Interpretability of Neural Networks

Kian Katanforoosh

Neural networks are deployed:

- In our phones: to recommend content,
- In banks: to manage investments,
- In hospitals: to help doctors diagnose disease symptoms,
- In insurance agencies: to evaluate risk and underwrite documents,
- In cars: to help avoid accidents.

How can one who will be held accountable for a decision trust a neural network's recommendation, and justify its use?

[Xie, Ras, et al. (2020): Explainable Deep Learning: A Field Guide for the Uninitiated]

Today's outline

I. Interpreting Neural Networks' outputs

A. with saliency maps

B. with occlusion sensitivity

C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside

A. with gradient ascent (class model visualization)

B. with dataset search

C. the deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)

I. A. Interpreting and visualizing Neural Networks with saliency maps

Context: You have built an animal classifier for a zoo. They are reluctant to use your model without human supervision, because they don't understand the decision process of the model.

Question: How can you alleviate their concerns?

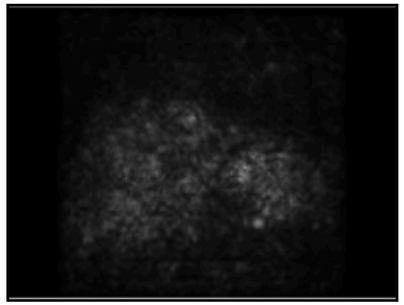
CNN

softmax

$$\begin{pmatrix} 0.04 \\ 0.85 \\ \vdots \\ 0.07 \\ 0.01 \\ \vdots \\ 0.02 \end{pmatrix} \begin{matrix} \text{“iguana} \\ \text{“dog”} \\ \\ \text{“cat”} \\ \text{“ant”} \\ \\ \text{“crab”} \end{matrix}$$

$$softmax\begin{pmatrix} s_{iguana} \\ s_{dog} \\ \vdots \\ s_{cat} \\ s_{ant} \\ \vdots \\ s_{crab} \end{pmatrix} = \begin{pmatrix} \frac{s_{iguana}}{\sum_{animals} s_{animal}} \\ \frac{s_{dog}}{\sum_{animals} s_{animal}} \\ \vdots \\ \frac{s_{crab}}{\sum_{animals} s_{animal}} \end{pmatrix}$$

$$\frac{\partial s_{dog}(x)}{\partial x} =$$



indicates which pixels need to be changed the least to affect the class score the most.

Can be used for segmentation?

Yes

Saliency maps

Kian Katanforoosh

[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

I. B. Interpreting and visualizing Neural Networks with occlusion sensitivity

Context: You have built an animal classifier for a zoo. They are a little reluctant to use your model without human supervision, because they don't understand the decision process of the model.

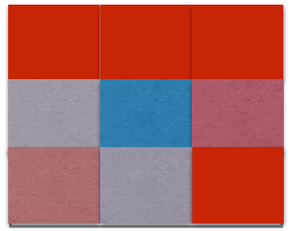
Question: What can you do to alleviate their concerns?

→ CNN →

0.04	"iguana"
0.85	**"dog"**
⋮	
0.07	"cat"
0.01	"ant"
⋮	
0.02	"crab"

→ CNN →

0.05	"iguana"
0.83	**"dog"**
⋮	
0.07	"cat"
0.01	"ant"
⋮	
0.02	"crab"



Probability map of the true class for different positions of the grey square

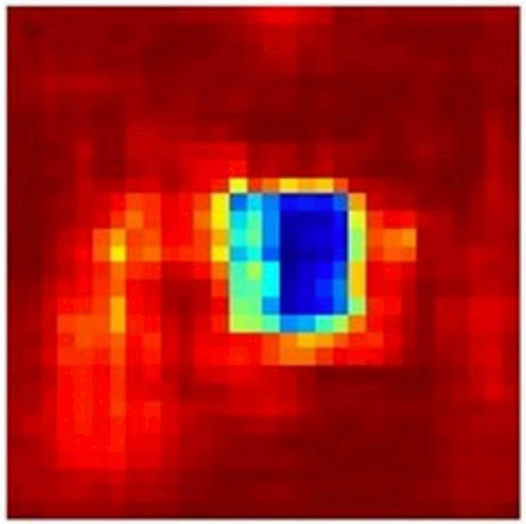
Indicates low confidence on the true class for the corresponding position of the grey square

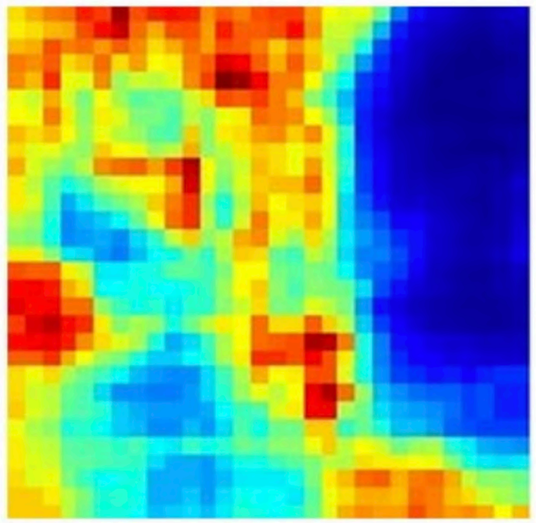
Indicates high confidence on the true class for the corresponding position of the grey square

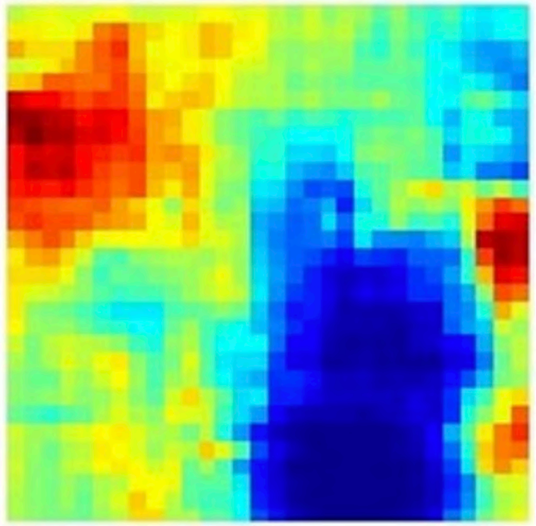
Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

I. B. Interpreting and visualizing Neural Networks with occlusion sensitivity







Probability map of the true class for different positions of the grey square

Indicates low confidence on the true class for the corresponding position of the grey square

Indicates high confidence on the true class for the corresponding position of the grey square

Occlusion sensitivity

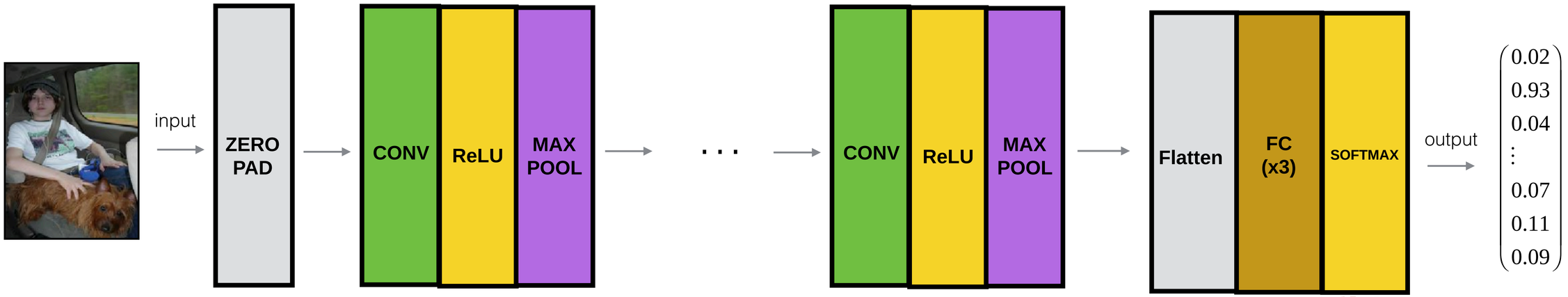
Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

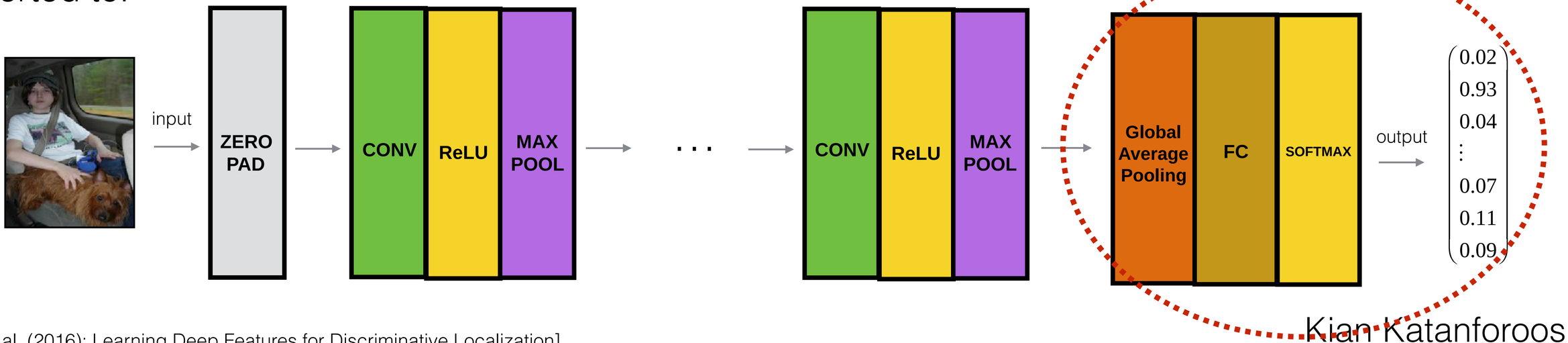
III. C. Interpreting NNs using class activation maps

Context: Along with the classification output, the zoo now wants real-time visualization of the model's decision process. You have one day. What do you do?

Using a classification network for localization:

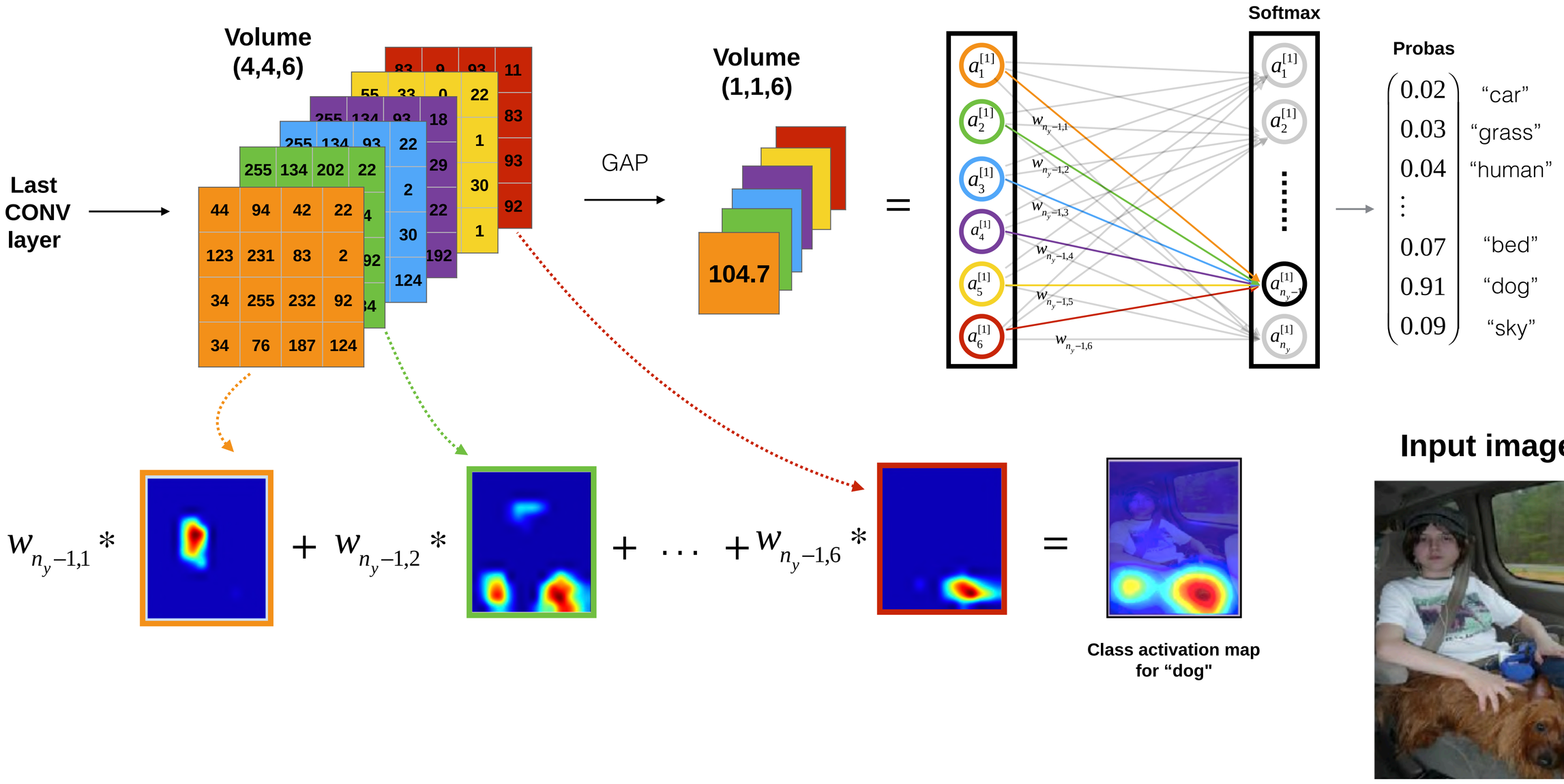


Converted to:



