Today's Focus

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- Appreciating CNNs
- Visualizing CNNs
- Representation and Practice
- Training CNNs
- Training RNNs

Appreciation of Convolutional Neural Networks

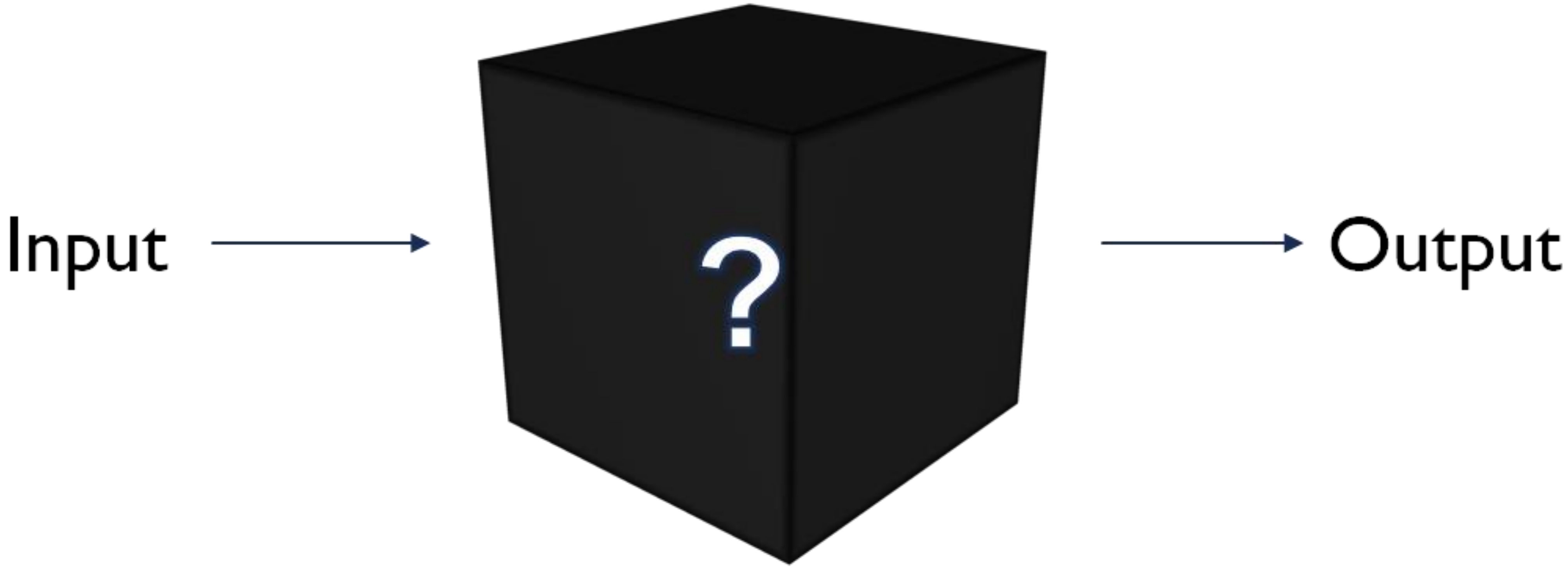
Visualization, Interpretation

Recap: CNNs

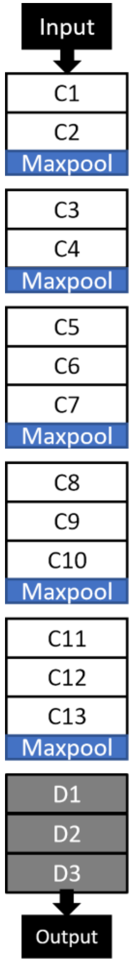
Recap: CNNs

Recap: CNNs

What goes on inside a convnet?



CNN Visualizations



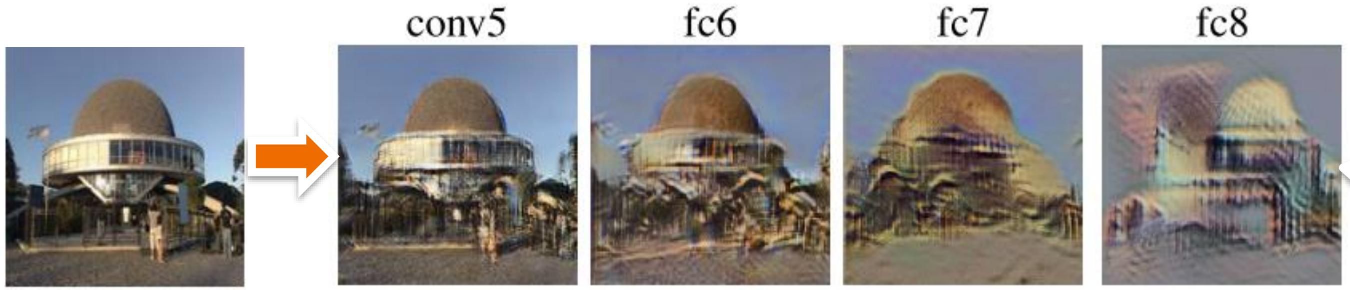
Saliency

What is in this image?

Cat Dog

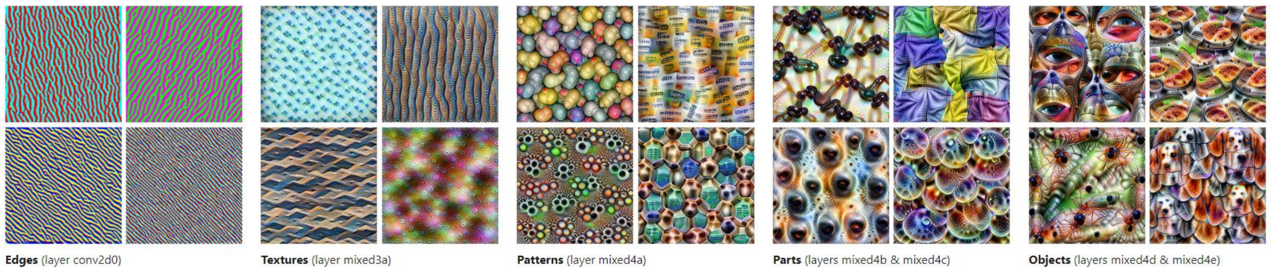
Which part of the image explains the classification?

Feature Inversion



How much information is retained/discarded at each layer of a deep network?

Feature visualization



What pattern does each convolutional filter search for?

Monitoring training

epochs

How well is the CNN getting trained?

- Activation maximization
- Deep Dream
- Feature inversion
- Saliency visualization
- Statistical methods

