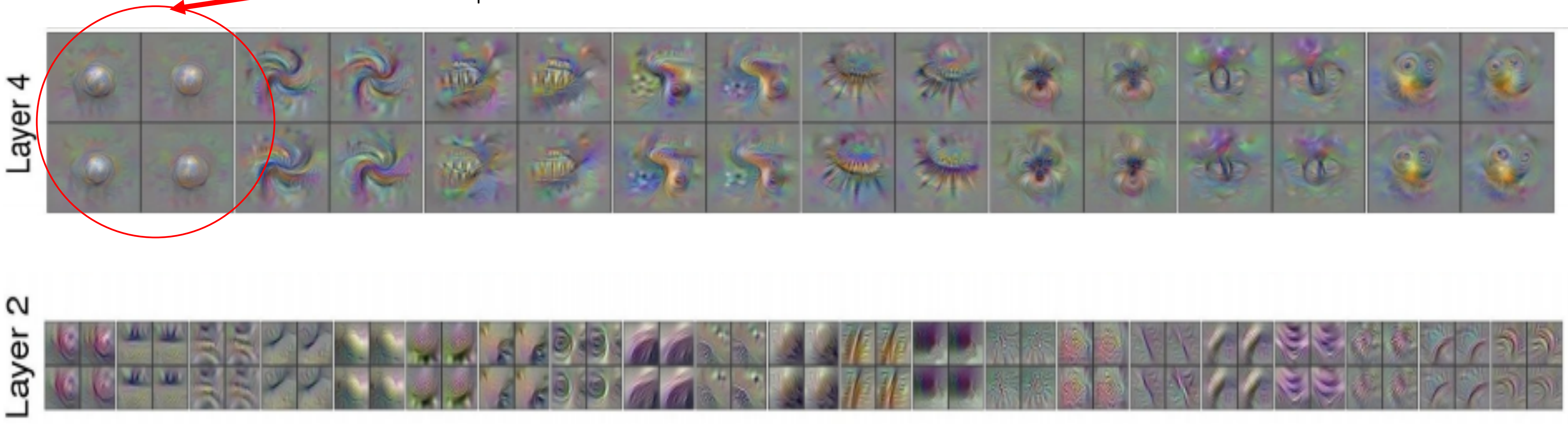
Optimization Approach - Examples



Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency MapsKaren Simonyan et al 2014

Visualizing Intermediate Layers

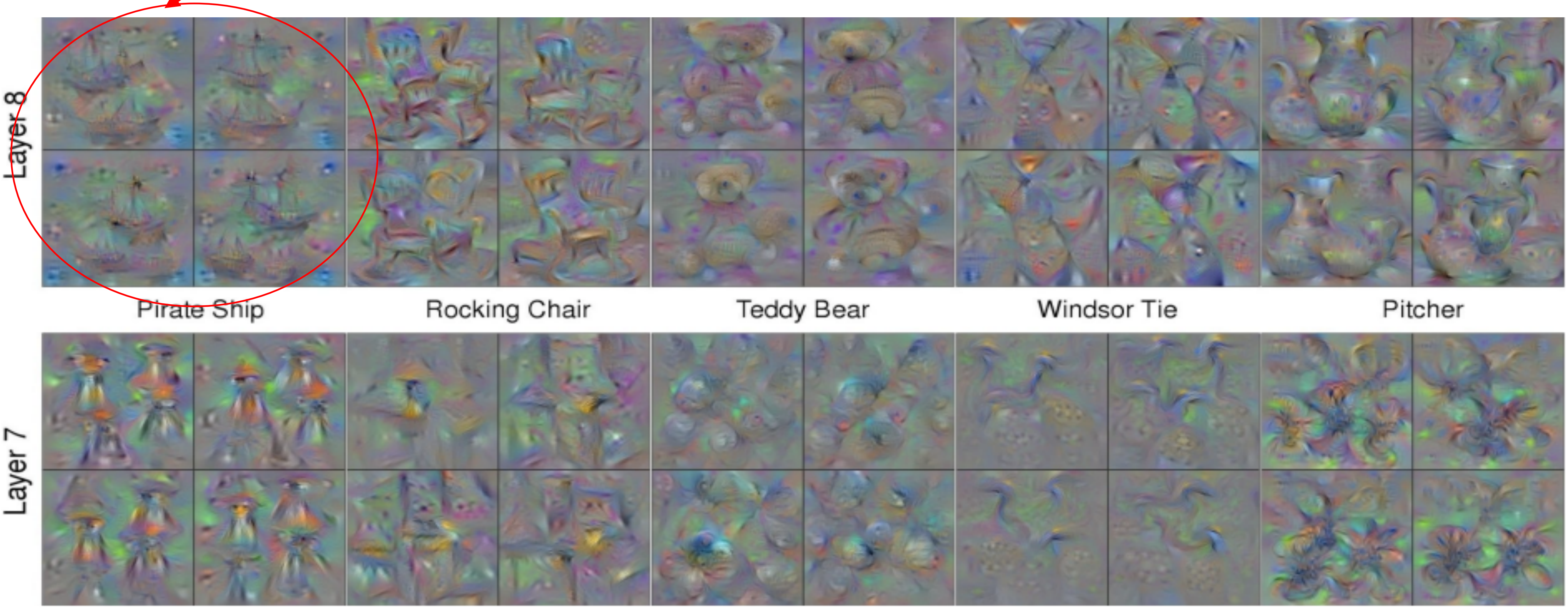
Smaller receptive field



Understanding Neural Networks Through Deep Visualization, Yosinski et al. - 2015]

Visualizing Intermediate Layers

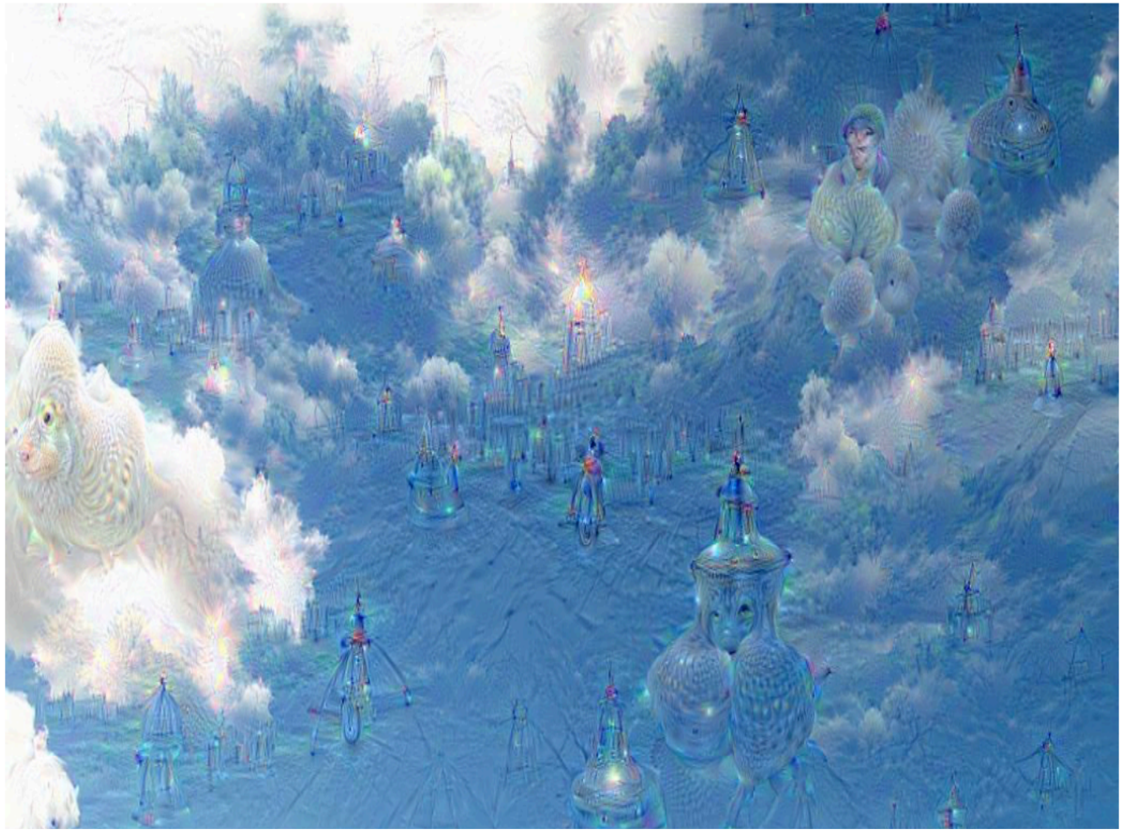
Large receptive field



Understanding Neural Networks Through Deep Visualization, Yosinski et al. - 2015]

What if we map back the gradients onto the original image?