[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]

III. C. Interpreting NNs using class activation maps



[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]

Kian Katanforoosh

III. C. Interpreting NNs using class activation maps

[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]
Source video: Kyle McDonald

Today's outline

I. Interpreting Neural Networks' outputs

   A. with saliency maps

   B. with occlusion sensitivity

   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside

   A. with gradient ascent (class model visualization)

   B. with dataset search

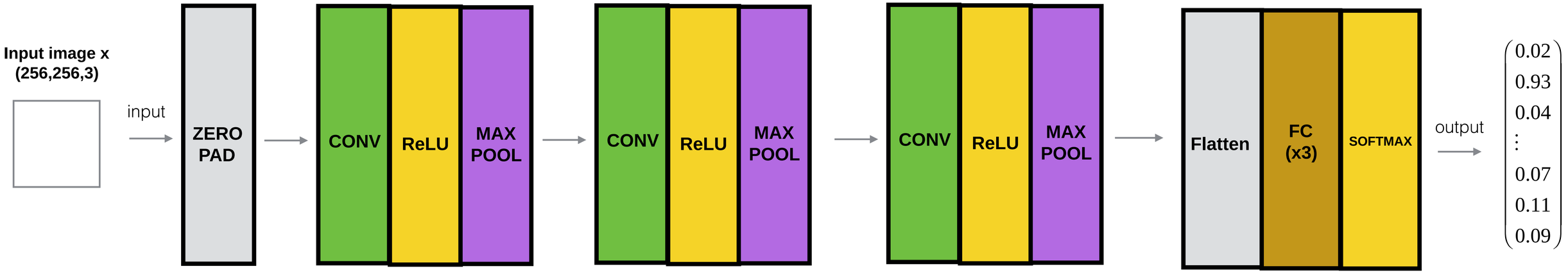
   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)

II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

Context: The zoo trusts that your model correctly locates animals. They get scared and they ask you whether the model understands what a dog is.

Given this trained ConvNet, generate an image which is representative of the class “dog” according to the ConvNet



Keep the weights fixed and use gradient ascent on the input image to maximize this loss :

$$L = s_{dog}(x) - \lambda \left|\left| x \right|\right|_2^2$$

“x should look natural”

Gradient ascent:

$$x = x + \alpha \frac{\partial L}{\partial x}$$

Repeat this process:

1. Forward propagate image x
2. Compute the objective L
3. Backpropagate to get dL/dx
4. Update x’s pixels with gradient ascent

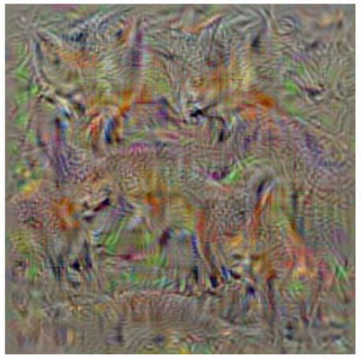
[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Kian Katanforoosh

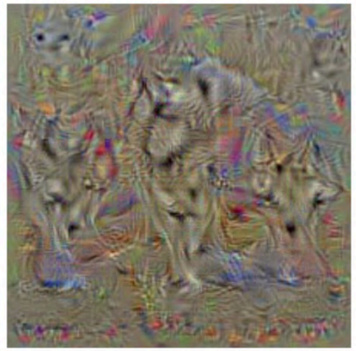
II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

goose

ostrich

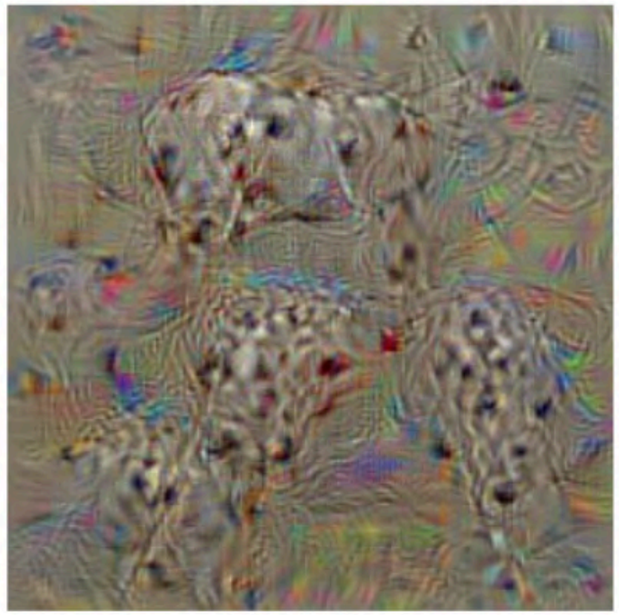


kit fox



husky

We can do this for all classes:

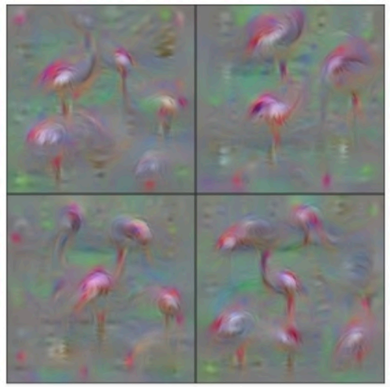


dalmatian

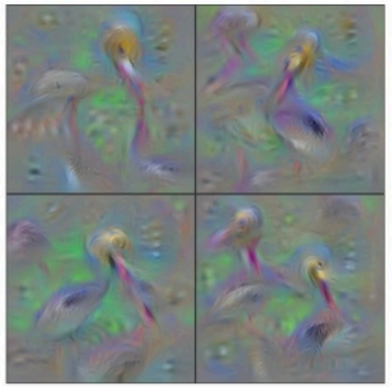
L2
Regularization

Looks better with additional regularization methods.

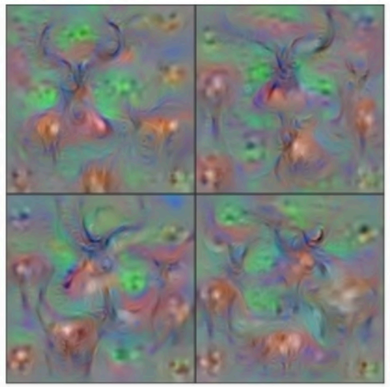
Class model visualization



Flamingo



Pelican



Hartebeest

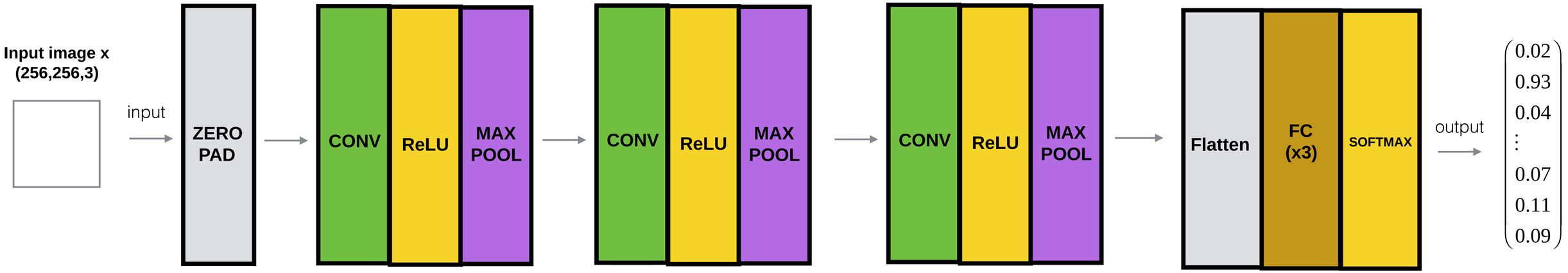
[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Kian Katanforoosh

II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

This method can be applied to any activation in the network
in order to interpret what a neuron is detecting



On the class score:

$$L = S_{dog}(x) - R(x)$$

change to

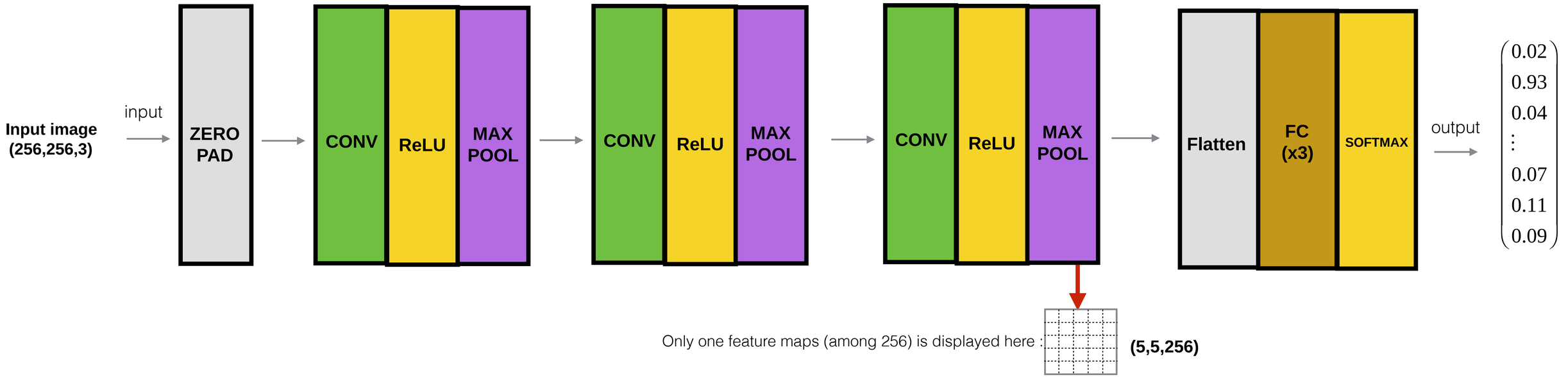
On any activation:

$$L = a_j^{[l]}(x) - R(x)$$

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

II. B. Visualizing NNs from the inside using dataset search

Context: The zoo loved the technique, and asks if there are other alternatives.



Given a filter, what examples in the dataset lead to a strongly activated feature map?