How do we study what happens inside convnets

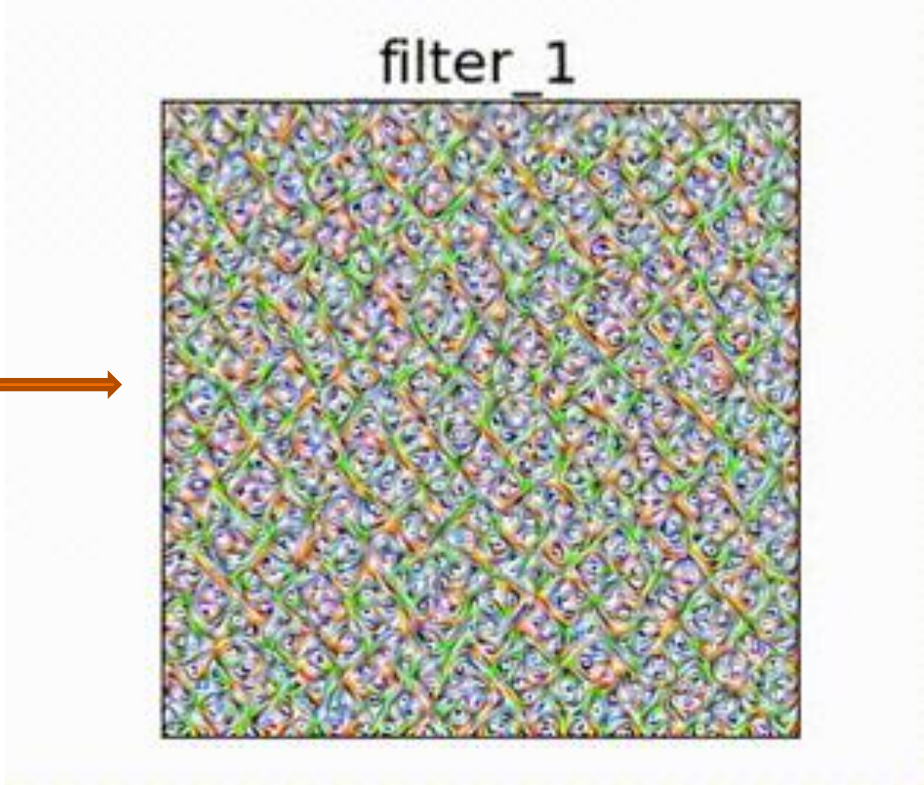
LOOK INSIDE CONVNETS

ACTIVATION MAXIMIZATION

Activation Maximization

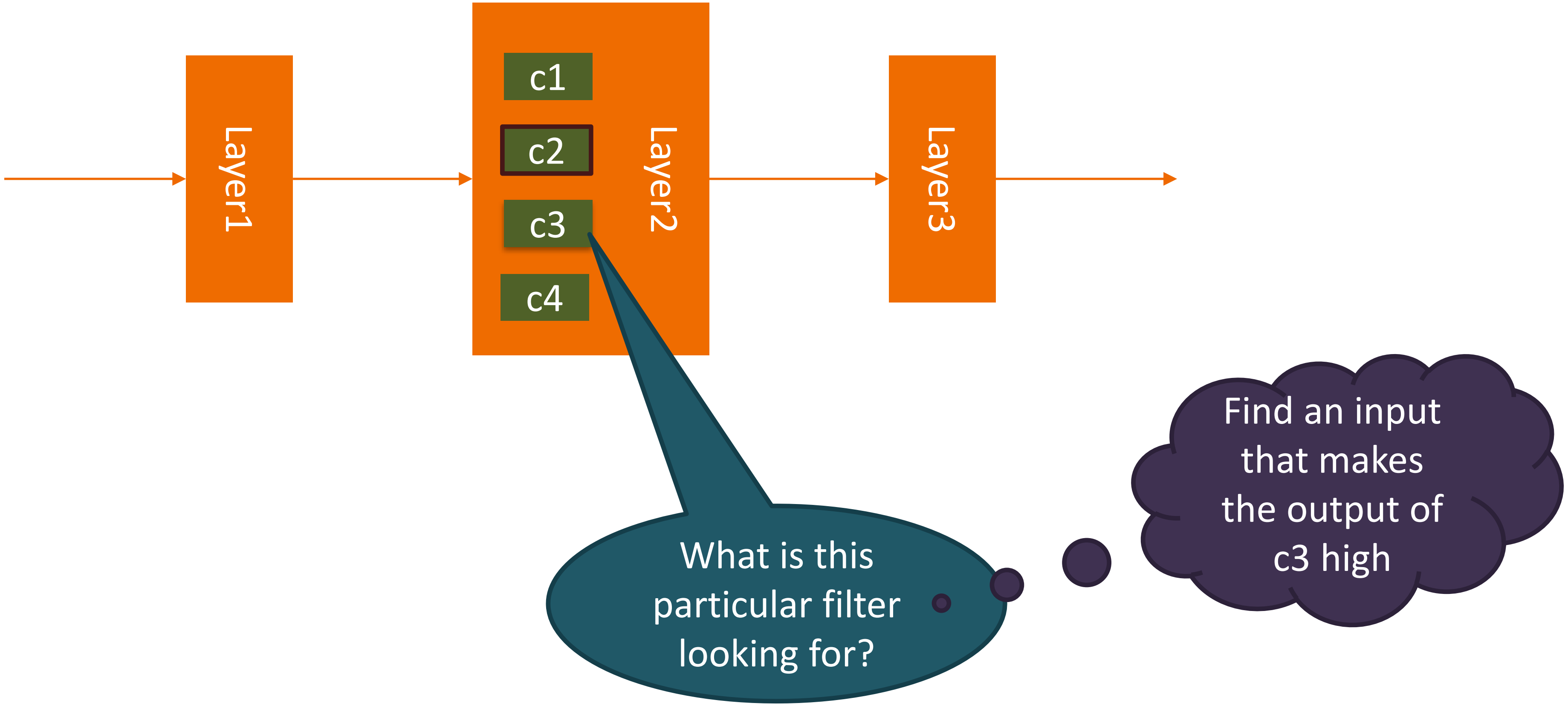
- Visualize filters of a conv net
- What patterns is a filter looking for in an input image?