What if we map back the gradients onto the original image?



Deep Dream Google

What if we map back the gradients onto the original image?

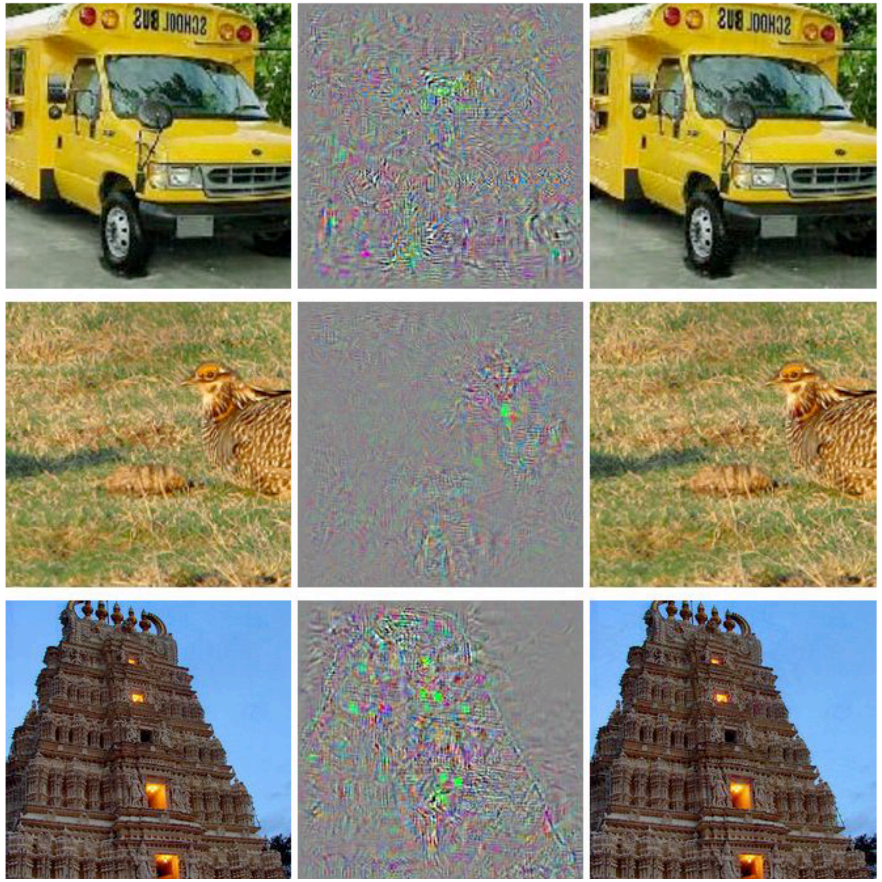
Deep Dream Google

What if we map back the gradients onto the original image?
[Deep Dream Grocery Store]

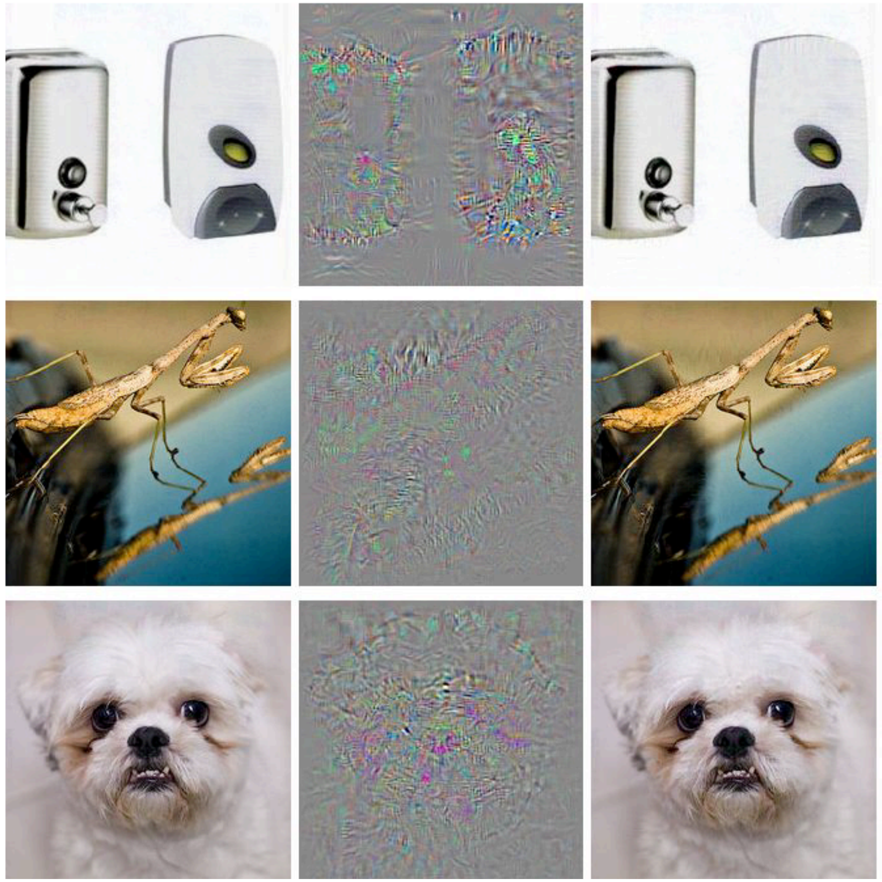
Deep Dream Google

Q: What is the difference between the gradient approach and the optimization approach for visualization?