Top 5 images

It seems that the filter has learned to detect shirts

Top 5 images

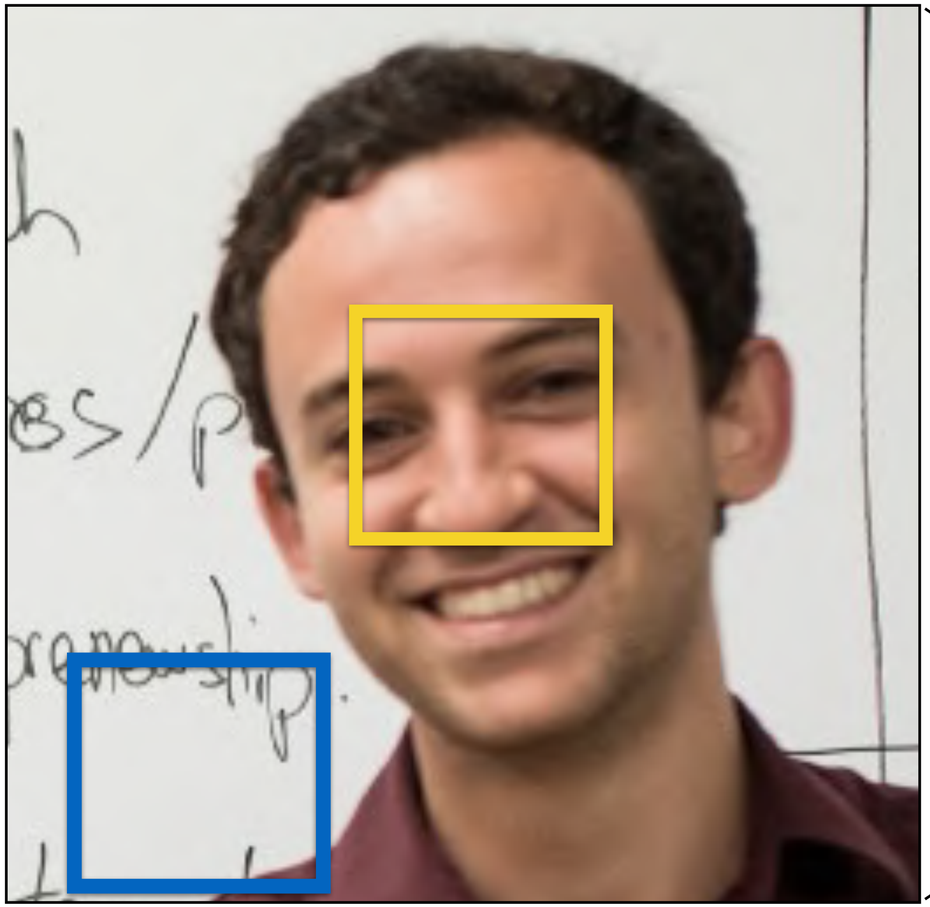
It seems that the filter has learned to detect edges

Kian Katanforoosh

II. B. Visualizing NNs from the inside using dataset search

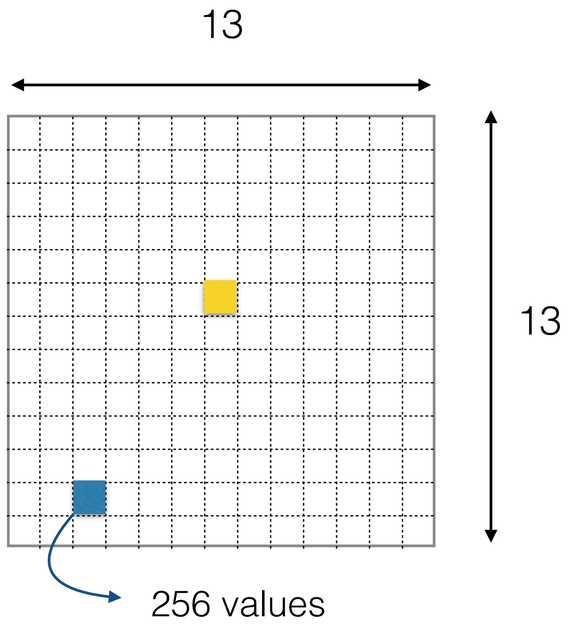
How did we crop the dataset images on the previous slide?

Input image
(64,64,3)



CONV
(5 layers)

Encoding volume
(13,13,256)



Conclusion: the deeper the activation, the more it “sees” from the image

Kian Katanforoosh

Today's outline

I. Interpreting Neural Networks' outputs

   A. with saliency maps

   B. with occlusion sensitivity

   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside

   A. with gradient ascent (class model visualization)

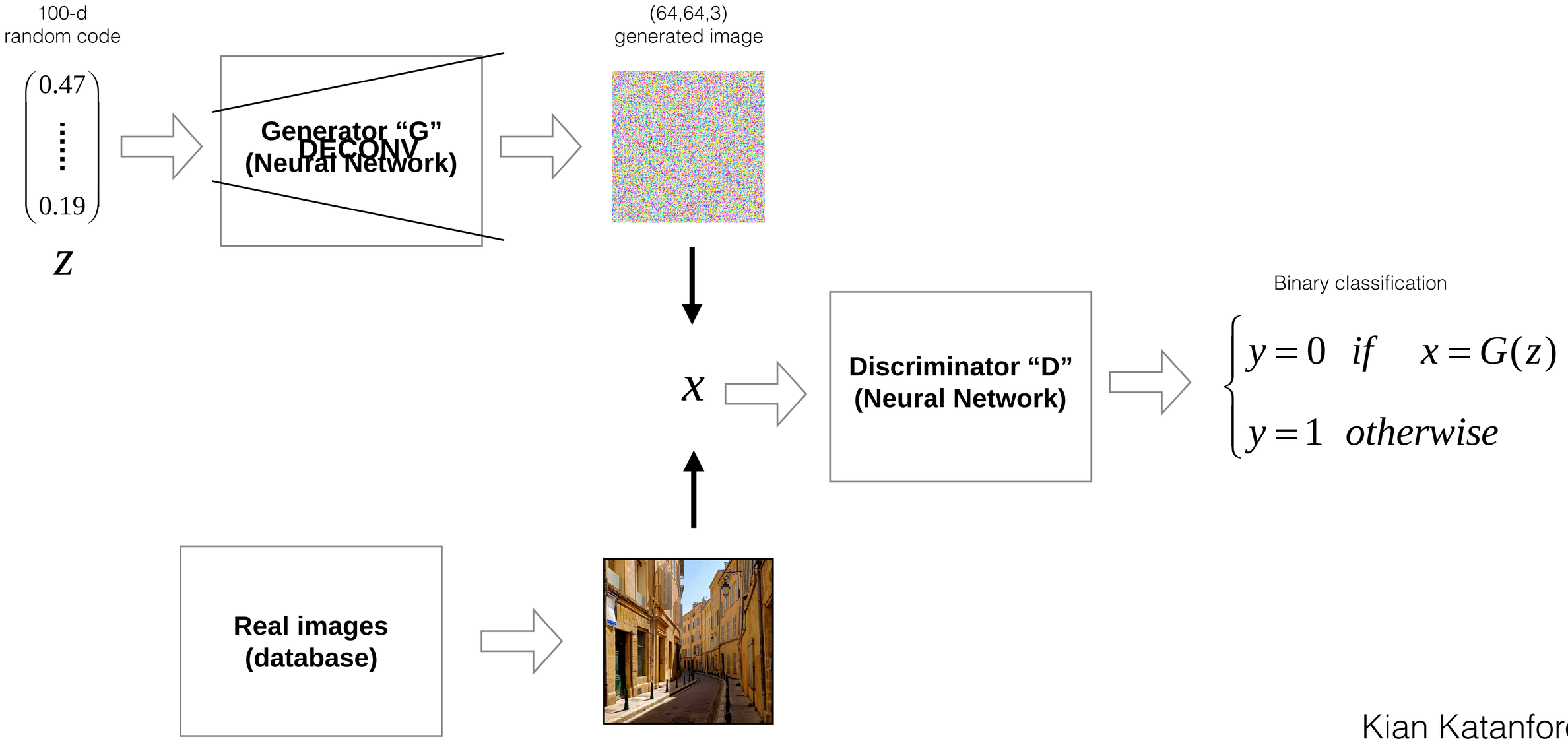
   B. with dataset search

   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)

II. C. The deconvolution and its applications

Motivation behind deconvolution/upsampling layers

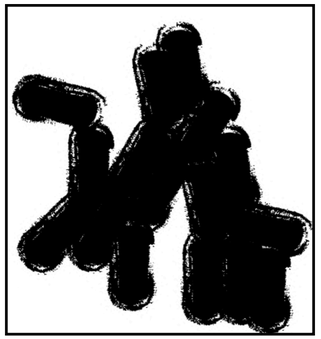


Kian Katanforoosh

II. C. The deconvolution and its applications

Motivation behind deconvolution/upsampling layers

Input image
(400, 400, 3)



Convolutions
(reduces volume height and width)

Information Encoded

Encoding

De-convolutions
(increases volume height and width)

Per-Pixel Classification
(400, 400, 1)

Get your pencils ready, we’re about to study the deconvolution.

Consider the following CONV1D:

$$n_y = \lfloor \frac{n_x - f + 2p}{s} \rfloor + 1 = \lfloor \frac{8 - 4 + 2 \times 2}{2} \rfloor + 1 = 5$$

$$x = \begin{bmatrix} 0 \\ 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_8 \\ 0 \\ 0 \end{bmatrix} \longrightarrow \boxed{\begin{array}{c}\text{CONV 1D} \\ \text{1 filter of size 4} \\ \text{stride = 2}\end{array}} \longrightarrow y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Let's define our filter as:

$$f = (w_1, w_2, w_3, w_4)$$

The system of equations is:

$$\begin{aligned} y_1 &= w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot x_1 + w_4 \cdot x_2 \\ y_2 &= w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 \\ &\vdots \\ y_5 &= w_1 \cdot x_7 + w_2 \cdot x_8 + w_3 \cdot 0 + w_4 \cdot 0 \end{aligned}$$

Let's rewrite the system of equations:

$$
\begin{aligned}
y_1 &= w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot x_1 + w_4 \cdot x_2 \\
y_2 &= w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 \\
&\vdots \\
y_5 &= w_1 \cdot x_7 + w_2 \cdot x_8 + w_3 \cdot 0 + w_4 \cdot 0
\end{aligned}
$$

$$
\underset{(5,1)}{y} = \underset{(5,12)}{W} \underset{(12,1)}{x}
$$

1D CONV can be rewritten as a matrix vector multiplication!

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & w_4 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 & w_4 & 0 & \dots \\ & & & & & & & & & \vdots \\ 0 & \dots & \dots & 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 & w_4 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_8 \\ 0 \\ 0 \end{bmatrix}
$$

Kian Katanforoosh

1D CONV

$$y = Wx$$

(5,1) (5,12) (12,1)

1D DECONV If W is invertible, then $\exists H = W^{-1}$ such that $x = Hy$

(12,1) (12,5) (5,1)

In practice, we would even assume that W is orthogonal, i.e. $W^{-1} = W^T$

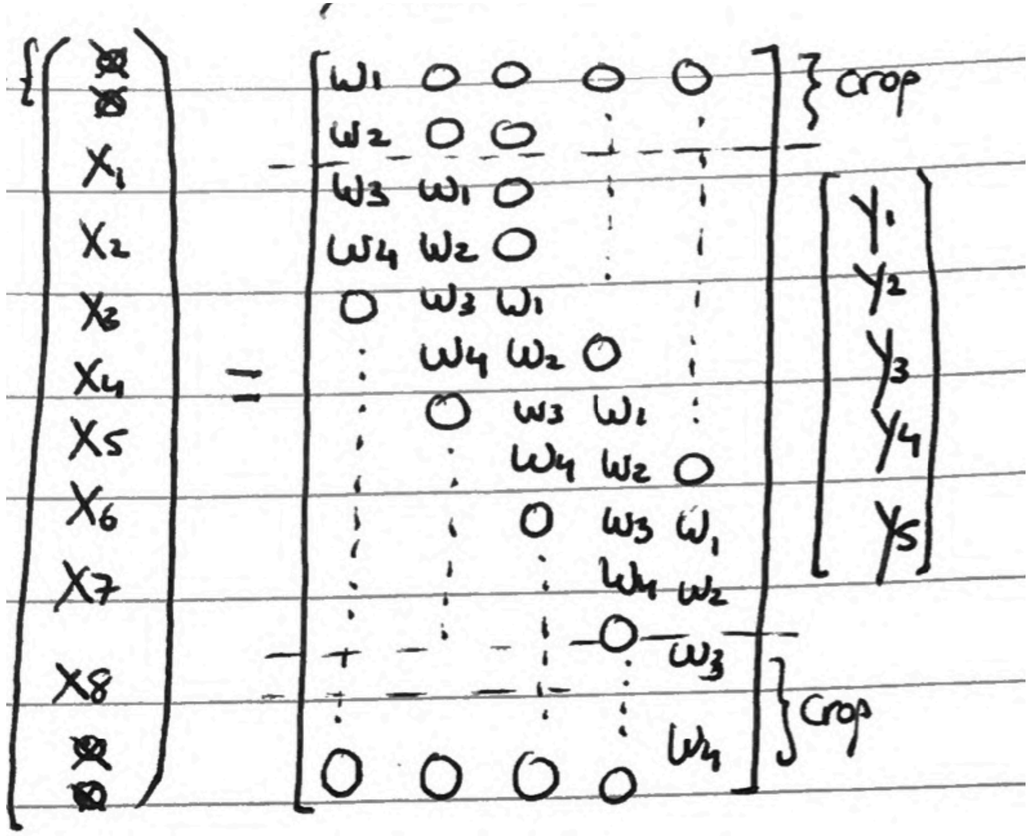
For example $(w_1, w_2, w_3, w_4) = (-1,0,0,0,1)$