Filters get more complicated towards the last layers

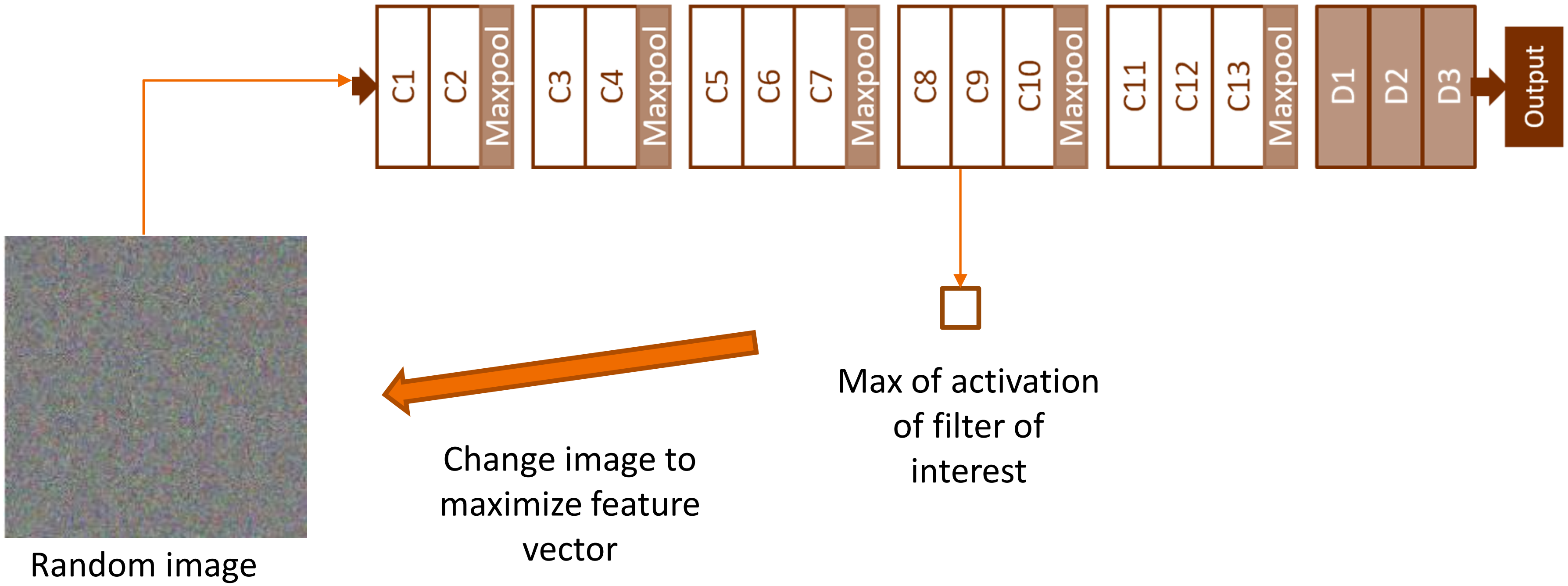


We need to find an image that maximizes the activation of a filter

Activation maximization

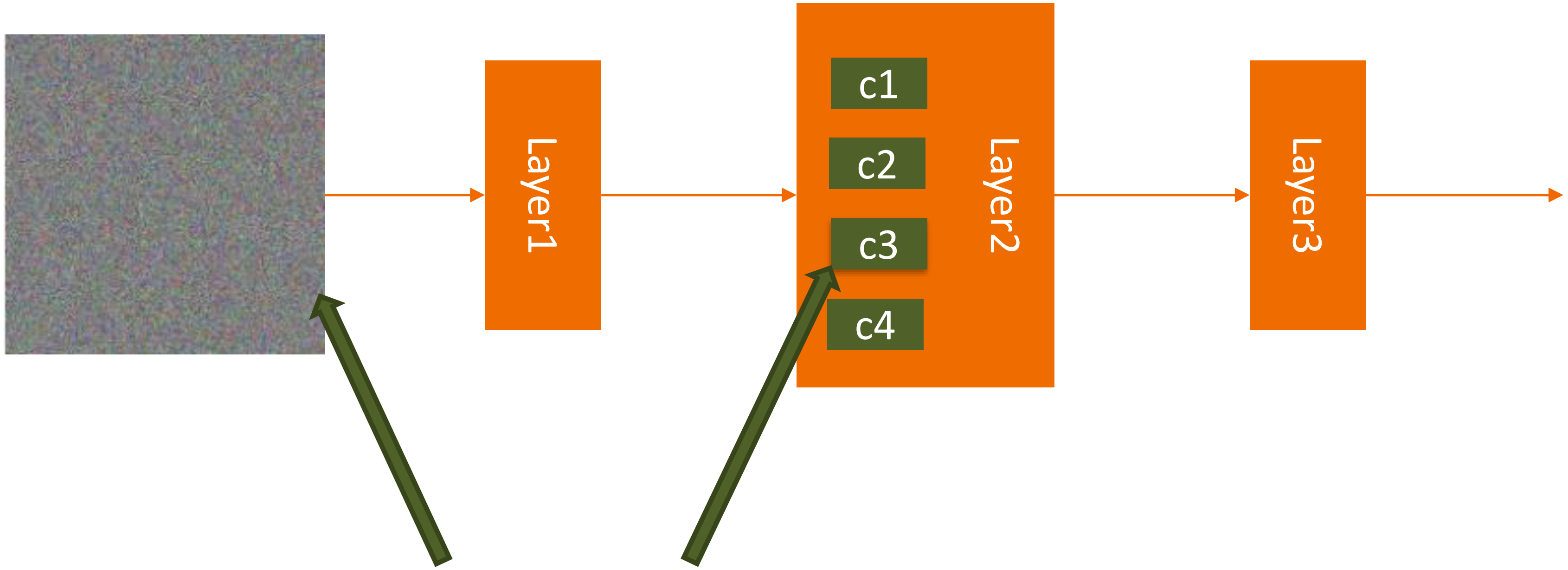


Activation maximization method



- $x^* = \underset{x}{arg\max}\, h(x) + \lambda R$
- where R is a regularization function

Activation maximization



Search through all images for one that makes the activation of c3 high

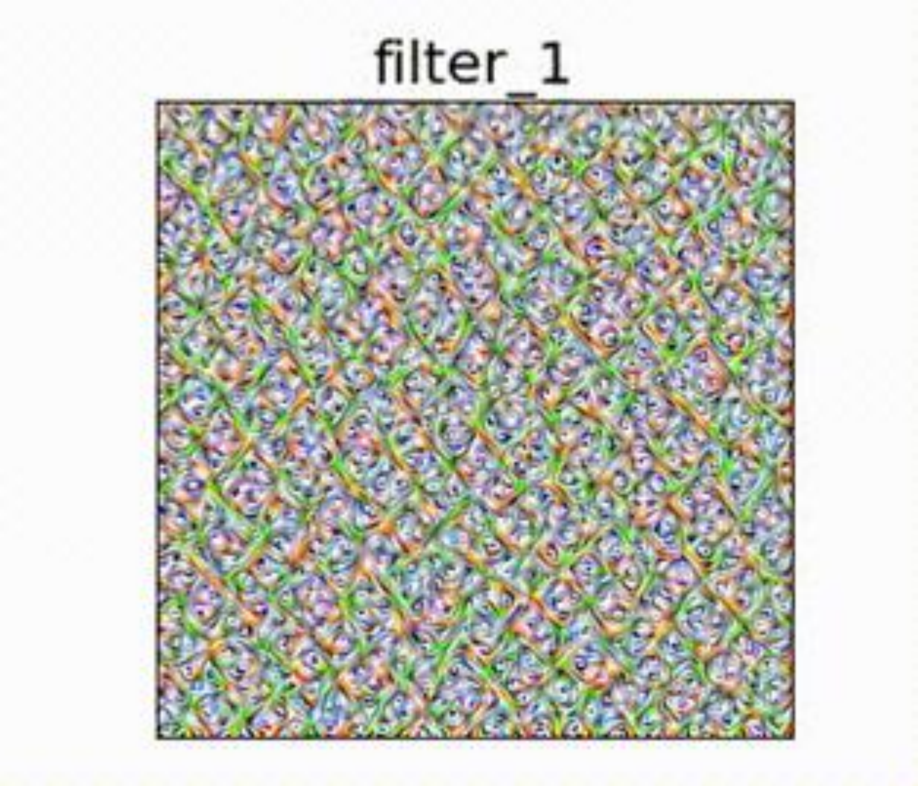
Activation = c3(Layer1(x))

We can use gradient descent for this!

Code Available

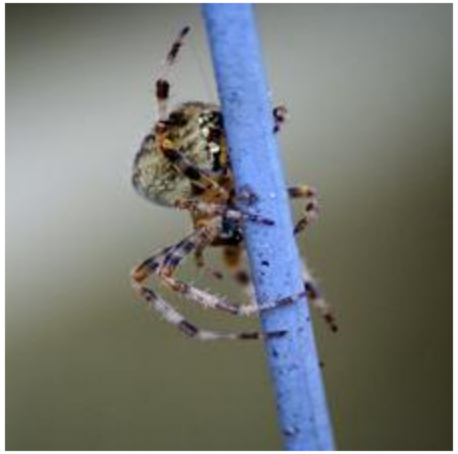
Convolutional Neural Network Filter Visualization

- Optimize the input image with respect to output of the specific convolution operation
- Used a pre-trained **VGG16**
- Visualizations of layers start with basic color and direction filters at lower levels
- Complexity of the filters also increase in the final layers

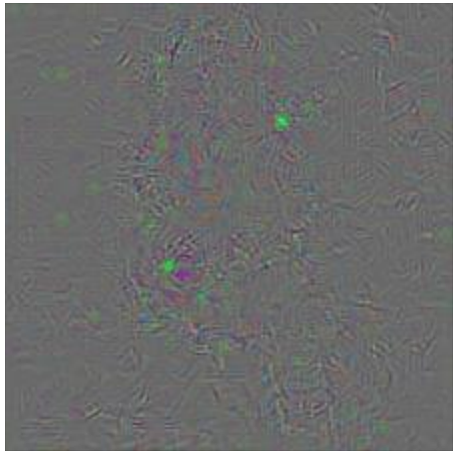


Code Available

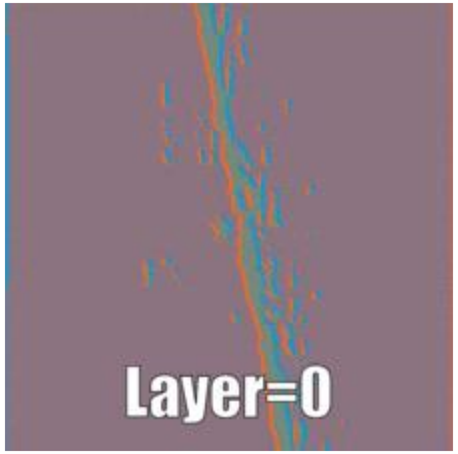
Understanding VGG net w.r.t an input



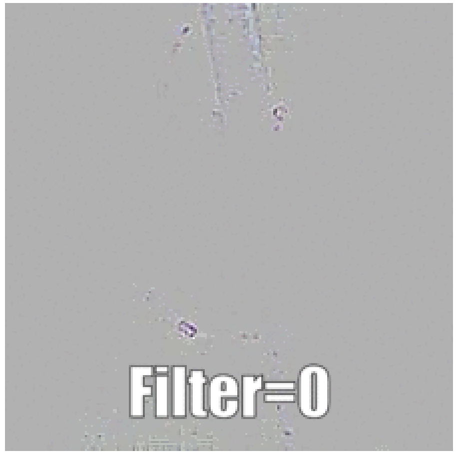
Input Image



Gradients generated with vanilla back propagation from Input



Visualize activations of Filter 0 in the first 30 layers of VGG16 net for given input