Adversarial Examples



Correct Perturbation Wrong



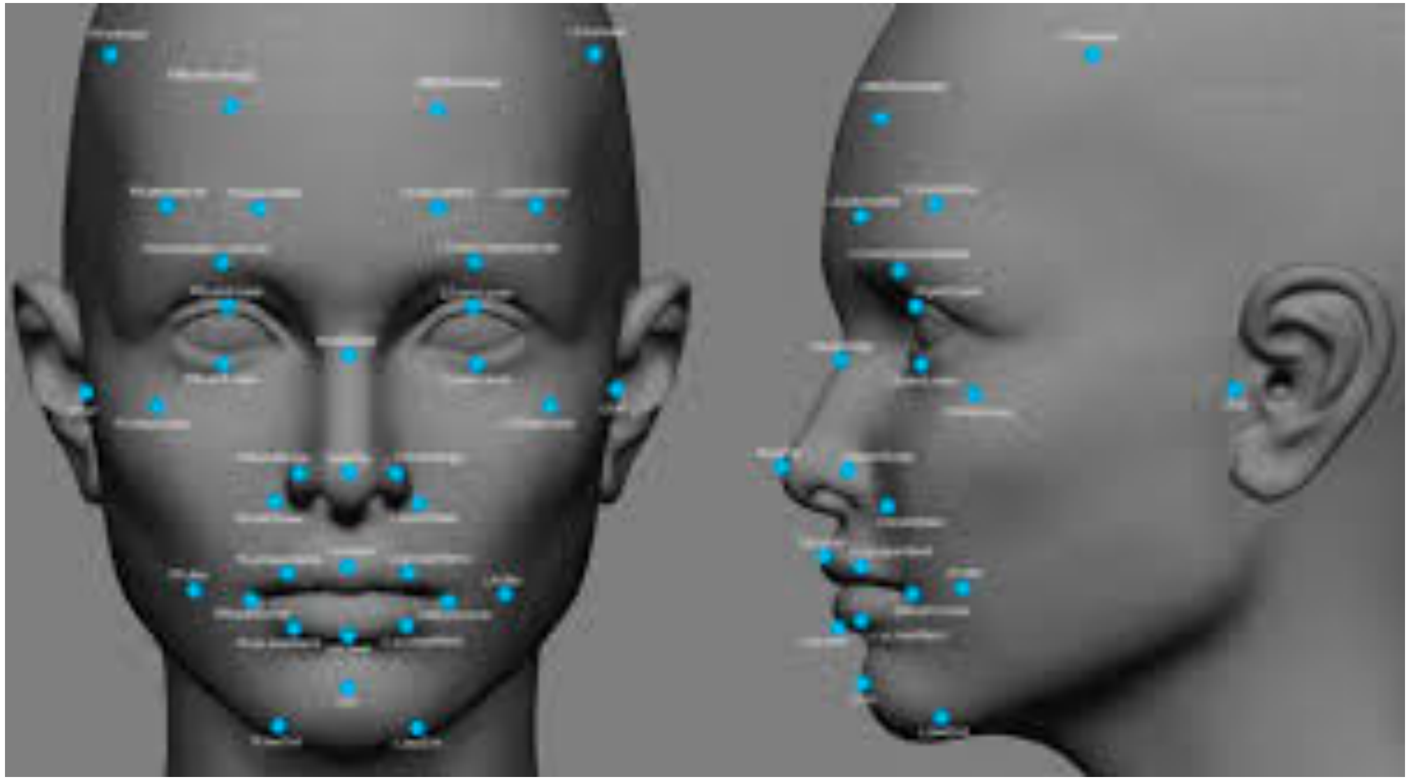
Correct Perturbation Wrong

$$K(X + v) \ != K(X),$$

where **K** is a classifier, **X** is input image, **v** is perturbation.

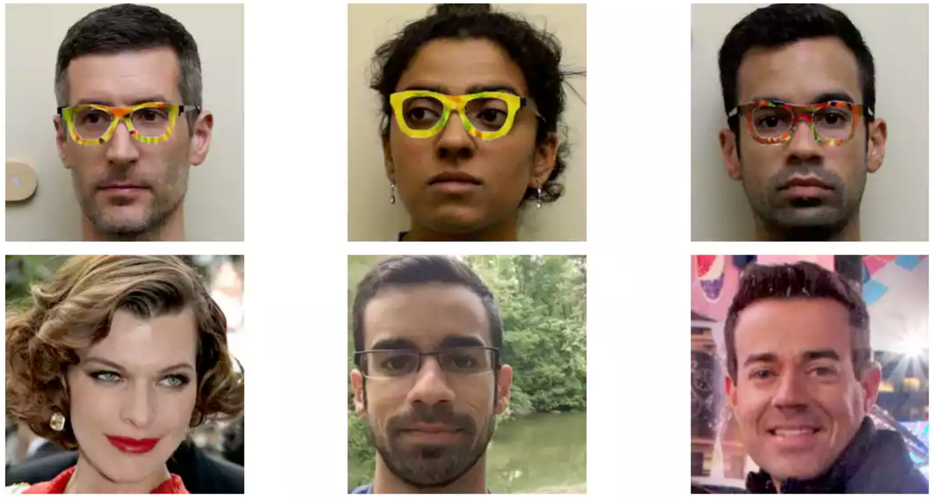
Intriguing properties of neural networks, Szegedy et al. - 2013

Why care about adversarial examples?



gizmodo.com & survivopedia.com & theguardian.com

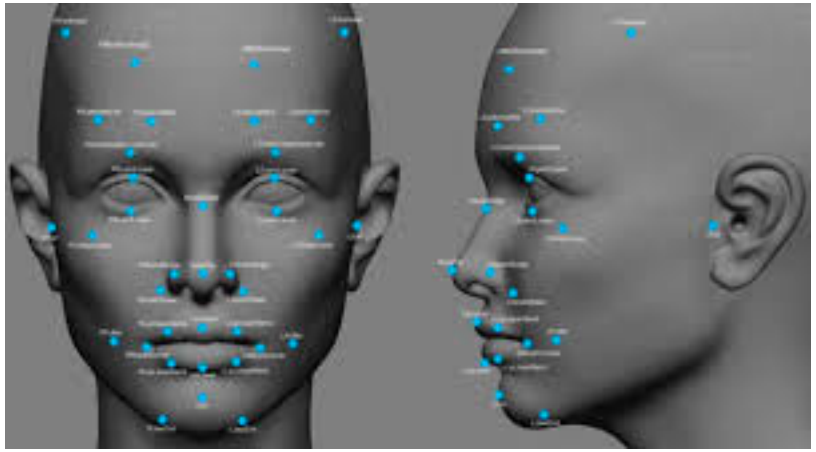
Why care about adversarial examples?



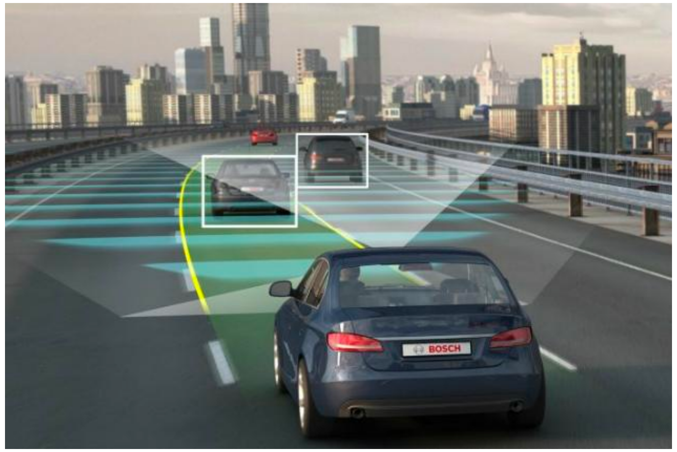
(b) (c) (d)

gizmodo.com & survivopedia.com & theguardian.com

Why care about adversarial examples?



Biometrics



Autonomous Driving

“Build safe, widely distributed AI.”
-- OpenAI

Security Guard Robot

Speech Recognition

extremetech.com & johndayautomotivelectronics.com & kingstonmouth.com & primecompetence.com

Outline –Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

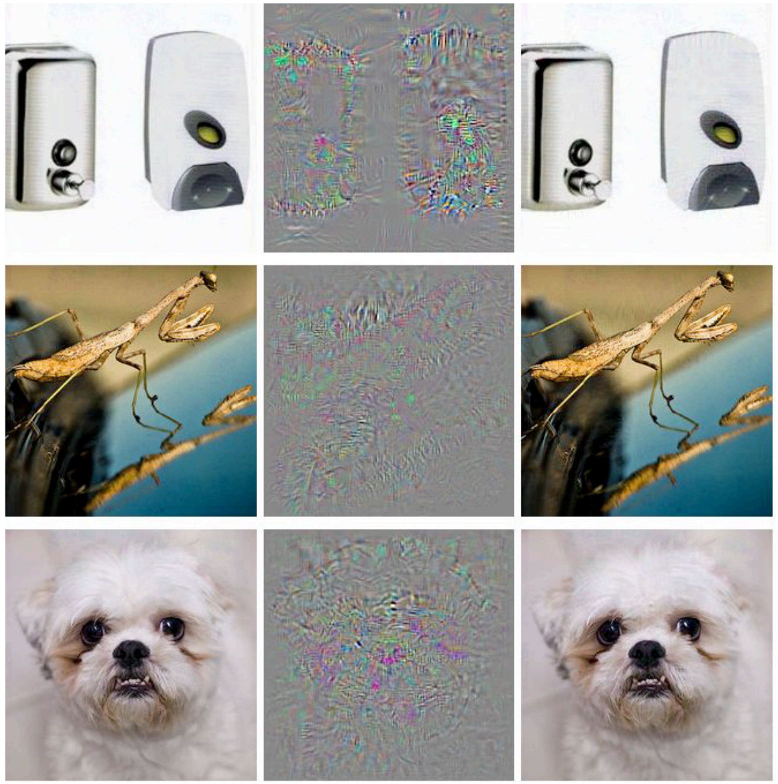
Outline –Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