Thus: $x = W^T y$ Deconvolution ~ Transposed convolution

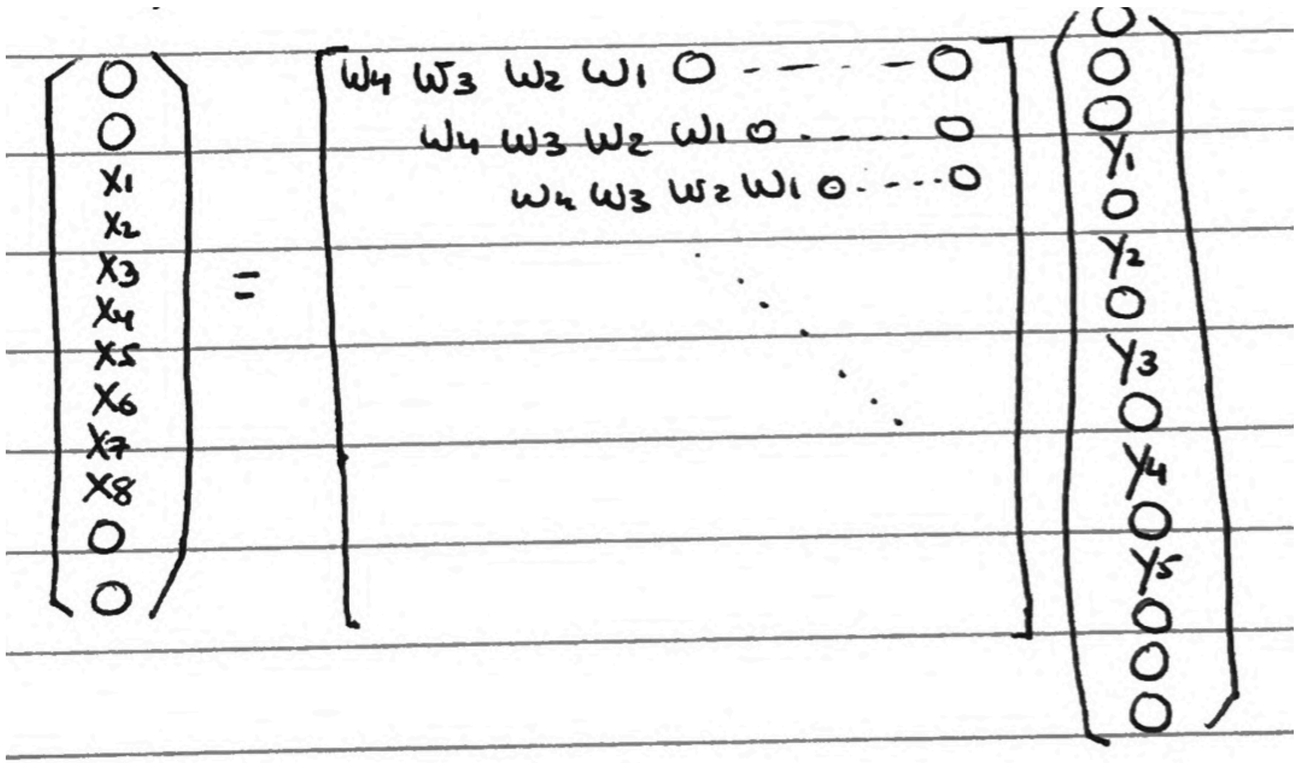
Kian Katanforoosh

Let's rewrite: $x = W^T y$

Transposed convolution with stride 2



Sub-pixel convolution with stride 1/2



Kian Katanforoosh

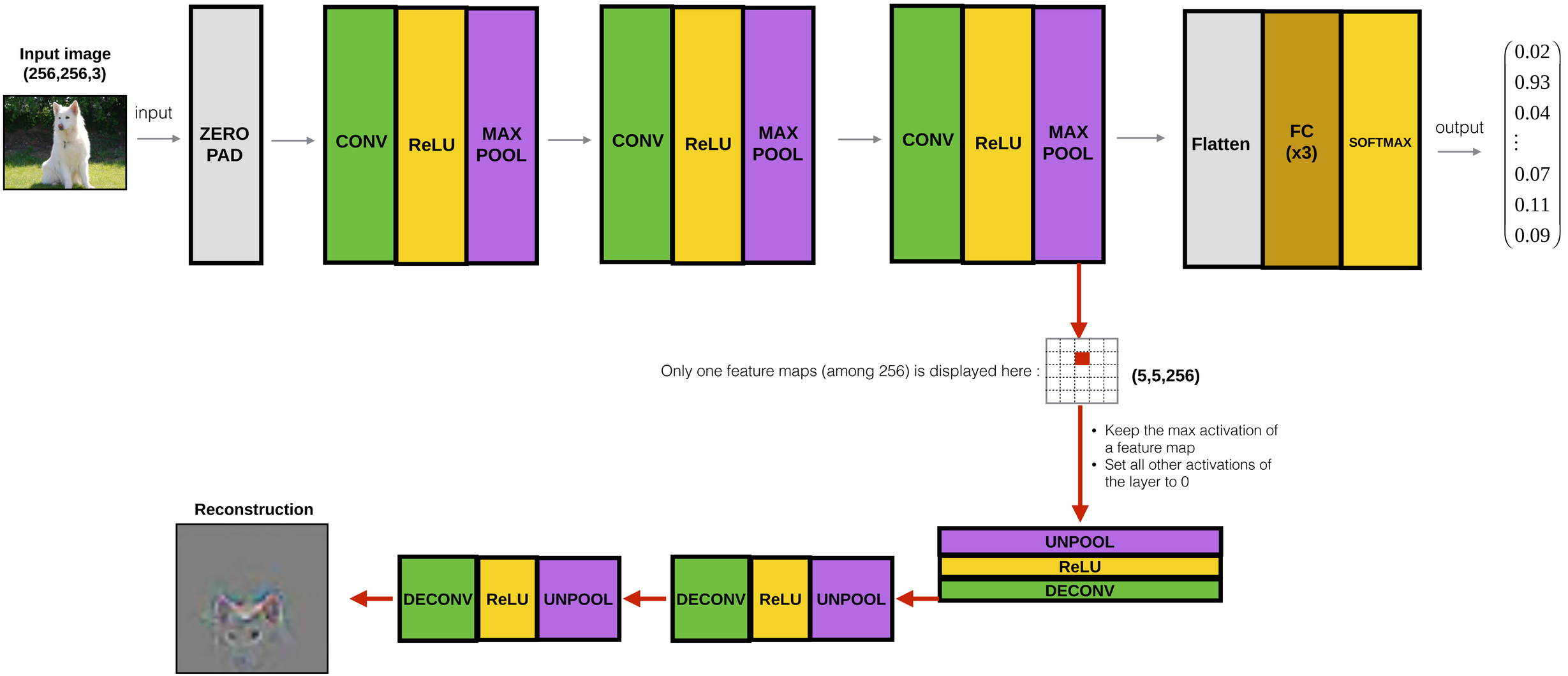
What to remember?

Implementing a deconvolution (sub-pixel version) is similar to the convolution, with:

1. Create a sub-pixel version of the input (i.e., insert zeros and pad)
2. Flip the filters.
3. Divide the stride by 2.

III. A. Interpreting NNs using deconvolutions

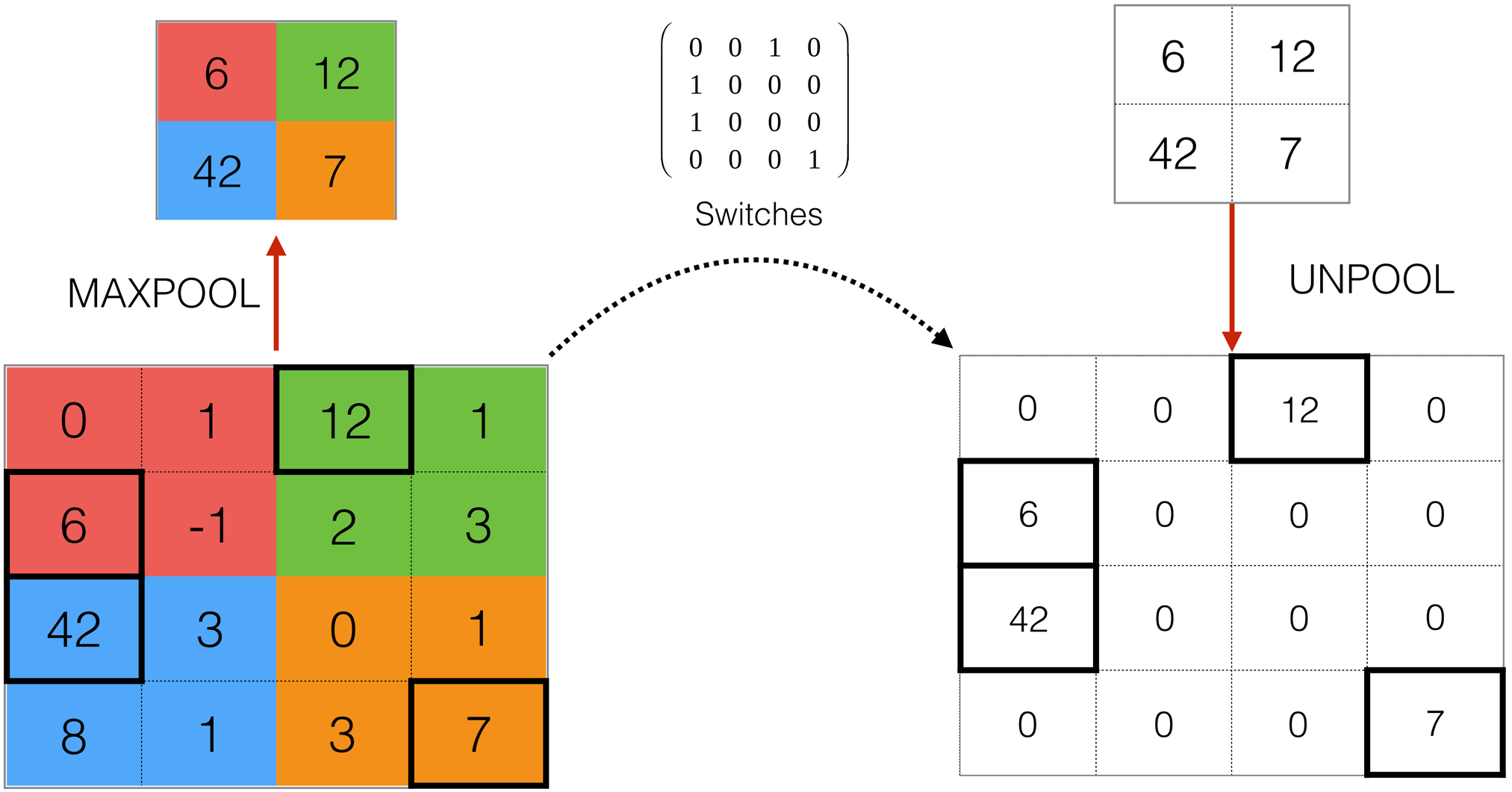
Motivation of DeconvNets for visualization: Here is a CNN, trained on ImageNet (1.3m images, 1000 classes), we're trying to interpret by reconstructing the activation's zone of influence in the input space.