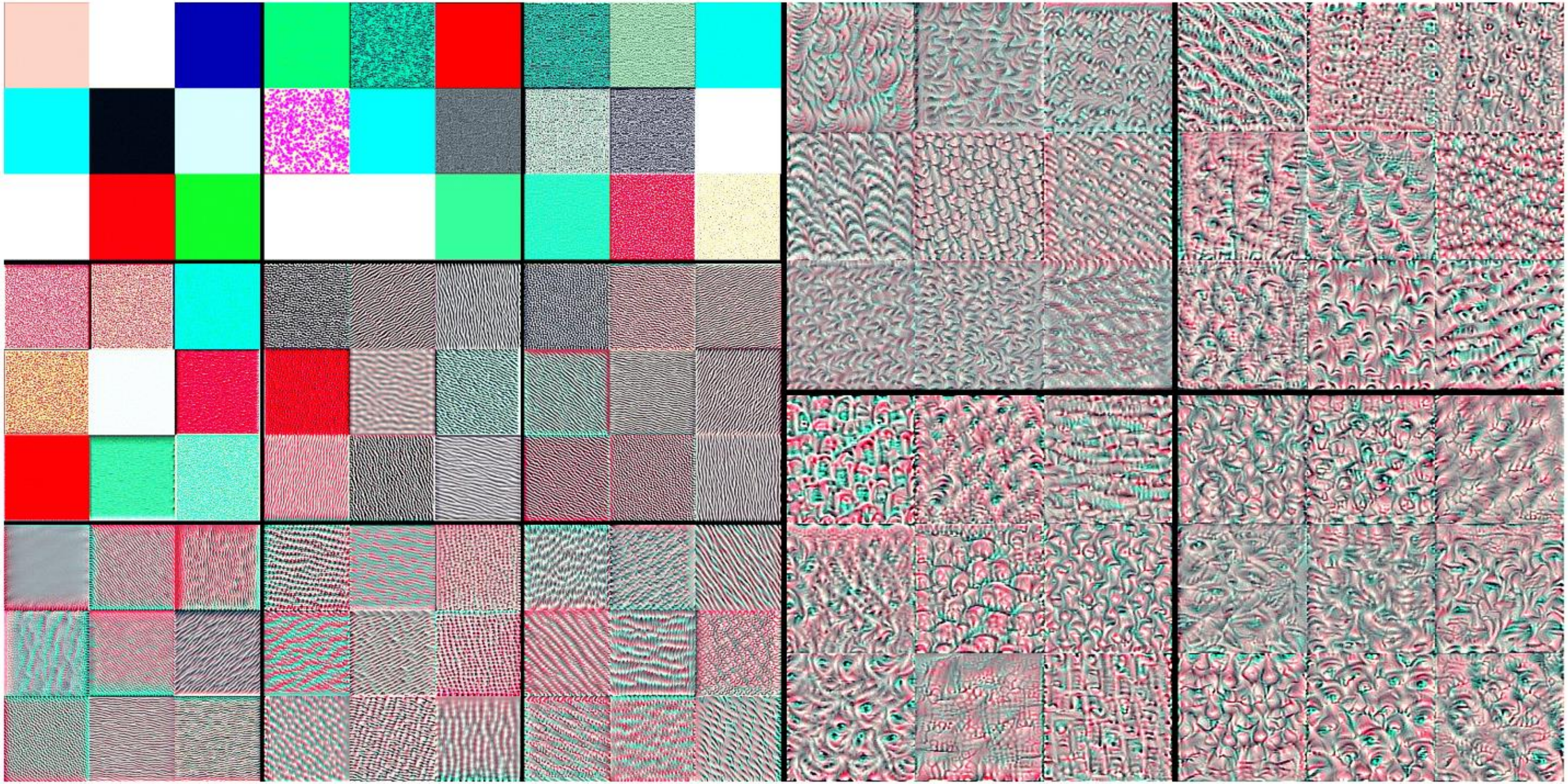
Visualize activations of the first 30 filters in layer 29 for given input

Filter hierarchy of object recognition network

Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

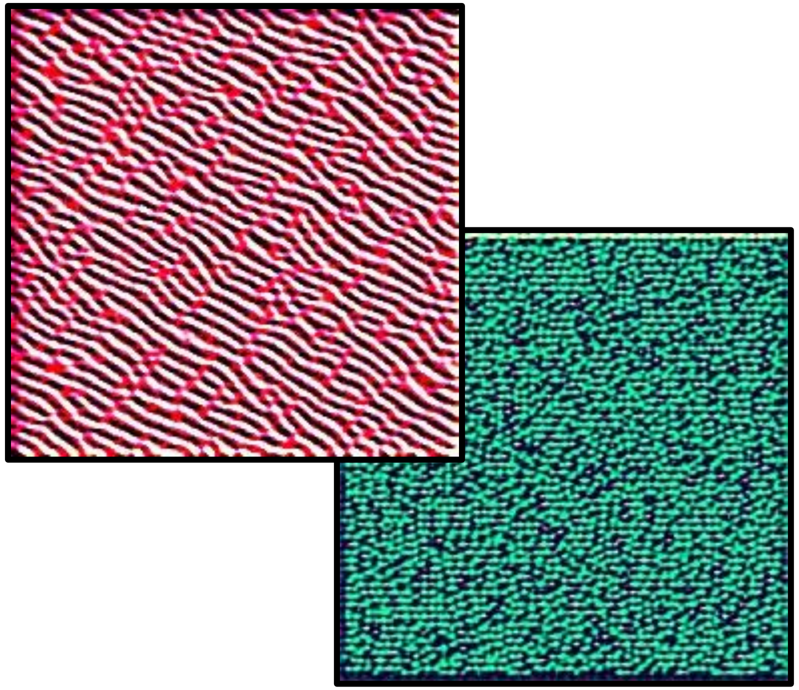
Image source: [Feature Visualization (distill.pub)]
Feature visualization of GoogleNet layers