Adversarial and Rubbish examples

Adversarial

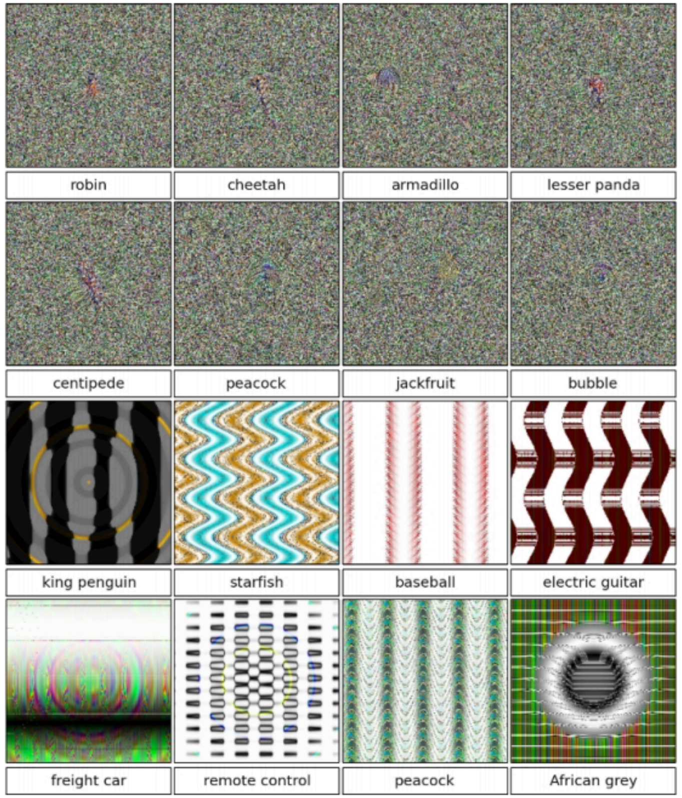
- corrupt an existing natural image



Correct Perturbation Wrong

Rubbish

- noisy meaningless pictures that achieve high confidence classification



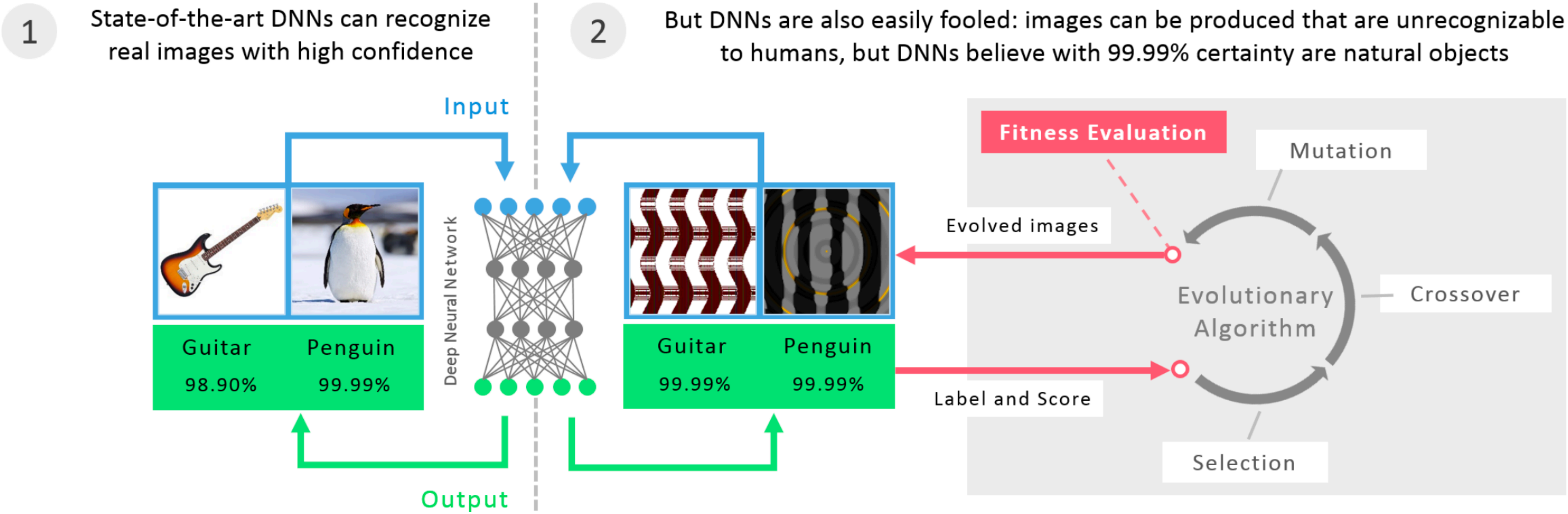
Intriguing properties of neural networks, Szegedy et al. - 2013

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014

Outline –Adversarial Examples

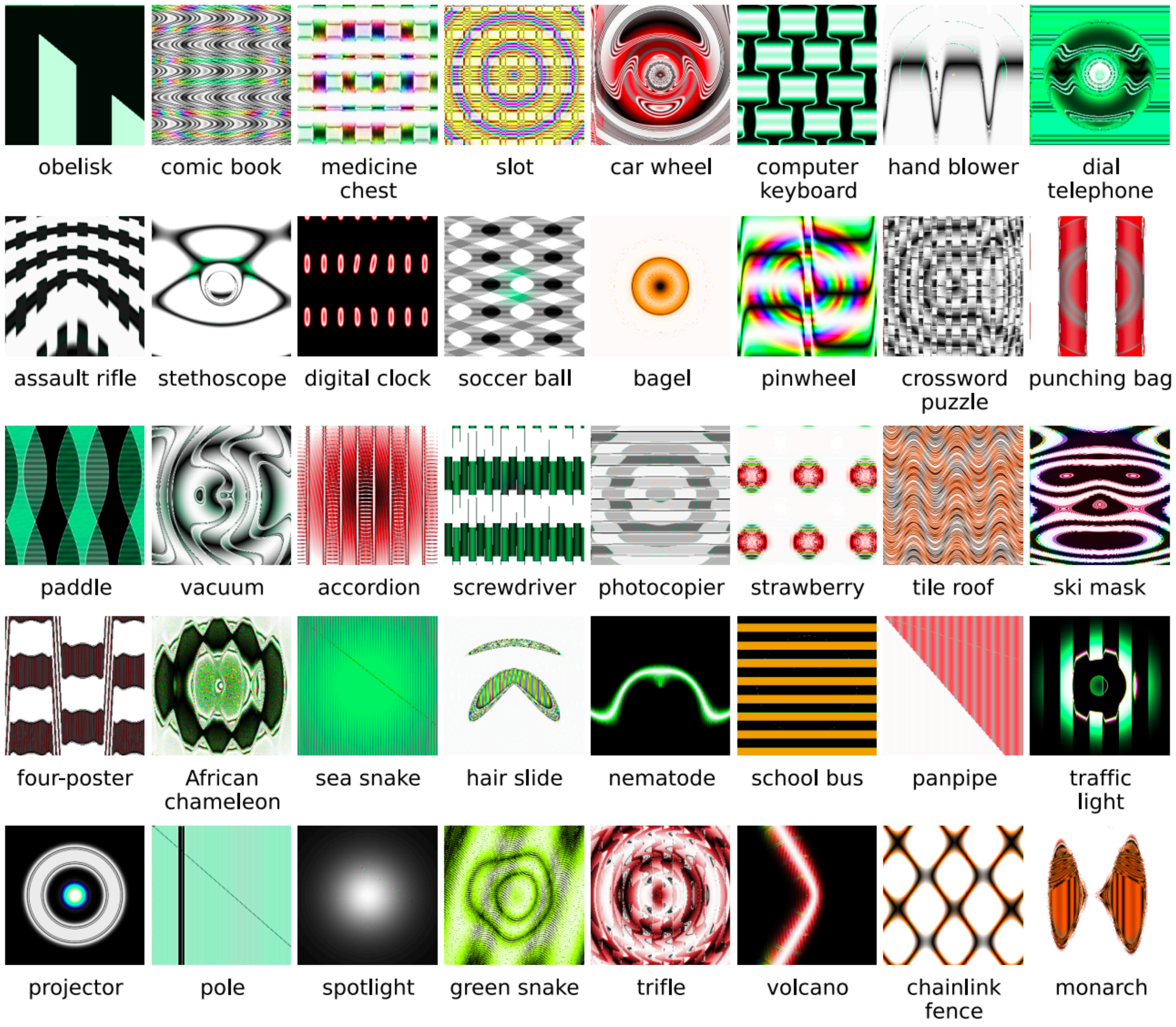
1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

Evolutionary Approach



Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014

Rubbish examples by evolutionary approach



Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014

Q: How can we change the image to fool the classifier?