[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

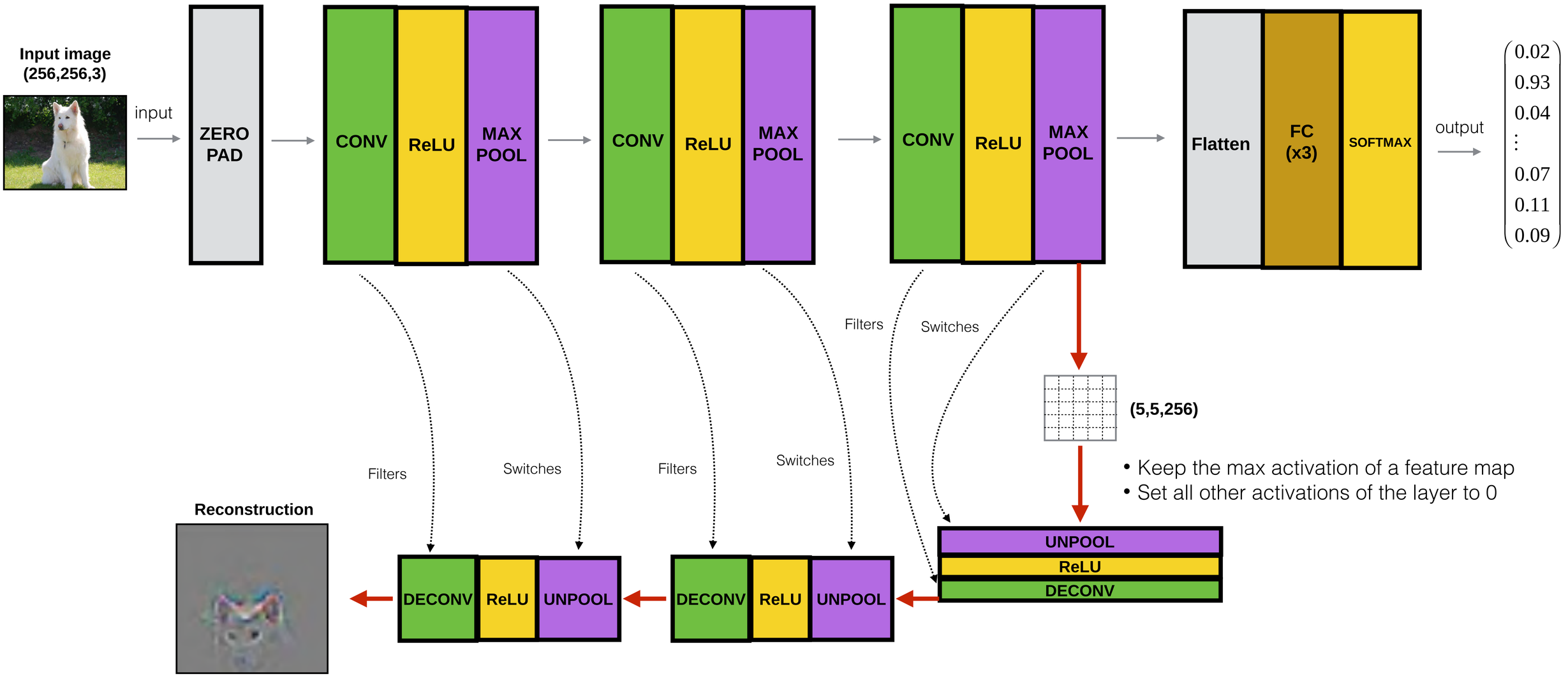
II. C. The deconvolution and its applications



Kian Katanforoosh

II. C. The deconvolution and its applications

We need to pass the filters and switches from the ConvNet to the DeconvNet.



[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

II. C. The deconvolution and its applications

$$a^{[l]} = \mathrm{I}\{a^{[l+1]} \geq 0\}.a^{[l+1]}$$

1	0	3	0
4	0	0	0
30	2	0	1
1	0	0	7

“ReLU forward”

1	-2	3	-4
4	-1	-1	-2
30	2	0	1
1	-2	-9	7

Switches

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

-2	1	-31	-3
3	4	2	14
-2	12	4	1
10	10	2	1

“ReLU backward”

-2	0	-31	0
3	0	0	0
-2	12	0	1
10	0	0	1

-2	1	-31	-3
3	4	2	14
-2	12	4	1
10	10	2	1

“ReLU DeconvNet”

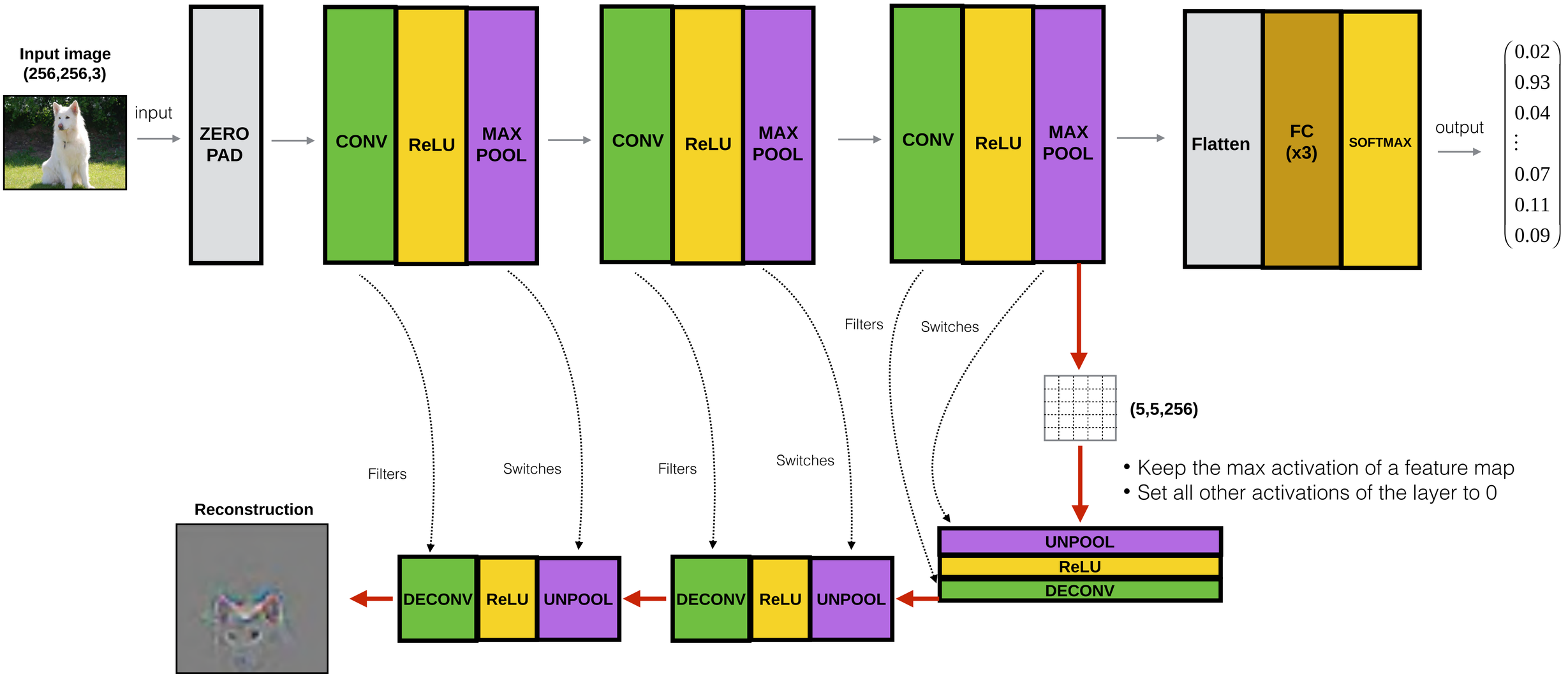
0	1	0	0
3	4	2	14
0	12	4	1
10	10	2	1

[Sringenberg & Dosovitskiy et al. (2015): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

II. C. The deconvolution and its applications

We need to pass the filters and switches from the ConvNet to the DeconvNet.