Filter hierarchy of face networks

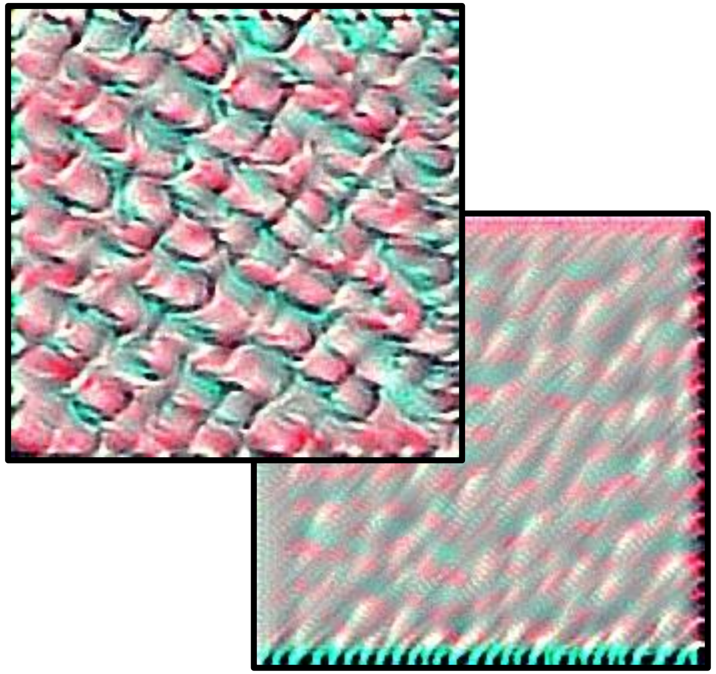


Feature visualization of 9 filters from each conv layer of VGG-Face

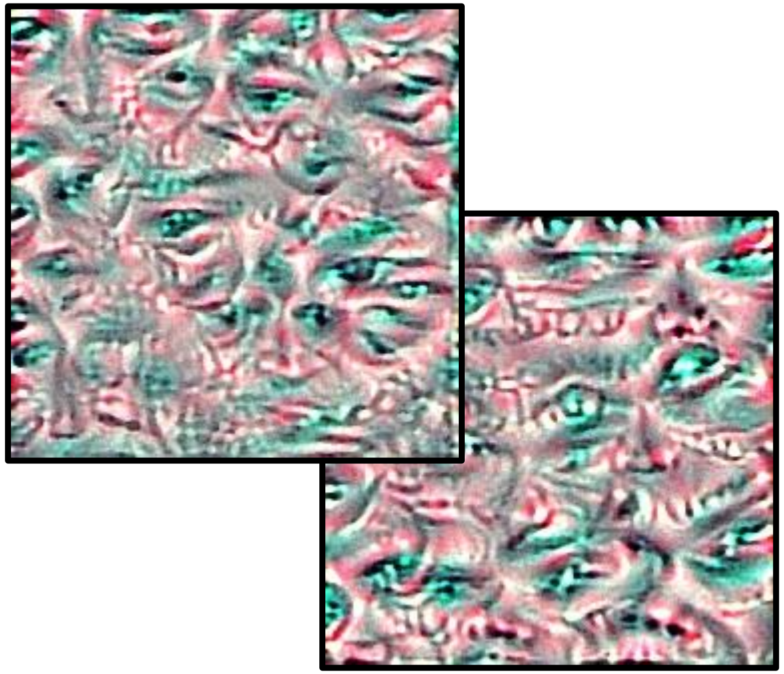
Filter hierarchy for face networks



Low-level features



Mid-level features



High-level features

Acquire Raw Data

Prepare / Clean Data, Visualization

Feature Engineering

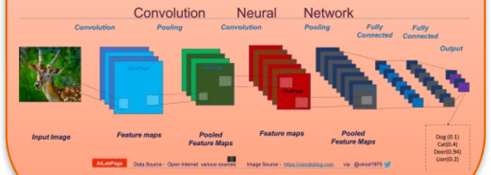
Pick Model and Hyper-params for Task

Model Training / Optimization

Evaluate Model Performance

Deploy Model

- Appreciating CNNs

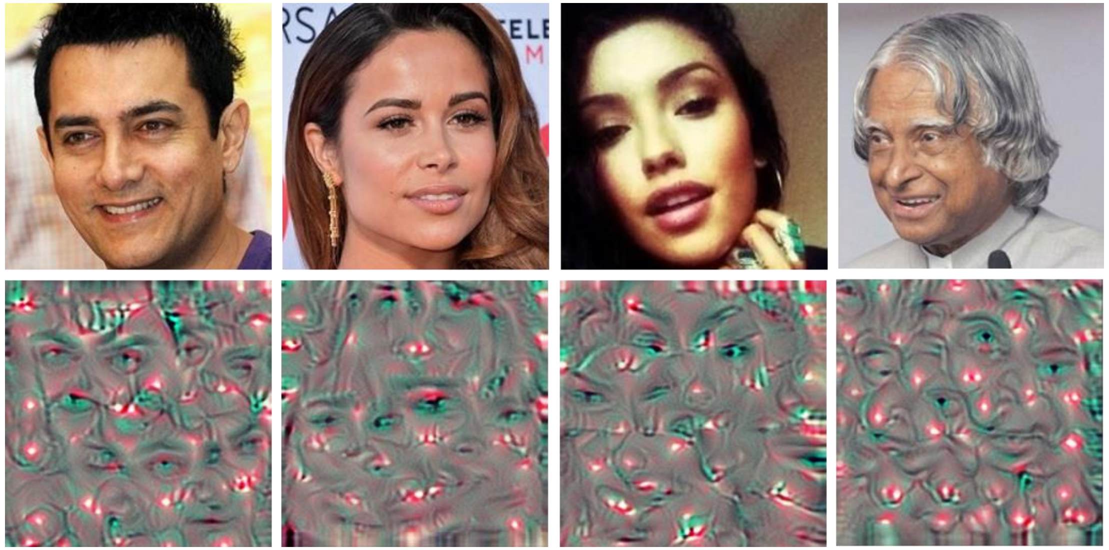


Insight into CNNs

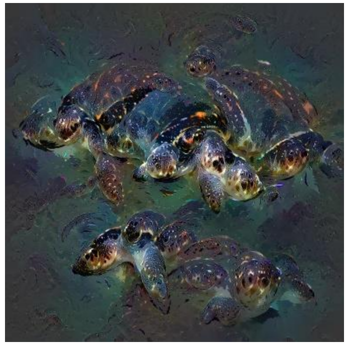
What do they learn?

Class visualization

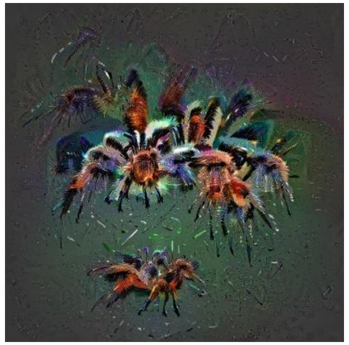
- Visualize the output neurons instead of a specific conv filter



Maximize the **output** of a class neuron



Turtle



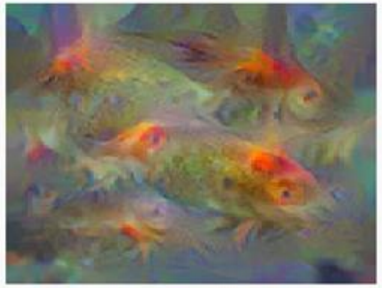
Tarantula

Pelican

Chains

Images on the left: Thrupthi Ann John
Images on the right: [Class visualization with bilateral filters]