Outline –Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

Gradient-based approaches for visualization

Input image

$$\underbrace{\arg\max_I S_c(I)}_{\text{Score of class c given input image}} - \underbrace{\lambda \|I\|_2^2}_{\text{Regularization term}}$$

Deep Inside Convolutional Networks Visualising Image Classification Models and Saliency Maps – Simonyan et al - 2013

Gradient-based approaches for ~~visualization~~ adversarial examples

Visualization:

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Adversarial examples:

1. Let $S_c(I)$ have high score for input I
2. We maximize the $- S_c\ (I + noise)$ w.r.t noise
3. and penalize the L2-norm of noise.
4. We get a new image X = (I + noise)

Deep Inside Convolutional Networks Visualising Image Classification Models and Saliency Maps – Simonyan et al - 2013

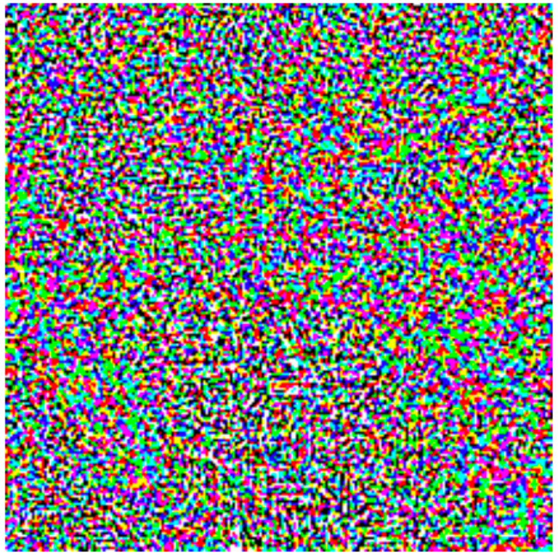
Fast Gradient Sign Method

Score of label y_{true}, given input image X

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}, y_{true})\right)$$

$=$

$+\,.007\,\times$



adversarial perturbation

Adversarial examples in the physical world - Kurakin, et al - 2016
Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014

Fast Gradient Sign Method

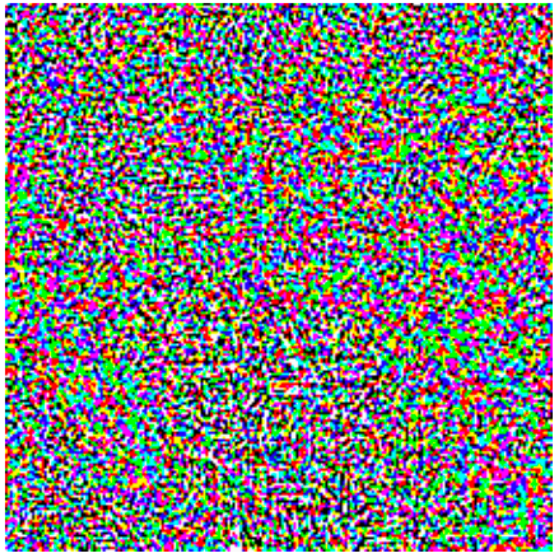
$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}, y_{true})\right)$$

“gibbon”

$=$

“panda”

$+ .007 \times$

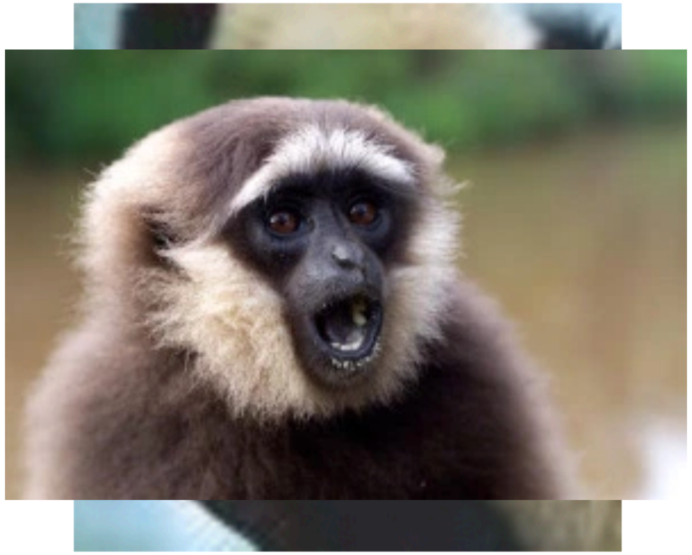


adversarial perturbation

Adversarial examples in the physical world - Kurakin, et al - 2016
Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014

Fast Gradient Sign Method

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}, y_{true})\big)$$

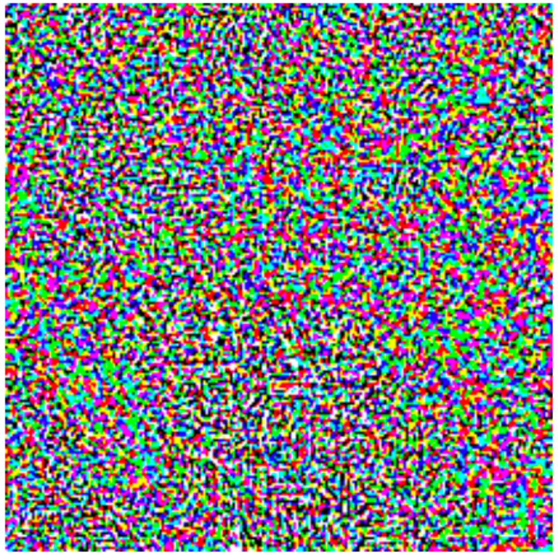


“gibbon”

$=$

“panda”

$+\, .007 \times$



adversarial perturbation

Adversarial examples in the physical world - Kurakin, et al - 2016

Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014

Gradients-based Methods

- Fast Gradient Sign Method:

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon\, \mathrm{sign}\big(\nabla_X J(\boldsymbol{X}, y_{true})\big)$$

- Iterative Gradient Sign Method

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} + \alpha\, \mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\Big\}$$

Iteratively repeat

Adversarial examples in the physical world - Kurakin, et al - 2016

Gradients-based Methods

- Fast Gradient Sign Method:

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon \operatorname{sign}\big(\nabla_X J(\boldsymbol{X}, y_{true})\big)$$

- Iterative Gradient Sign Method

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} + \alpha \operatorname{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\Big\}$$

- Iterative Least-likely Class Method

$$y_{LL} = \arg\min_y \big\{p(y|\boldsymbol{X})\big\}$$

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} - \alpha \operatorname{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{LL})\right)\right\}$$

Adversarial examples in the physical world - Kurakin, et al - 2016

Visual Comparison of Gradients-based Methods

Natural Image

Fast Gradient Sign

Clean image

“Fast”; L_∞ distance to clean image = 32

“Basic iter.”; L_∞ distance to clean image = 32

“L.l. class”; L_∞ distance to clean image = 28

Iterative Gradient Sign

Iterative LL-Class Gradient Sign

Adversarial examples in the physical world - Kurakin, et al - 2016

Outline –Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

Adversarial Training

Q: How can we use adversarial examples to train a robust network?

A: Train it both on natural images and constructed adversarial images.

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \underbrace{\alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y)}_{\text{Training Target}} + \underbrace{(1-\alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right))}_{\text{Adversarial regularizer}}$$

Adversarial examples in the physical world - Kurakin, et al - 2016

Adversarial Training

How can we use adversarial examples to train a robust network?

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)$$

Training Target

Adversarial regularizer

For natural images, error rate drops from 0.94% to 0.84% on mnist.
For adversarial images, error rate drops from 89.4% to 17.9% on mnist.

Adversarial examples in the physical world - Kurakin, et al - 2016

Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

How much information do we need to fool a neural net?