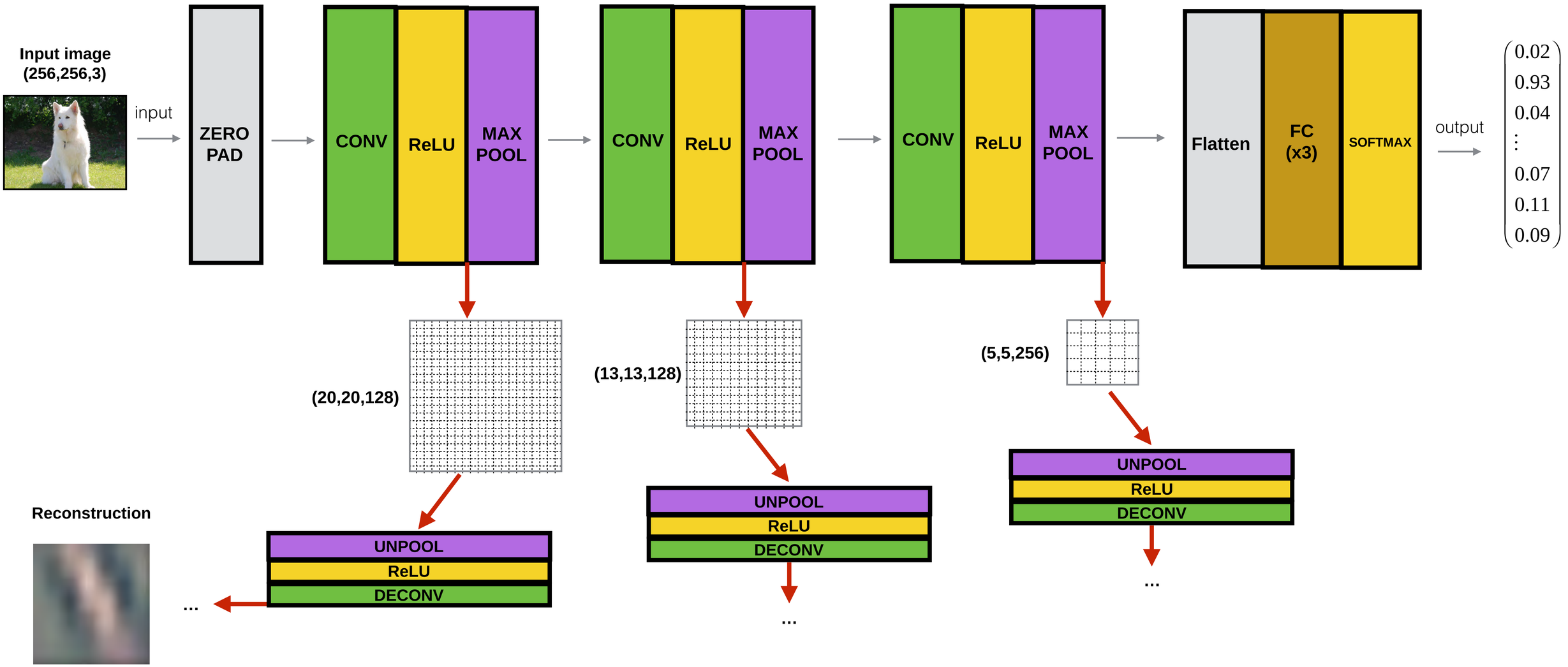


[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

II. C. The deconvolution and its applications

Other CONV layers can be visualized the same way

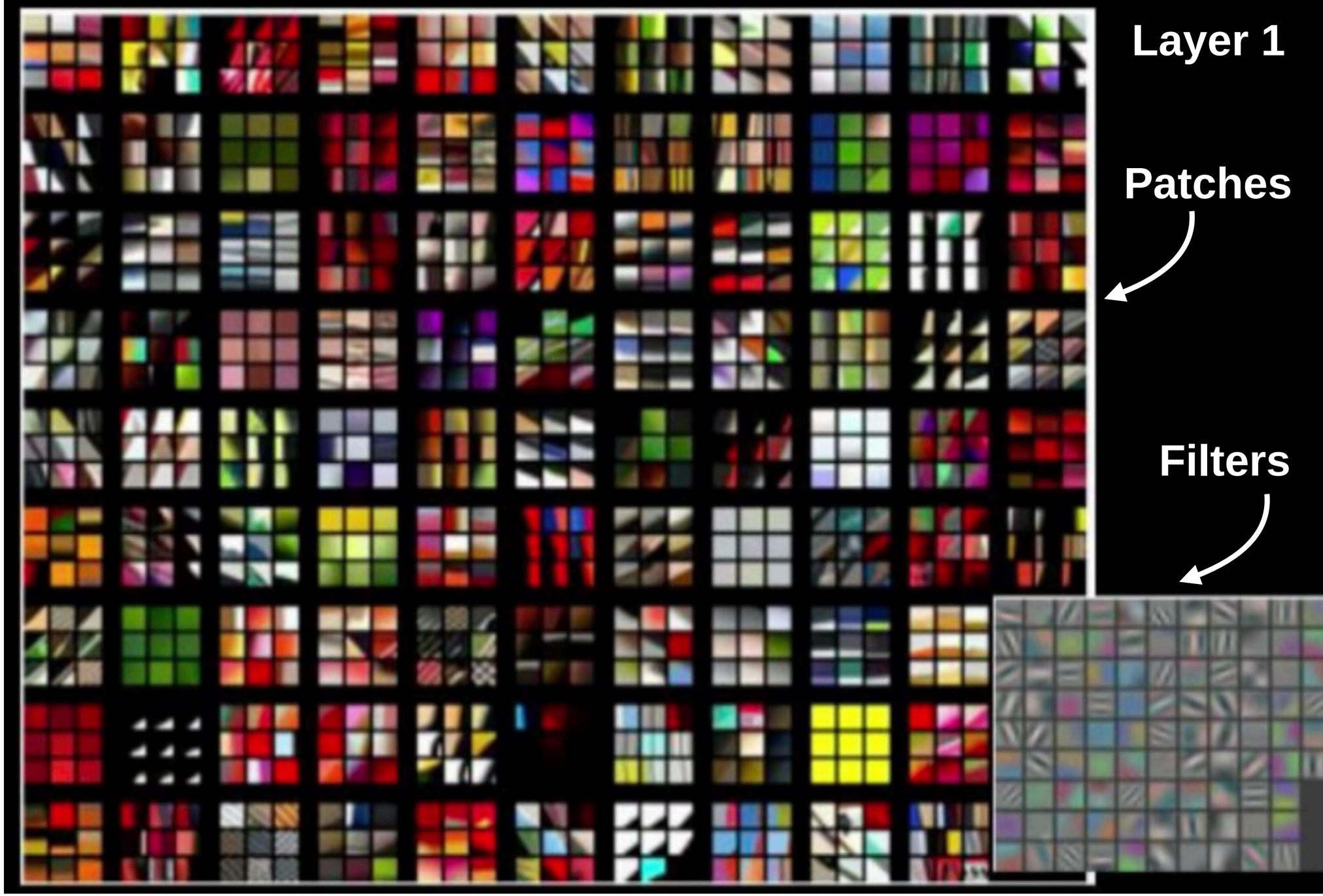


[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

Results on a validation set of 50,000 images

- Top-9 strongest activations per filter in the 1st layer
- Because we know the position of the activation and all the pooling switches we can crop the part of the image that fired the activation



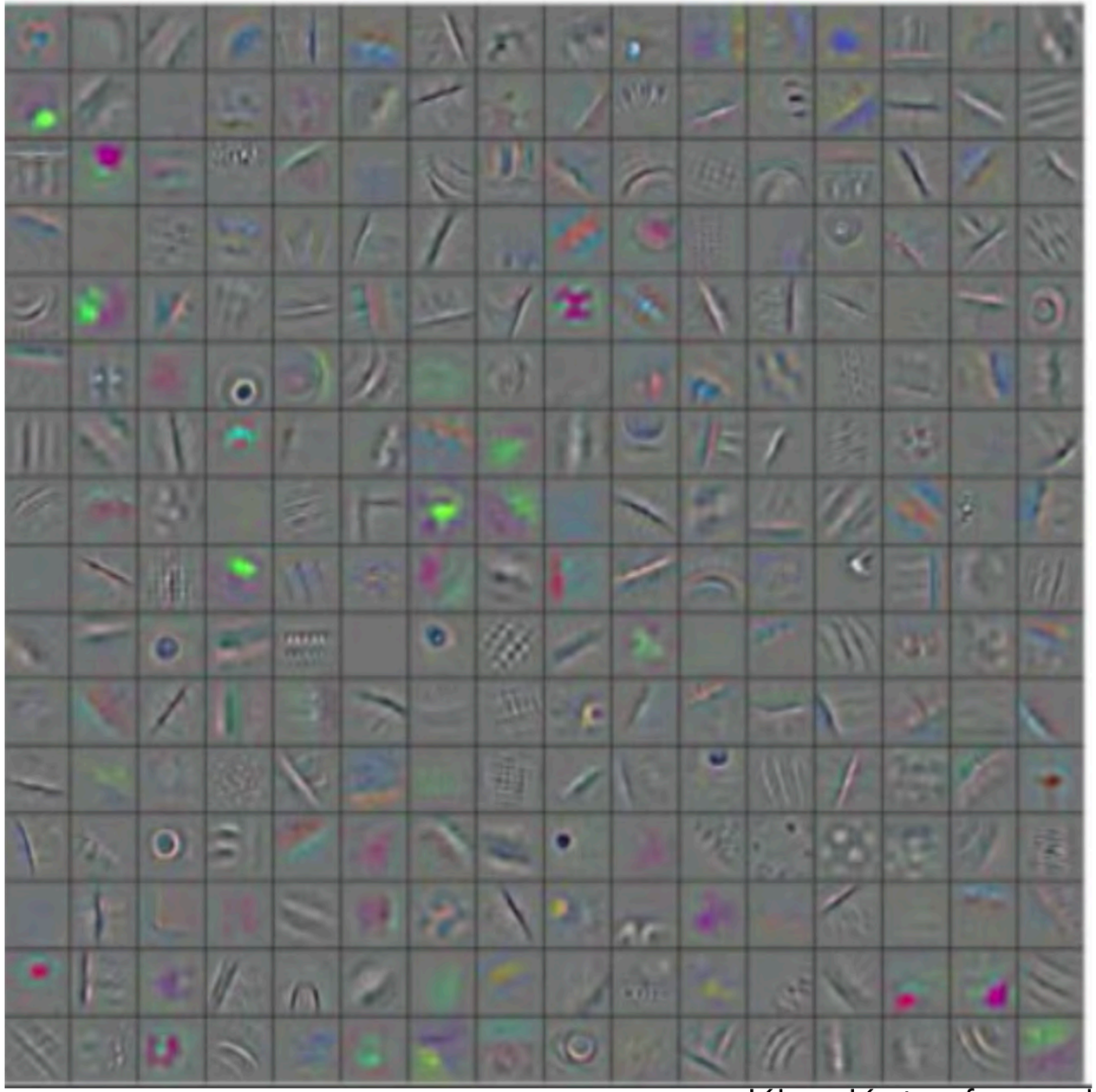
Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Results on a validation set of 50,000 images

- Learning a more complex set of patterns than 1st layer edges.
- Covers a much larger space of the image because of the pooling layer before.
- Top-1 strongest activation per feature map in the 2nd layer (256 feature maps.)

Layer 2 reconstructions (deconv)

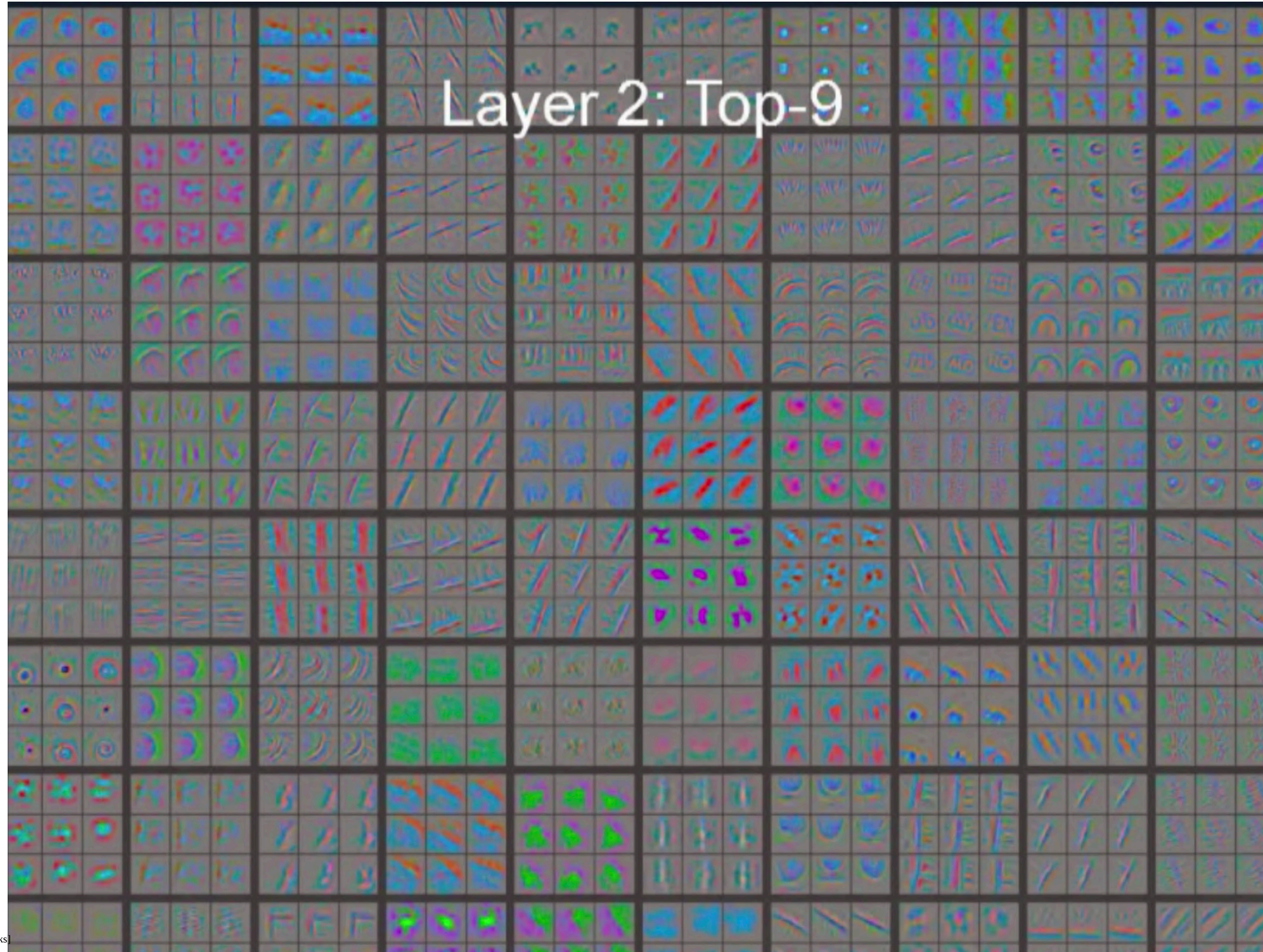


Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Results on a validation set of 50,000 images

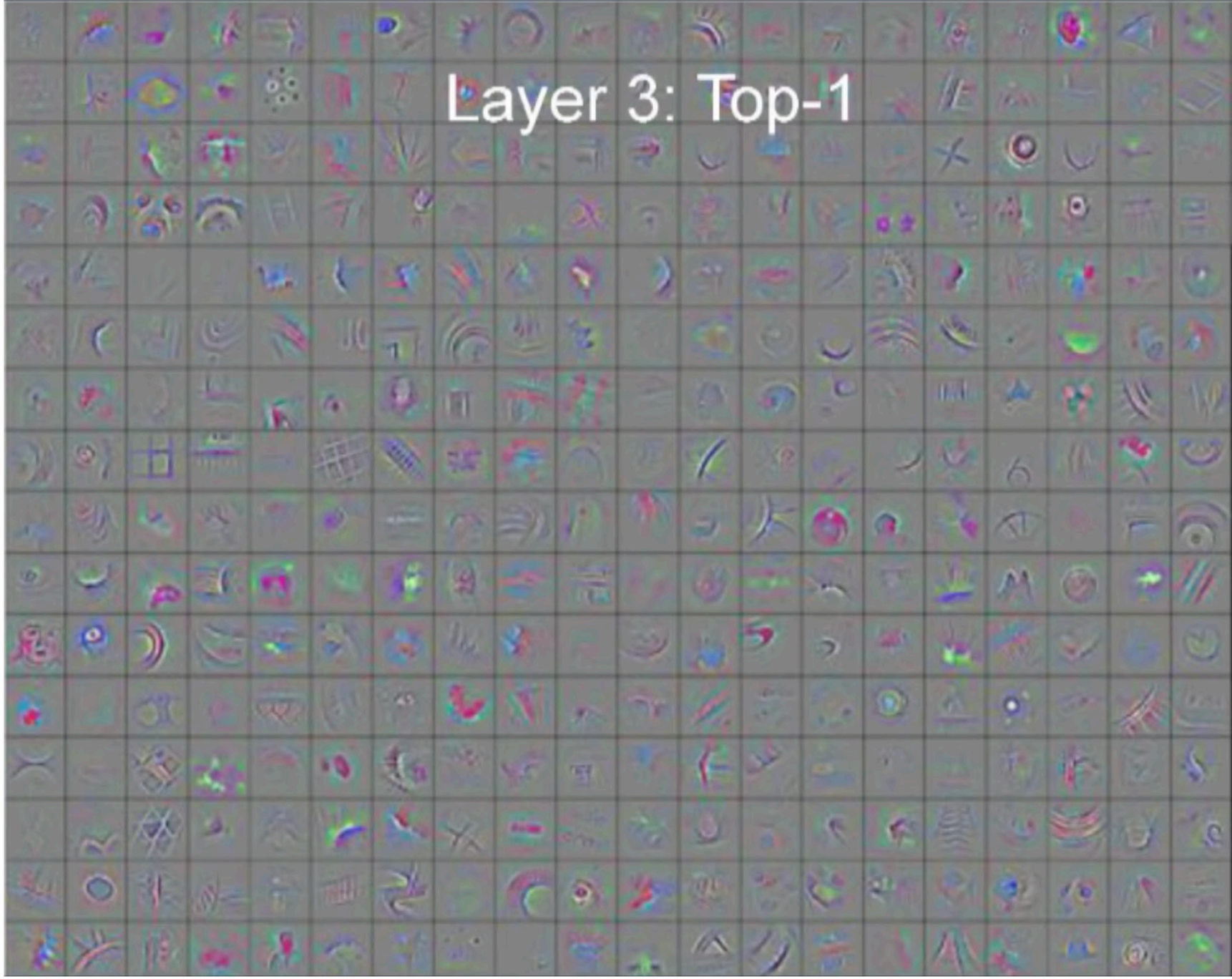
- Learning a more complex set of patterns than 1st layer edges
- Covers a much larger space of the image probably because of the pooling layer before.
- Features are more invariant to small changes. Ex: A dot, spiral, circle all fire the same 2nd layer feature very strongly



[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Results on a validation set of 50,000 images

- 3rd layer: increased complexity
- An activated neuron is seeing ≈80x80 part of a 256x256 image
- Learning objects, faces etc..