Better Visualizations


(a) Goldfish


(b) Indigo Bunting

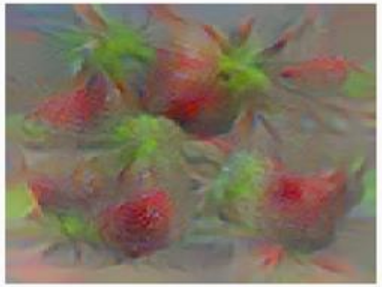

(c) Magpie


(d) Kite


(e) Goose

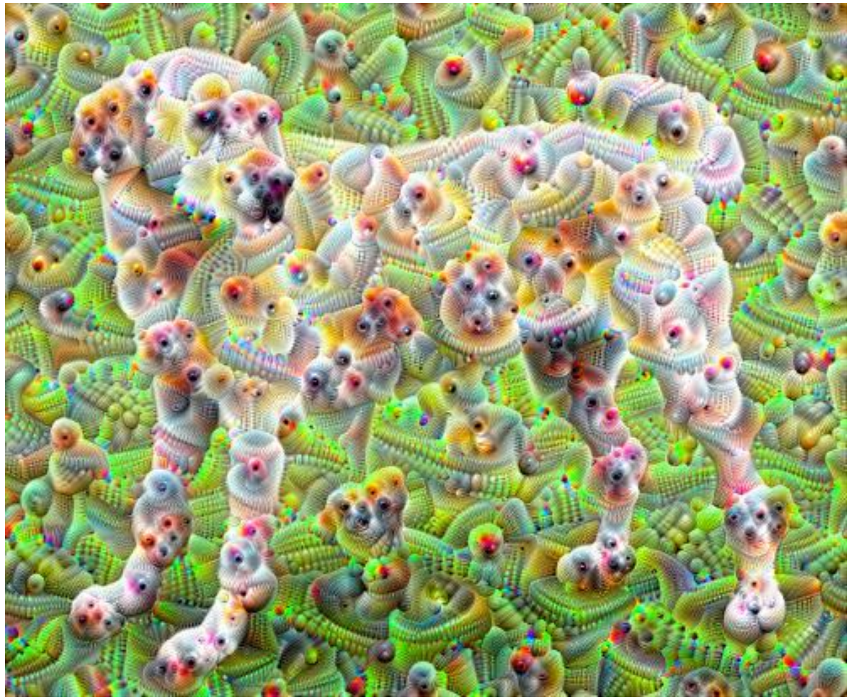

(f) Flamingo

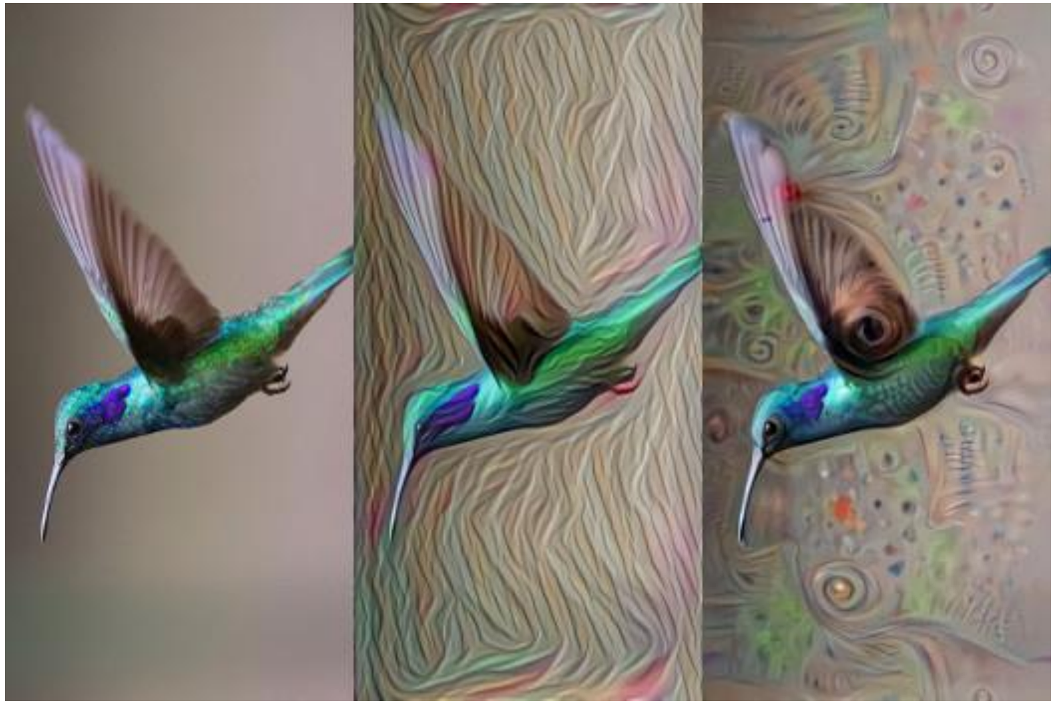

(g) Japanese Spaniel


(h) Strawberry

Deep dream: AI Creativity?

- How does a convnet interpret an image?

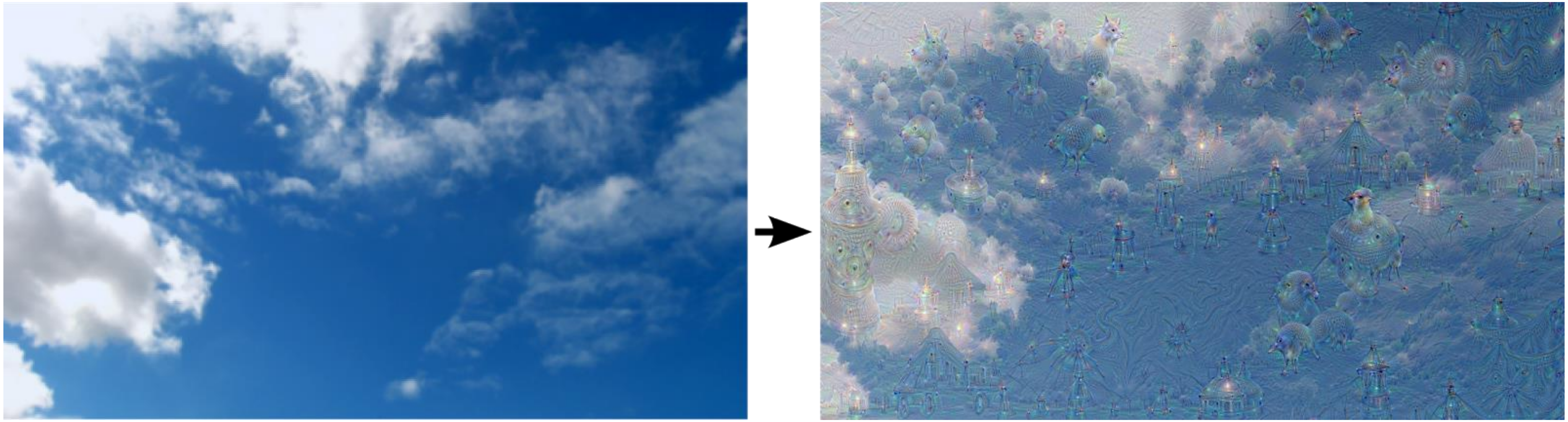




https://www.tensorflow.org/tutorials/generative/deepdream

Interpreting a cloudy sky

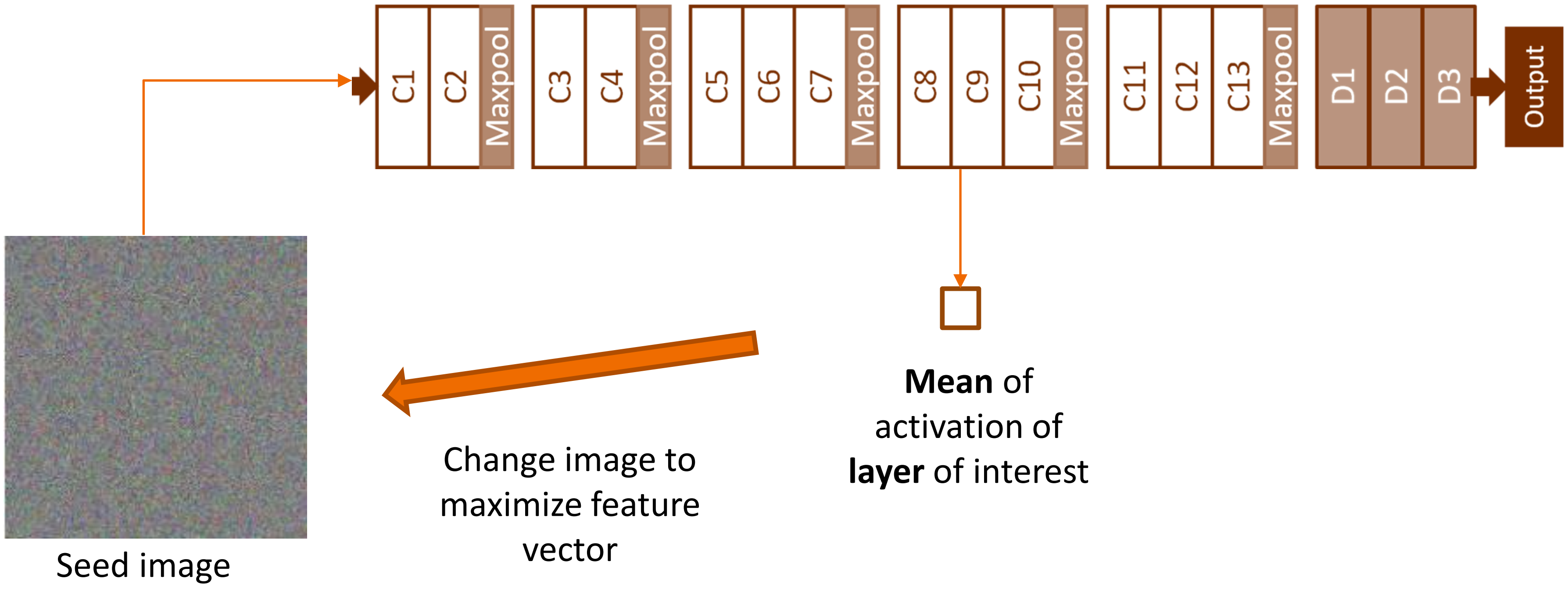
http://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html