Model weights	Have full access to model weights
Architecture	Know what the model looks like
Training data	Know what training dataset was used
Oracle/black box	Query model with input **X**, get label **Y**

Black box example – what we hear

You have lettuce in your teeth

https://www.youtube.com/watch?v=vM5C4nHUQDs

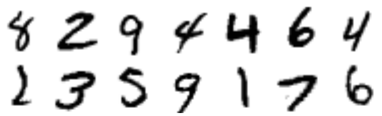
Black box example – what we hear

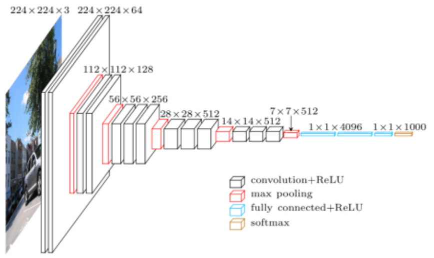
https://www.youtube.com/watch?v=vM5C4nHUQDs

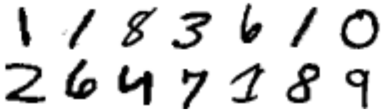
Transferability scenarios

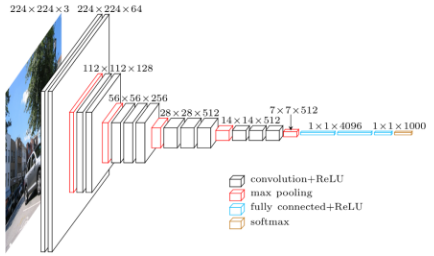
Cross training-set generalization

- Same architecture, different training set





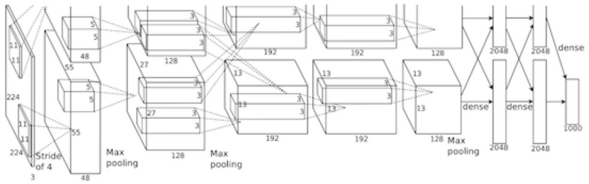




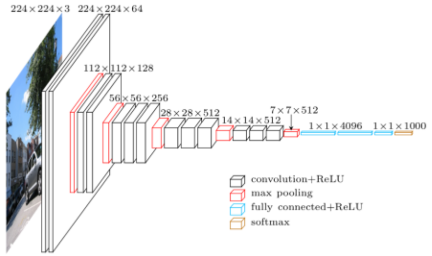
Cross model generalization

- Different architecture, same training set

IMAGENET



IMAGENET



https://www.cs.toronto.edu/~frossard/post/vgg16/
http://johnloeber.com/docs/kmeans.html

Generalization error rates

	Model 1 - Dataset 1	Model 2 - Dataset 1	Model 1 - Dataset 2
M1 D1	100%	26.2%	5.9%
M2 D1	6.25%	100%	5.1%
M1 D2	8.2%	8.2%	100%
Gaussian noise	2.2%	2.6%	2.4%

Table 1: Fooling rate for average perturbation $\mathbf{stddev} = 0.06$

Intriguing properties of neural networks - Szegedy et al - 2013

Generalization error rates

	Model 1 - Dataset 1	Model 2 - Dataset 1	Model 1 - Dataset 2
M1 D1	100%	98%	43%
M2 D1	96%	100%	22%
M1 D2	27%	50%	100%
Gaussian noise	2.6%	2.8%	2.7%

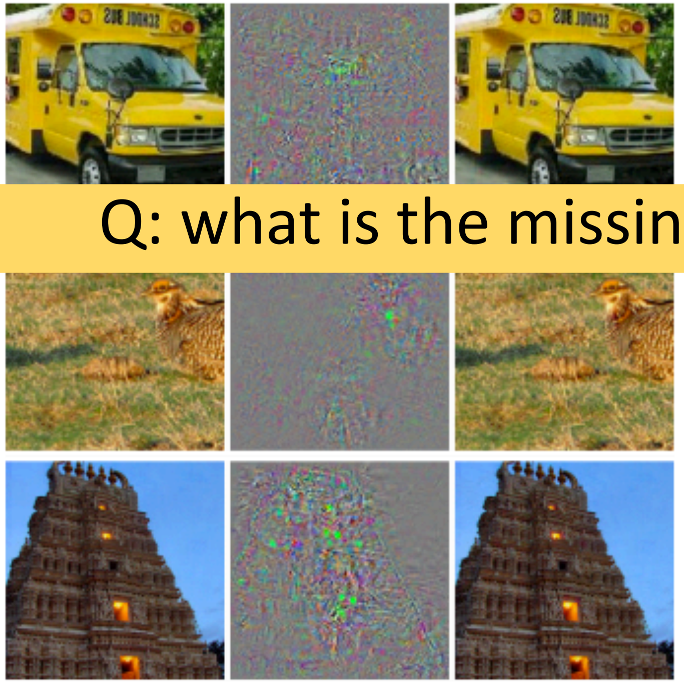
Table 2: Fooling rate for average perturbation $\mathbf{stddev} = 0.1$

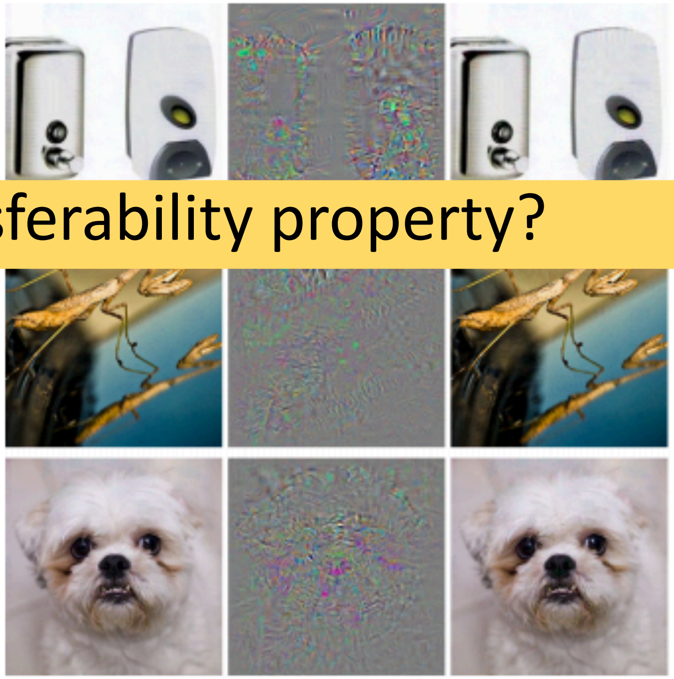
Intriguing properties of neural networks - Szegedy et al - 2013

This is a very inefficient process

Intriguing properties of neural networks - Szegedy et al - 2013

This is a very inefficient process





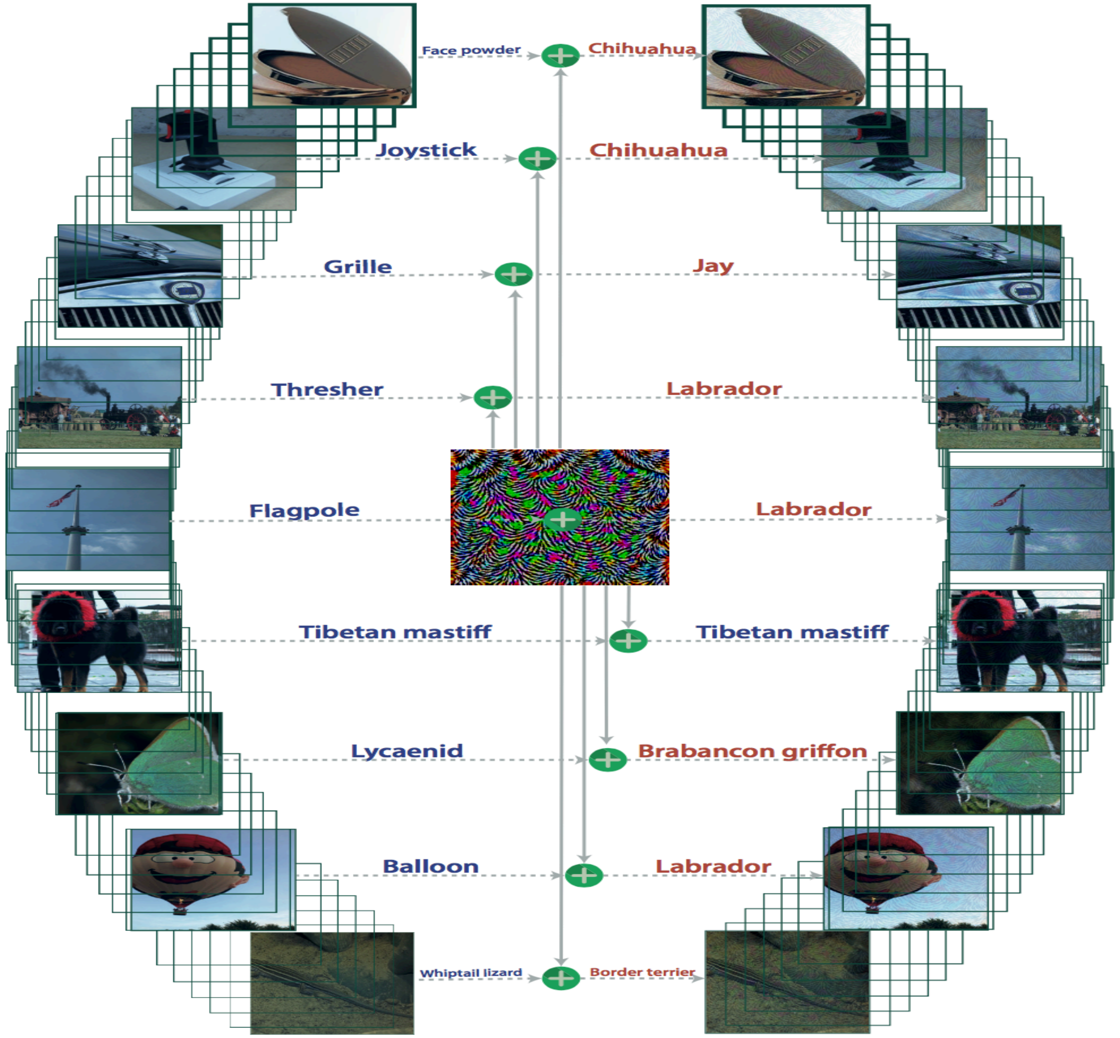
Q: what is the missing transferability property?