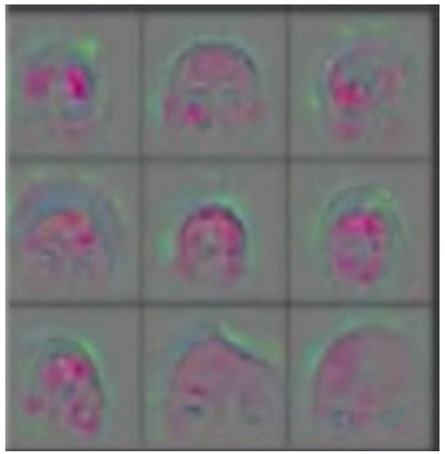
Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

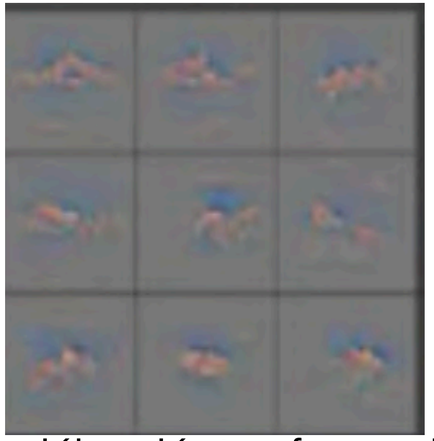
Results on a validation set of 50,000 images

- 3rd layer: increased complexity
- An activated neuron is seeing ≈80x80 part of a 256x256 image
- Learning objects, faces etc..
- Patches: Semantic grouping, not structural

Faces

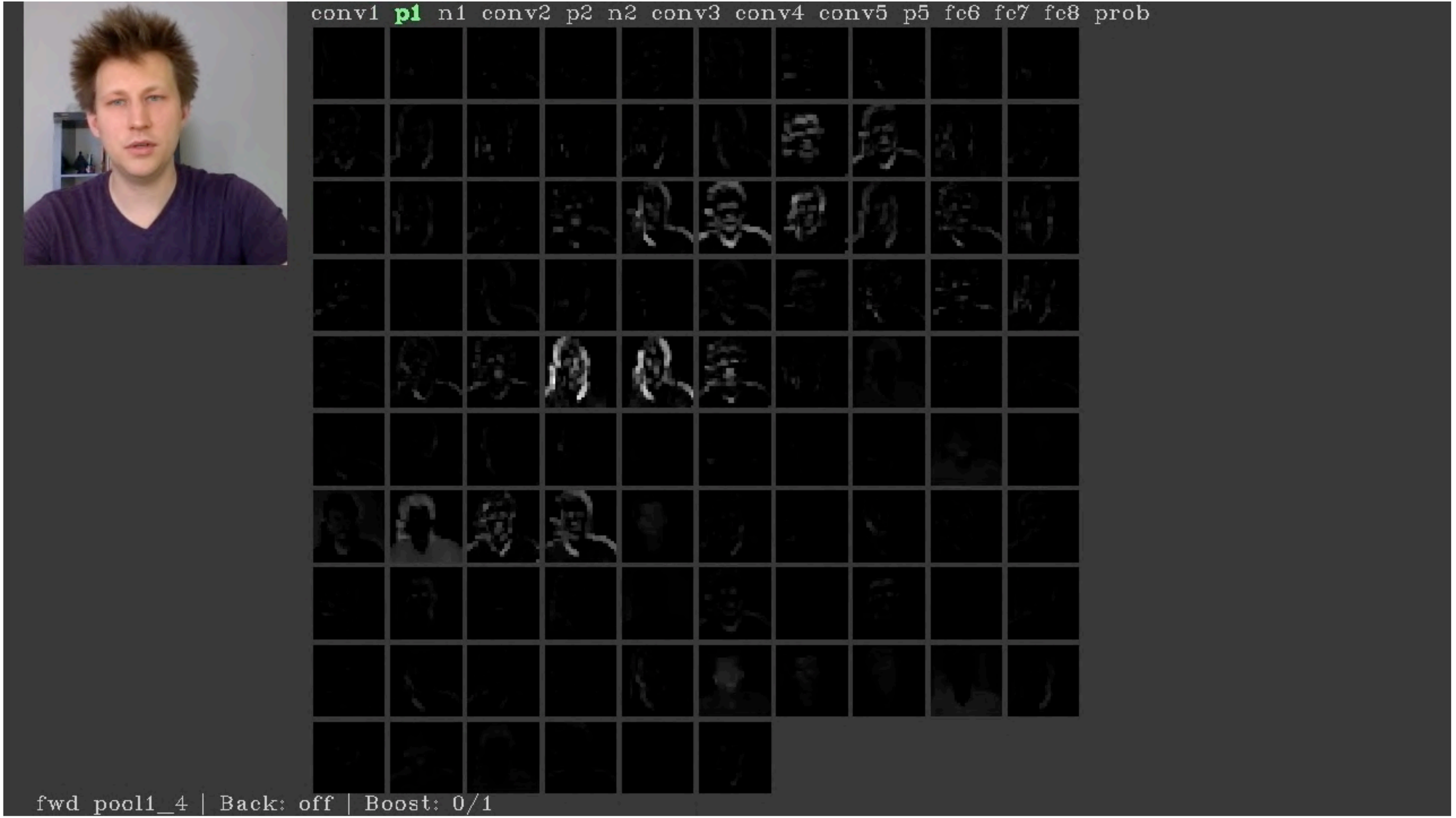


Clouds



Kian Katanforoosh

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]



[Link to video: https://www.youtube.com/watch?v=AgkfIQ4IGaM]
[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

Today's outline

I. Interpreting Neural Networks' outputs

   A. with saliency maps

   B. with occlusion sensitivity

   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside

   A. with gradient ascent (class model visualization)

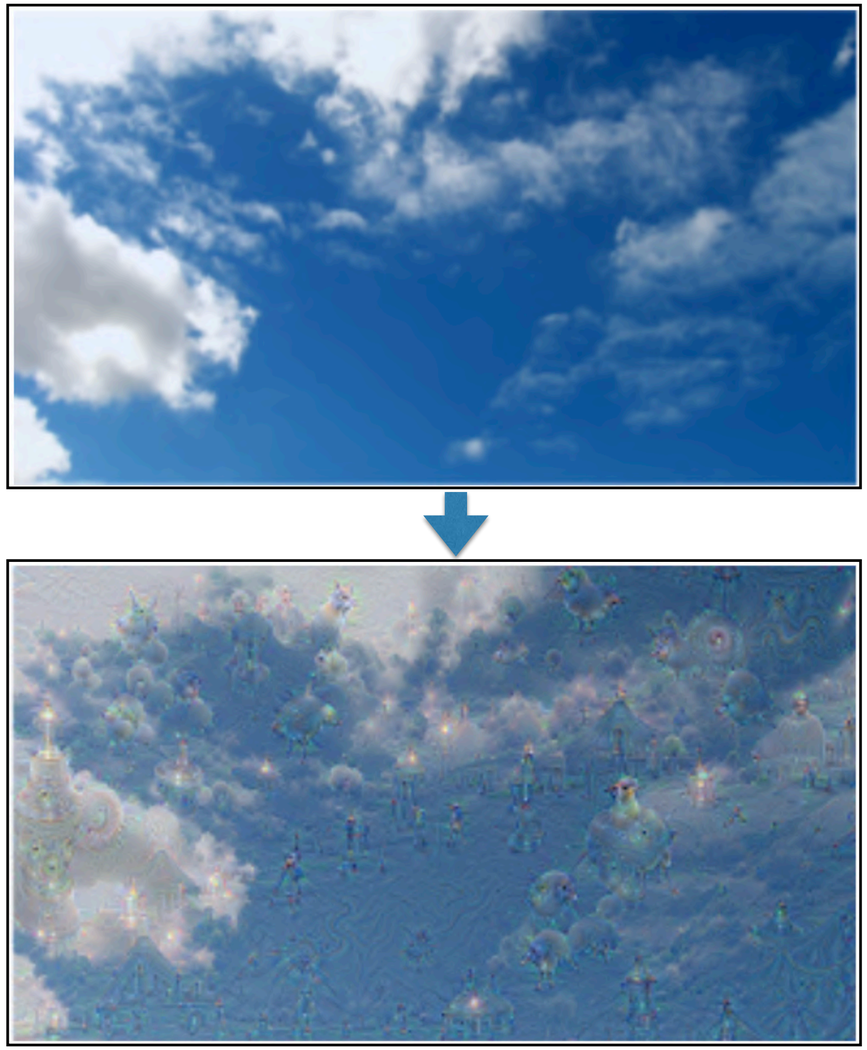
   B. with dataset search

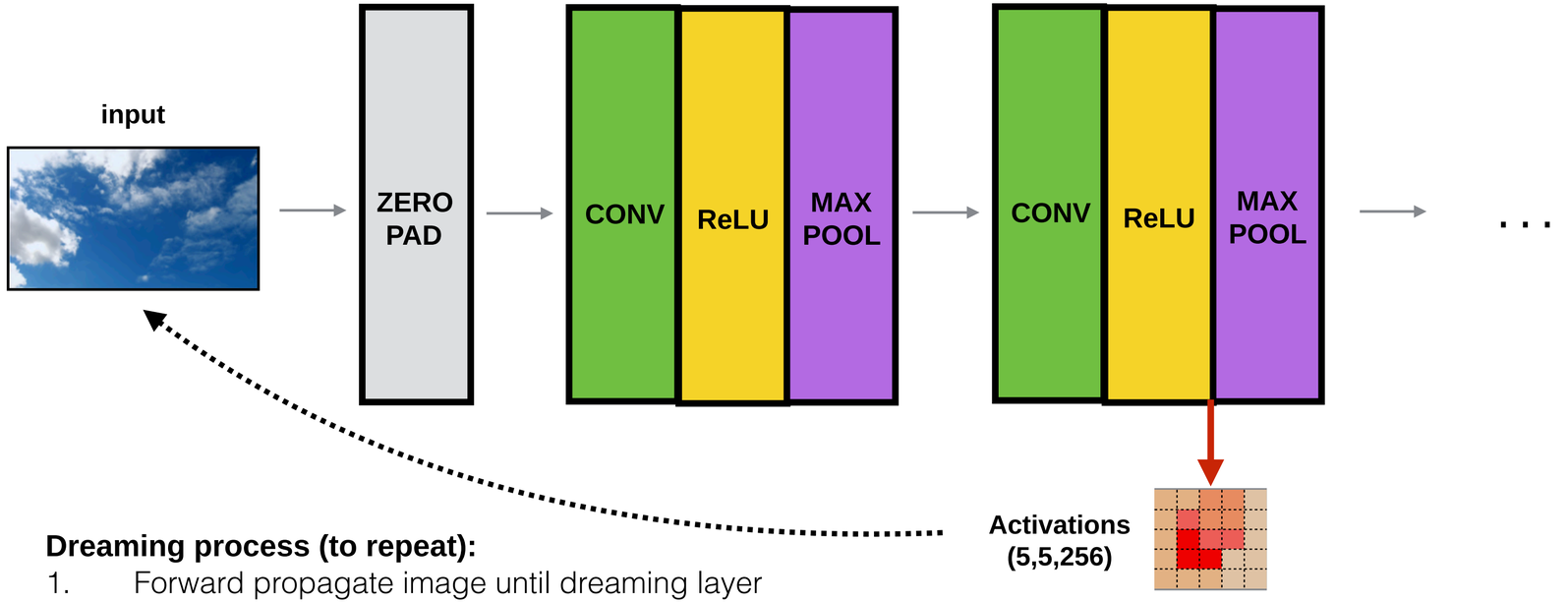
   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)