Procedure

- Start with a natural image instead of a blank image
- Pick a layer we want to visualize
- Instead of choosing a filter, the loss function is the mean of the entire layer (This makes the model ‘choose’ what it sees)

Deep Dream method



- $x^* = \underset{x}{argmax}\, h(x) + \lambda R$
- where R is a regularization function

Deep dream: AI Creativity?

https://deepdreamgenerator.com/; https://en.wikipedia.org/wiki/DeepDream

Feature Inversion

- Invert a feature representation back to image space
- What information is retained or discarded down the layers?

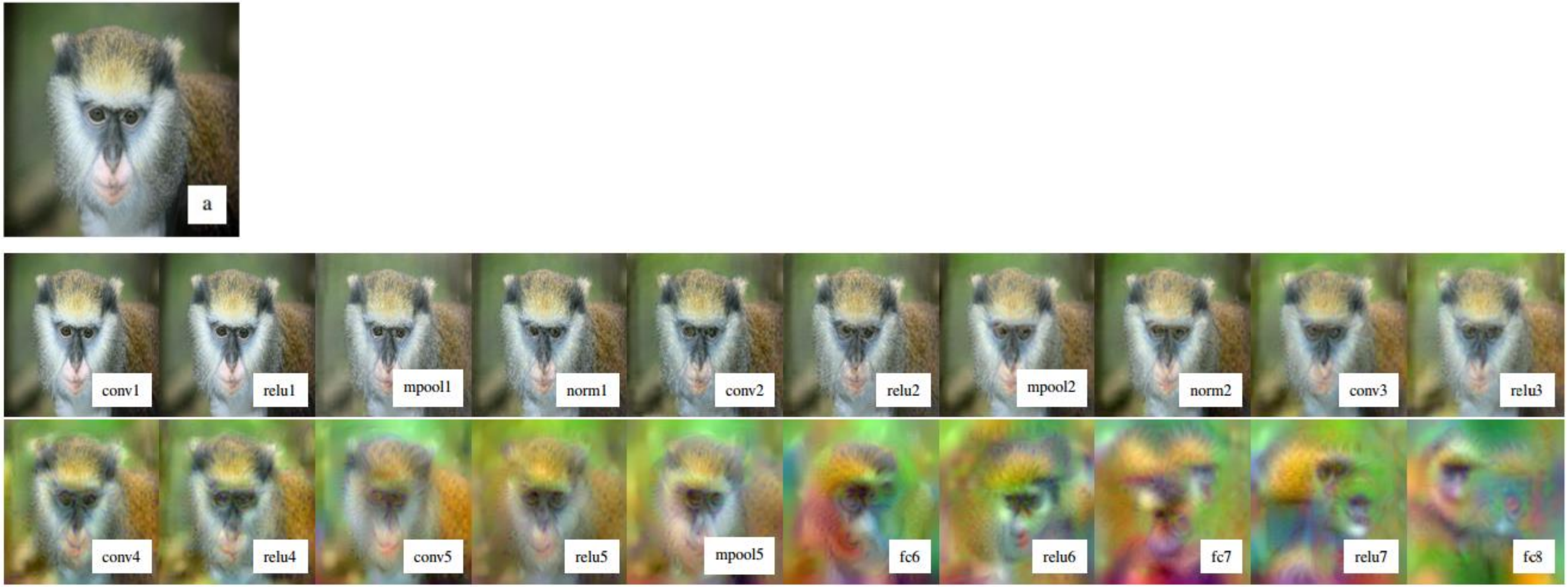
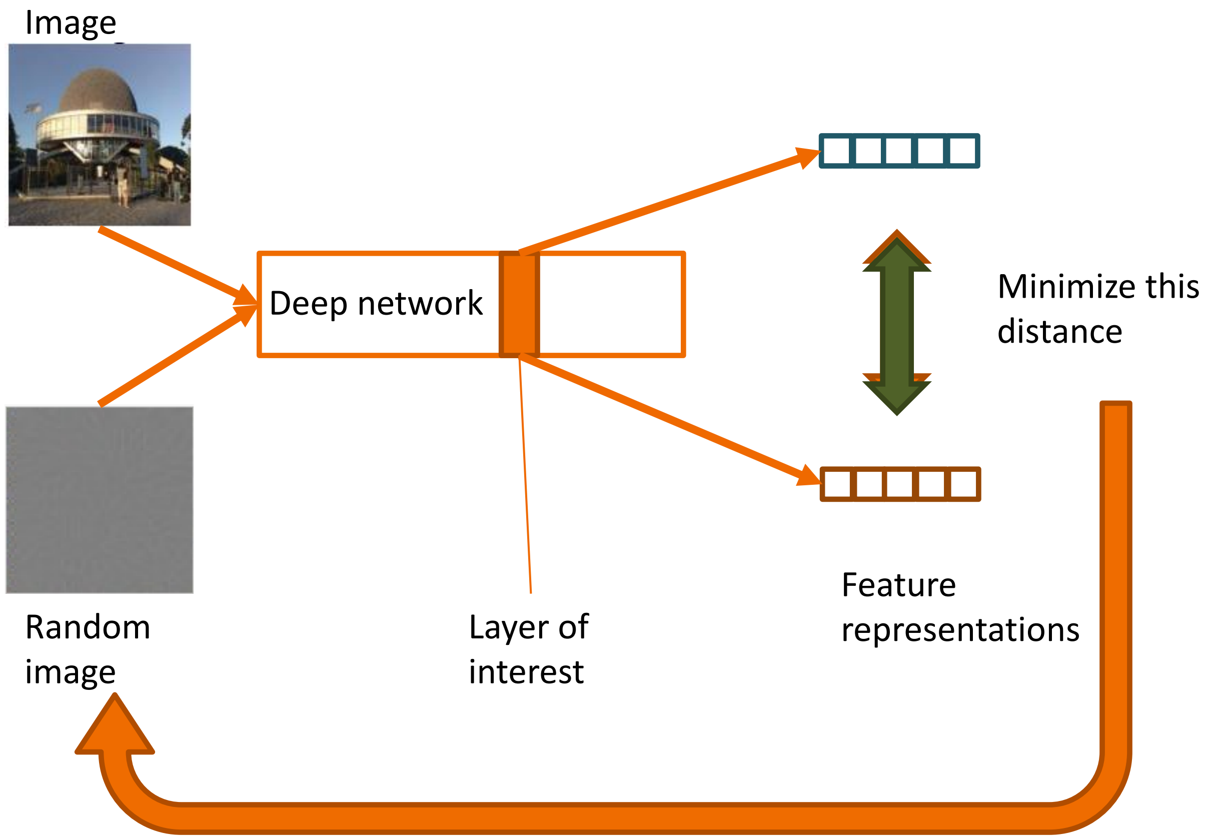


Image source: Mahendran, Aravindh, and Andrea Vedaldi. "Understanding deep image representations by inverting them." *CVPR* 2015.