Intriguing properties of neural networks - Szegedy et al - 2013

Outline –Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

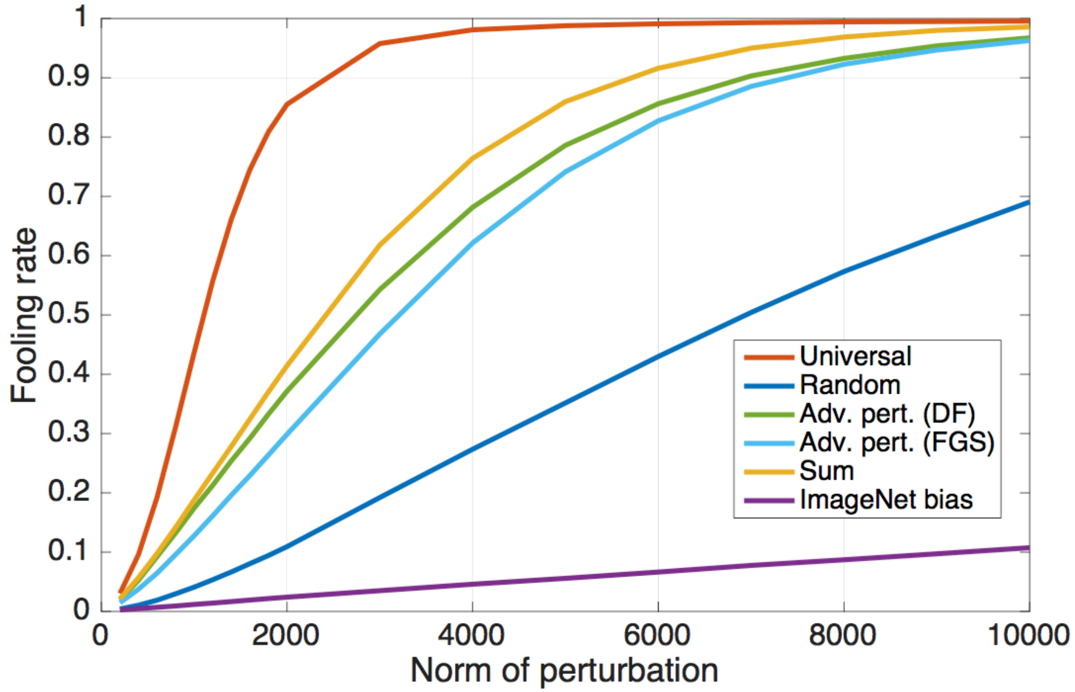
Universal Adversarial Perturbations



Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016

Candidate universal perturbations

- **Random noise**
  - Easy to compute
  - Needs high norm to be effective
  - Obvious to human
- **Sum of all adversarial perturbations over X**
  - Less obvious
  - Components known to be effective
  - Very expensive (compute |**X**| times)
- **Universal Adversarial Perturbations (new method)**
  - Adaptively expensive (compute for a subset of **X**)
  - Very subtle



Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016

Algorithm

Intuition:

1. Start with $\mathbf{v = 0}$
2. If $(\mathbf{X_i + v})$ is misclassified, skip to $\mathbf{X_{i+1}}$
3. Find minimum perturbation $\mathbf{\Delta v}$ that takes $\mathbf{X_i + v + \Delta v}$ to another class
4. Update $\mathbf{v = v + \Delta v}$
5. Repeat with $\mathbf{X_{i+1}}$

Algorithm 1 Computation of universal perturbations.

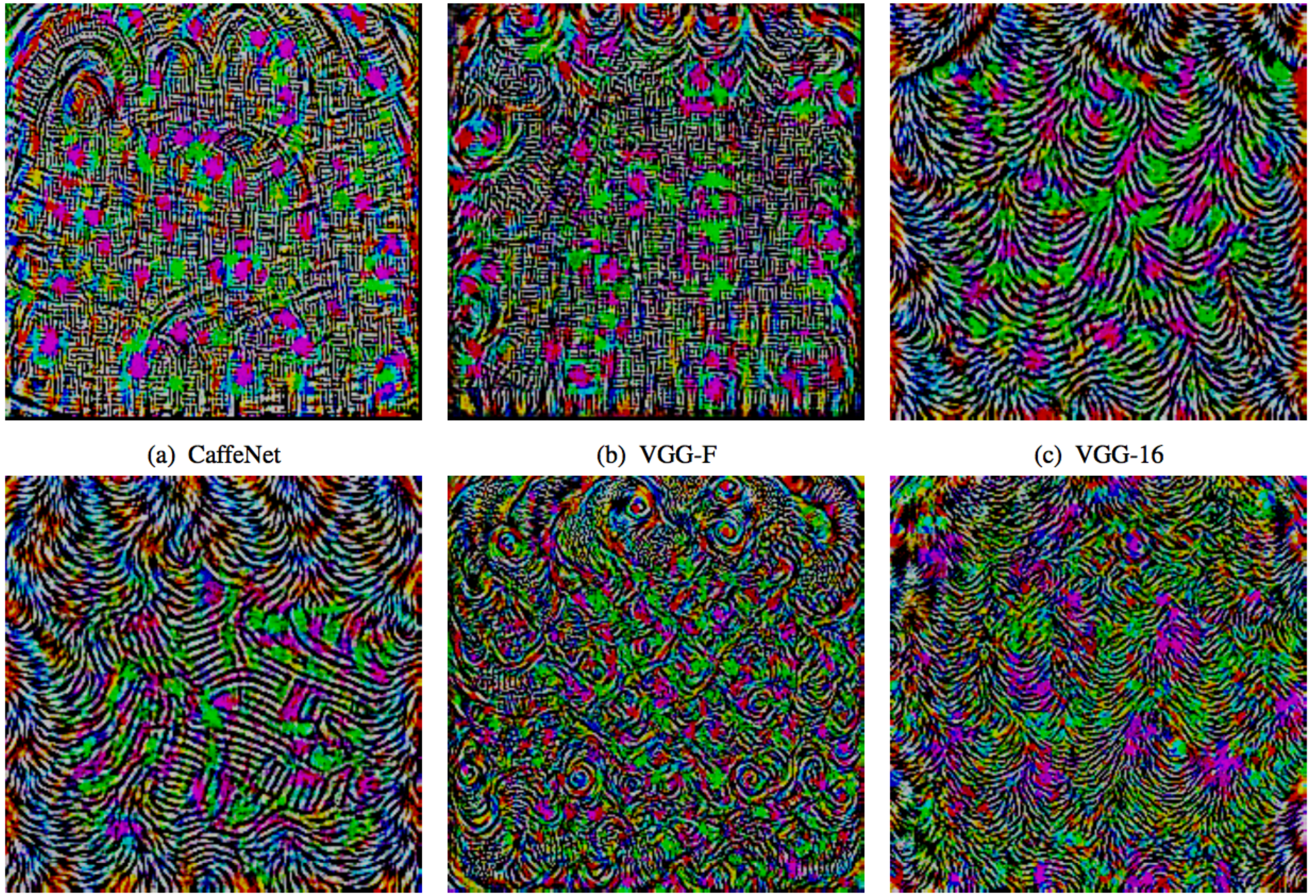
```
input: Data points X, classifier k̂, desired ℓ_p norm of the perturbation ξ, desired accuracy on perturbed samples δ.
output: Universal perturbation vector v.
Initialize v ← 0.
while Err(X_v) ≤ 1 − δ do
    for each datapoint x_i ∈ X do
        if k̂(x_i + v) = k̂(x_i) then
            Compute the minimal perturbation that sends x_i + v to the decision boundary:
                Δv_i ← arg min_r ||r||_2
                s.t. k̂(x_i + v + r) ≠ k̂(x_i).
            Update the perturbation:
                v ← P_{p,ξ}(v + Δv_i).
        end if
    end for
end while
```