III. Deep Dream: going deeper in NNs

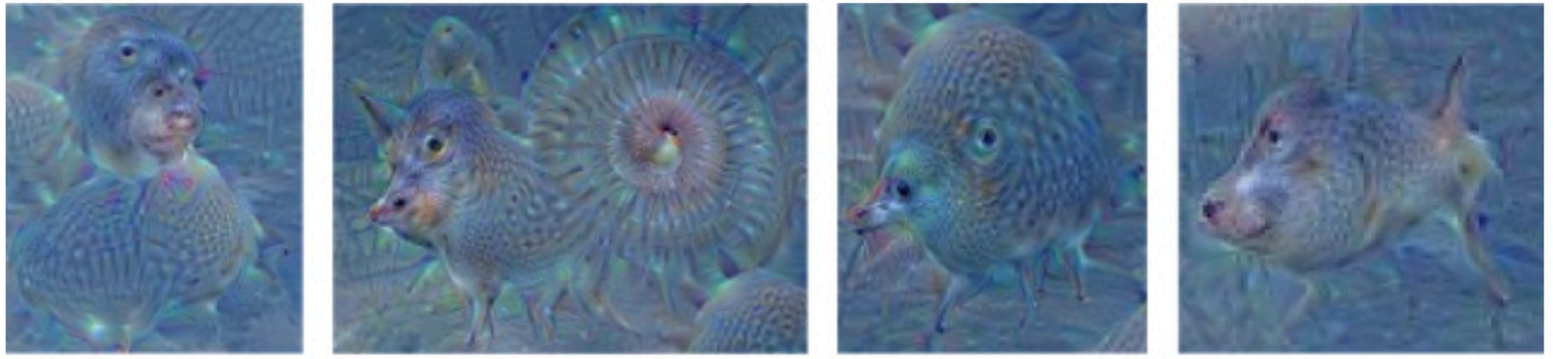
How to boost the activation of a neuron?





Dreaming process (to repeat):

1. Forward propagate image until dreaming layer
2. **Set gradients of dreaming layer to be equal to its activations**
3. Backpropagate gradients to input image
4. Update Pixels of the image



[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

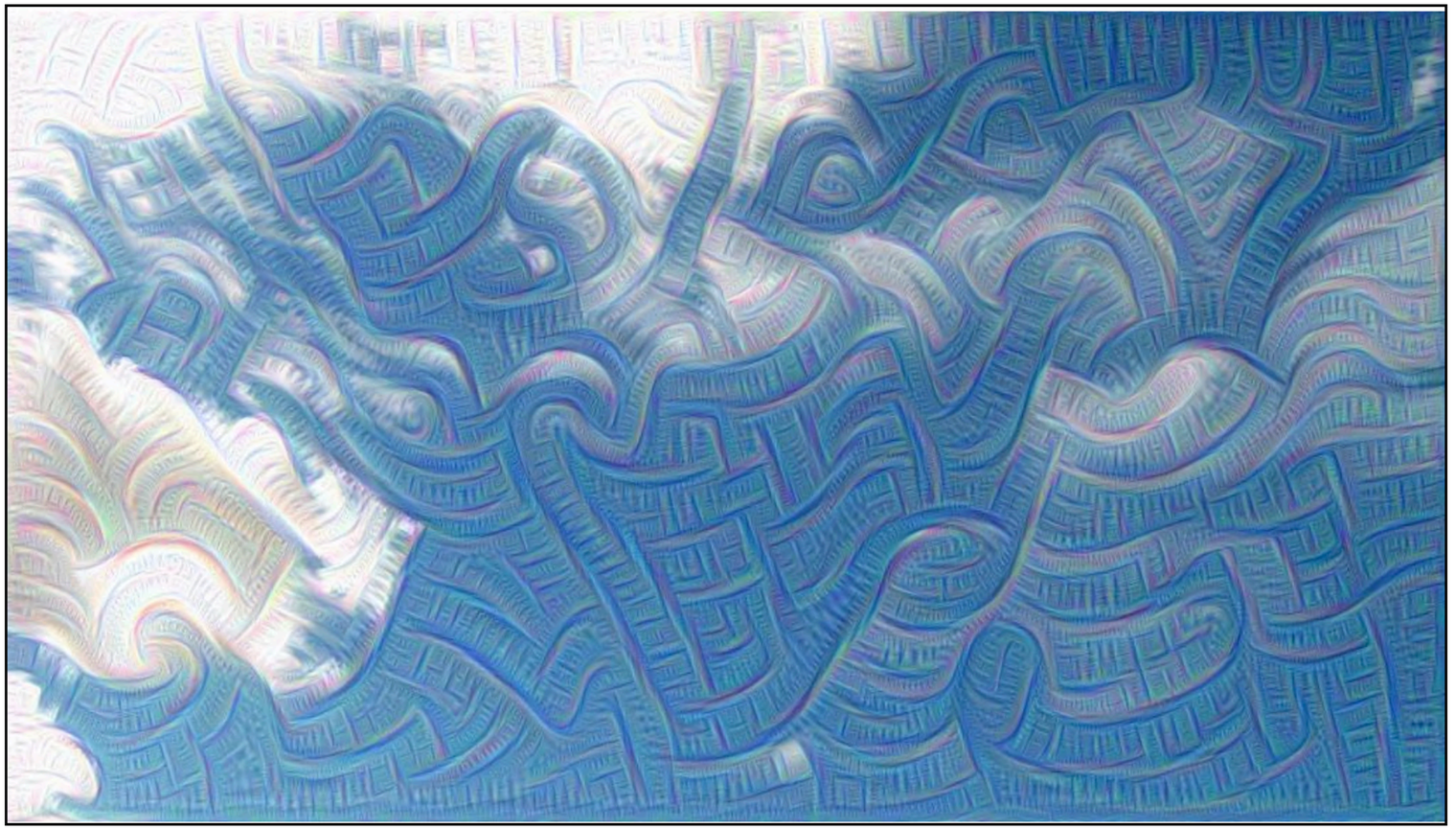
Kian Katanforoosh

III. Deep Dream: going deeper in NNs

[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

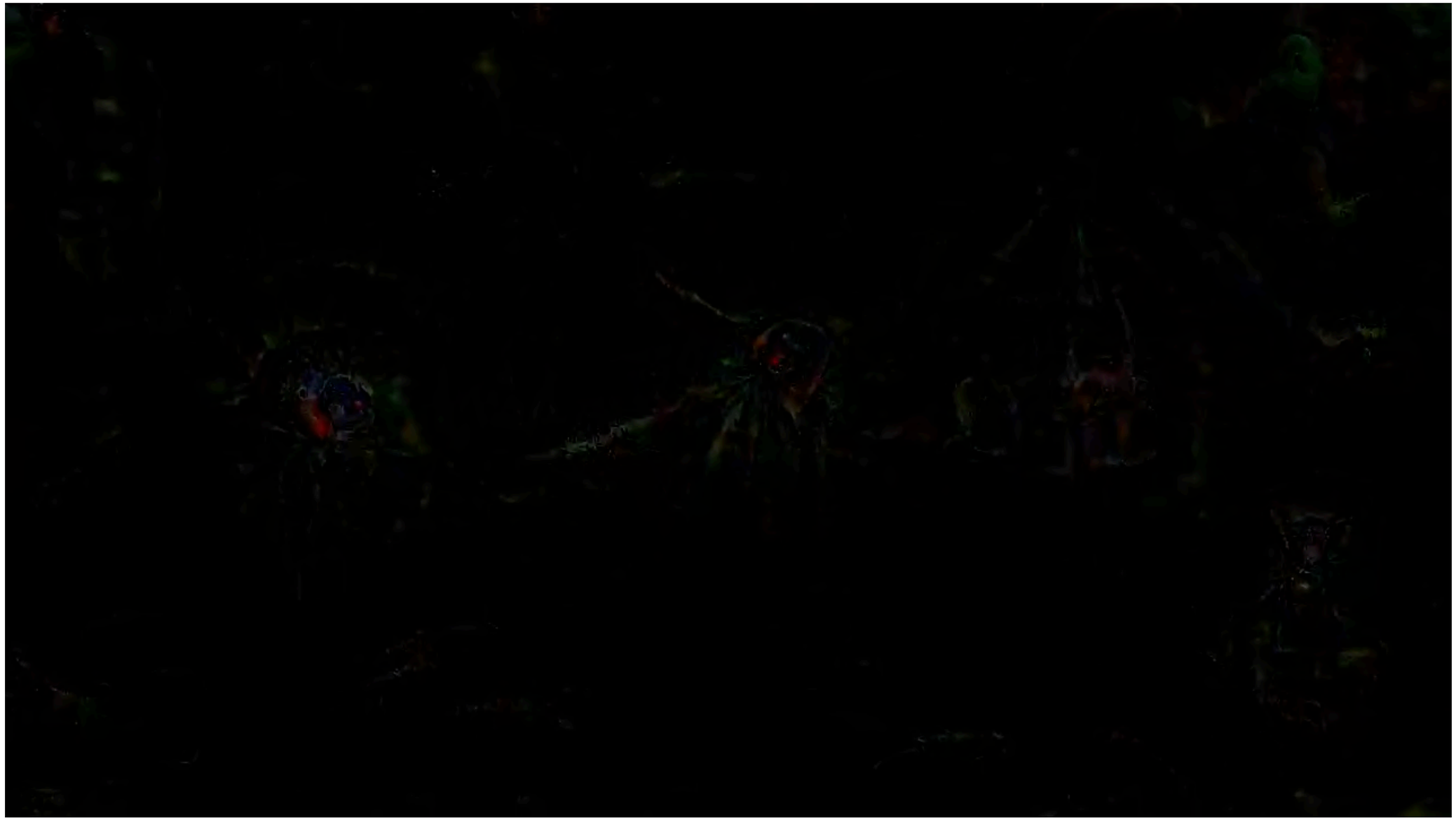
III. Deep Dream: going deeper in NNs

If you dream in lower layers:



[Github repo (deepdream): https://github.com/google/deepdream/]
[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

III. Deep Dream: going deeper in NNs



[Link to video: https://www.youtube.com/watch?v=DgPaCWJL7XI]
[Github repo (deepdream): https://github.com/google/deepdream/]
[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

III. Deep Dream

Did the neural network learned the right features to detect an object?

Dumbbells

the network failed to understand the essence of a dumbbell

[Github repo (deepdream): https://github.com/google/deepdream/]
[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

Questions we are now able to answer:

- *What part of the input is responsible for the output?*
    - *Occlusion sensitivity*
    - *Class Activation Maps*

- *What is the role of a given neuron/filter/layer?*
    - *Deconvolutions can help visualize the role of a neuron*
    - *Search dataset images maximizing the activation*
    - *Gradient ascent (class model visualization)*

- *Can we check what the network focuses on given an input image?*
    - *Occlusion sensitivity*
    - *Saliency maps (one-time gradient ascent)*
    - *Class Activation Maps*

- *How does a neural network see our world?*
    - *Gradient ascent (class model visualization)*
    - *Deep Dream*

- *Do these visualization have use cases?*
    - *Segmentation (saliency maps)*
    - *Art (Deep Dream)*

Duties for next week

Meet with your assigned TA between 1/28 and 2/26 to discuss your milestone report.

Project Milestone due 02/26 Friday 11:59 PM

Completed modules for 03/04:

- C5M1: Recurrent Neural Networks (slides)

Quizzes (due at 8 30am PST):

- Recurrent Neural Networks

Programming Assignments (due at 8 30am PST):

- Building a Recurrent Neural Network - Step by Step
- Dinosaur Land -- Character-level Language Modeling
- Jazz improvisation with LSTM