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$

$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016

Sample universal perturbations



(a) CaffeNet (b) VGG-F (c) VGG-16

(d) VGG-19 (e) GoogLeNet (f) ResNet-152

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016

Cross-model universality

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Fooling rate when computing a perturbation for one model (rows) and testing it on others (columns)

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016

Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

Models are too linear

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights

class 1 score = dot product:
= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3

Models are too linear

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	?	?	?	?	?	?	?	?	?	?	

class 1 score = dot product:
= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3

Models are too linear

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	1.5	-1.5	3.5	-2.5	2.5	1.5	1.5	-3.5	4.5	1.5	

class 1 score before:
-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3

-1.5+1.5+3.5+2.5+2.5-1.5+1.5-3.5-4.5+1.5 = 2

Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack

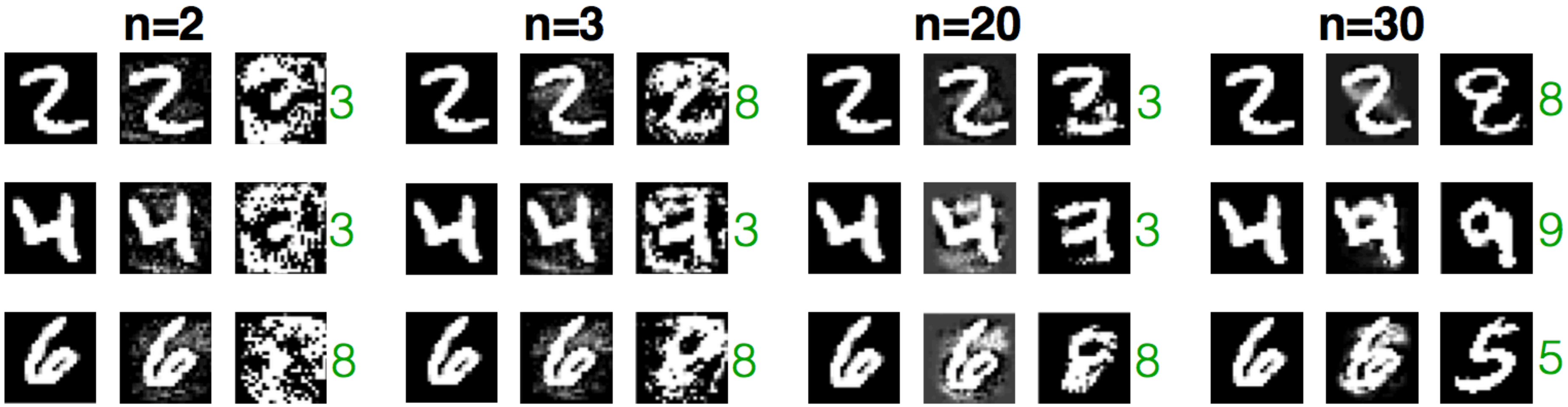
Proposed solution: highly non-linear models

- Use a rectified polynomial as the activation

$$F_n(x) = \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

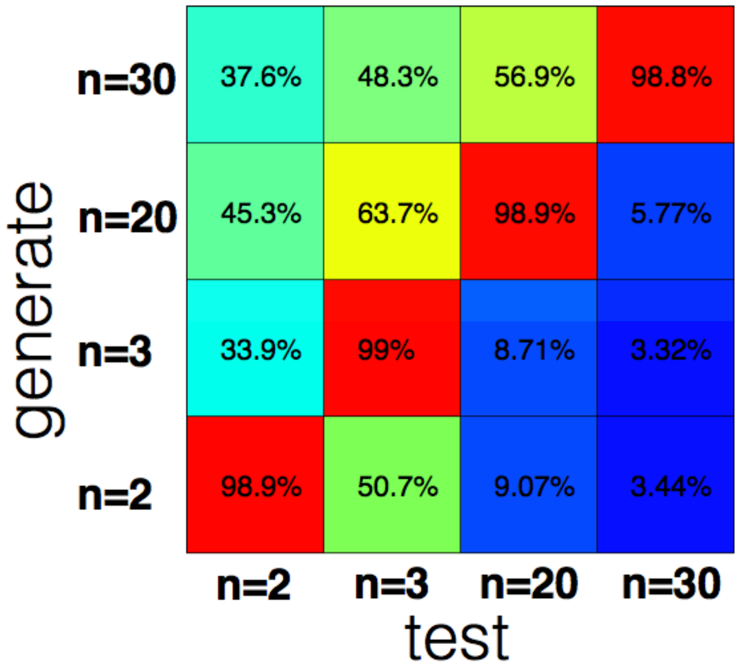
Dense Associative Memory is Robust to Adversarial Inputs - Dmitri Kotrov, John J Hopfield - 2017

Robustness against Adversarial Examples



Dense Associative Memory is Robust to Adversarial Inputs - Dmitri Kotrov, John J Hopfield - 2017

Fooling Rate



clean MNIST test set:

$$\text{error}_{n=2} = 1.51\%$$

$$\text{error}_{n=3} = 1.44\%$$

$$\text{error}_{n=20} = 1.61\%$$

$$\text{error}_{n=30} = 1.80\%$$

Dense Associative Memory is Robust to Adversarial Inputs - Dmitri Kotrov, John J Hopfield - 2017

Summary

Visualization

- ✓ What is Visualization?
- ✓ Visualize patches that maximally activate neurons
- ✓ Visualize the weights
- ✓ Gradient based approaches
- ✓ Optimization based approach

Adversarial Examples

- ✓ Adversarial and Rubbish examples
- ✓ Evolutionary approach
- ✓ Gradient based approaches
- ✓ Adversarial training
- ✓ Transferability
- ✓ Universal Adversarial Perturbations
- ✓ Why are neural networks easily fooled?
- ✓ Proposed Solutions for adversarial attack

Reading list

- Matthew D. Zeiler, Rob Fergus Visualizing and Understanding Convolutional Networks, ECCV 2014
- Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps arXiv:1312.6034v2
- Alexey Dosovitskiy Thomas Brox, Inverting Visual Representations with Convolutional Networks, CVPR 2016
- Anh Nguyen, Jason Yosinski, Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, CVPR 2015
- Christian Szegedy, et al. Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199v4
- Alexey Kurakin, et al, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533
- Seyed-Mohsen Moosavi-Dezfooli, et al, Universal adversarial perturbations, arXiv preprint arXiv:1610.08401v2
- Dmitry Krotov, et al, Dense Associative Memory is Robust to Adversarial Inputs, arXiv preprint arXiv:1701.00939
- Ian J. Goodfellow, et al, Explaining and Harnessing Adversarial Examples, arXiv preprint arXiv:1412.6572
- Nicholas Carlini et al, Hidden Voice Commands, 25th USENIX Security Symposium
- Brian Chu et al, Visualizing Residual Networks, arXiv preprint arXiv:1701.02362
- Nicolas Papernot et al, SoK: Towards the Science of Security and Privacy in Machine Learning, arXiv preprint arXiv:1611.03814
- Nicolas Papernot et al, Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples, arXiv preprint arXiv:1602.02697
- Ian J. Goodfellow et al, Attacking machine learning with adversarial examples, OpenAI blog post

Conclusion