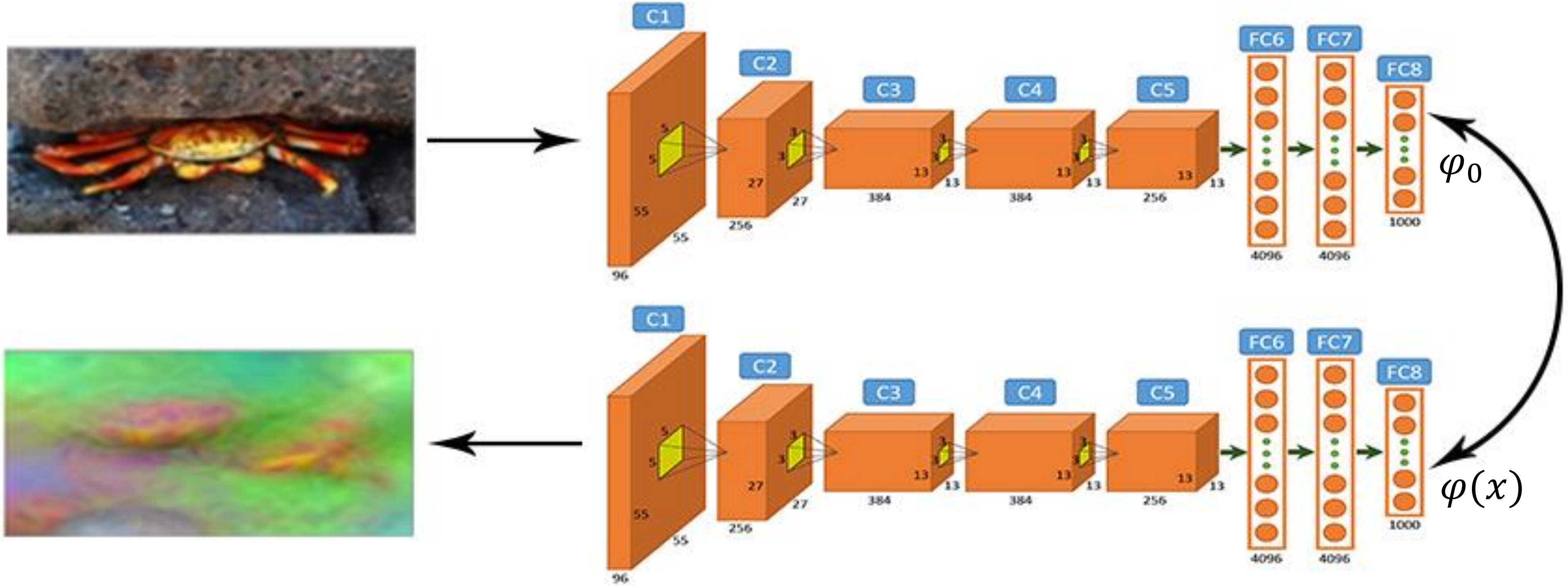
Method

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

Where $\ell()$ is a distance function between two feature representations and $\mathcal{R}()$ is a regularization function.



Inverting Specific Representation



$$x^* = \underset{x \epsilon R\ H \times W \times C}{argmin}\, L(\varphi(x), \varphi_0) + \lambda R(x)$$

Aravindh Mahendran and Andrea Vedaldi, Understanding Deep Image Representations by Inverting Them, CVPR’15

Inverting at different stages

- As we go down the layers, unnecessary information is discarded, and only discriminatory information remains

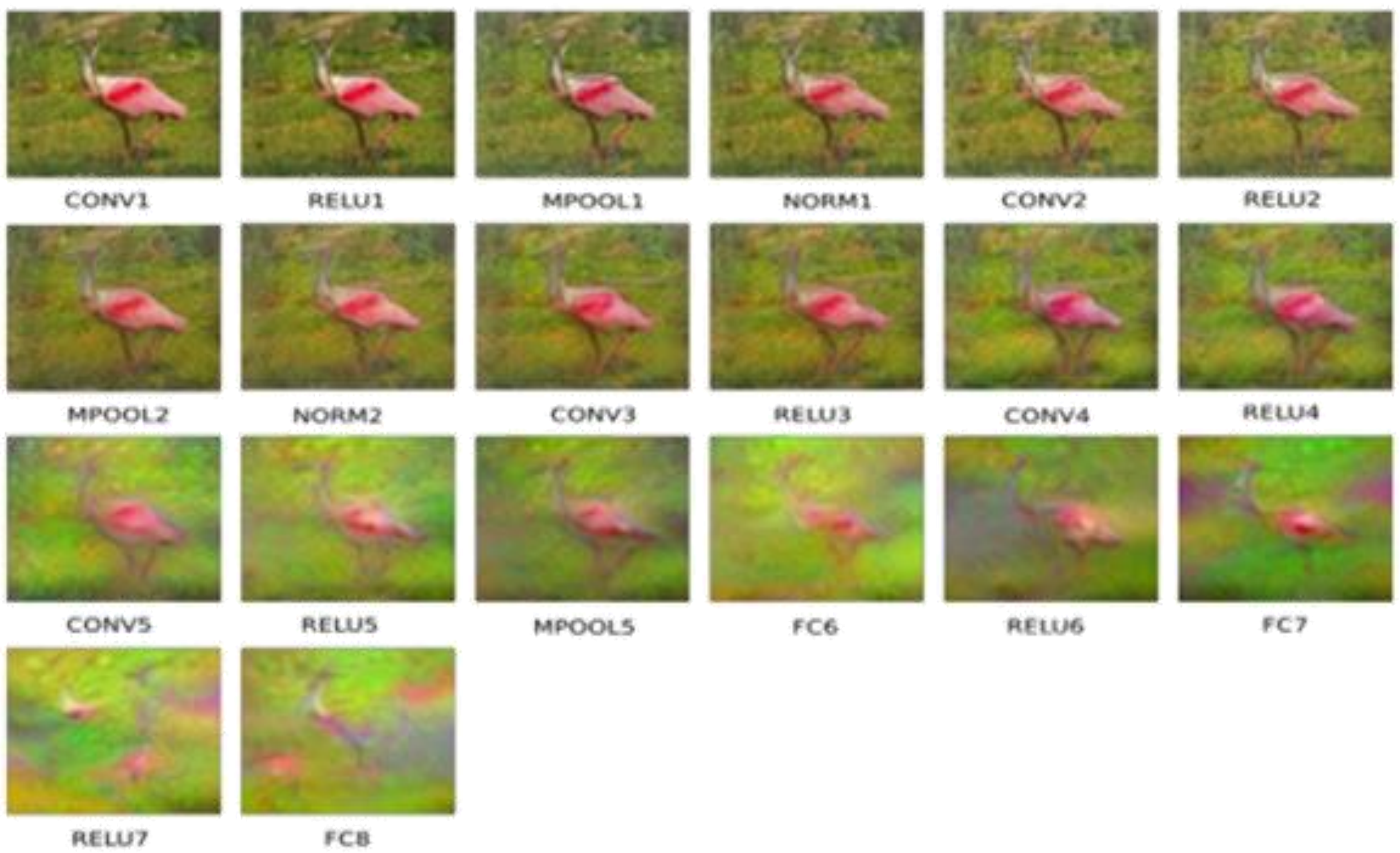


Image source: Mahendran, Aravindh, and Andrea Vedaldi. "Understanding deep image representations by inverting them." *CVPR* 2015.

Inverting faces

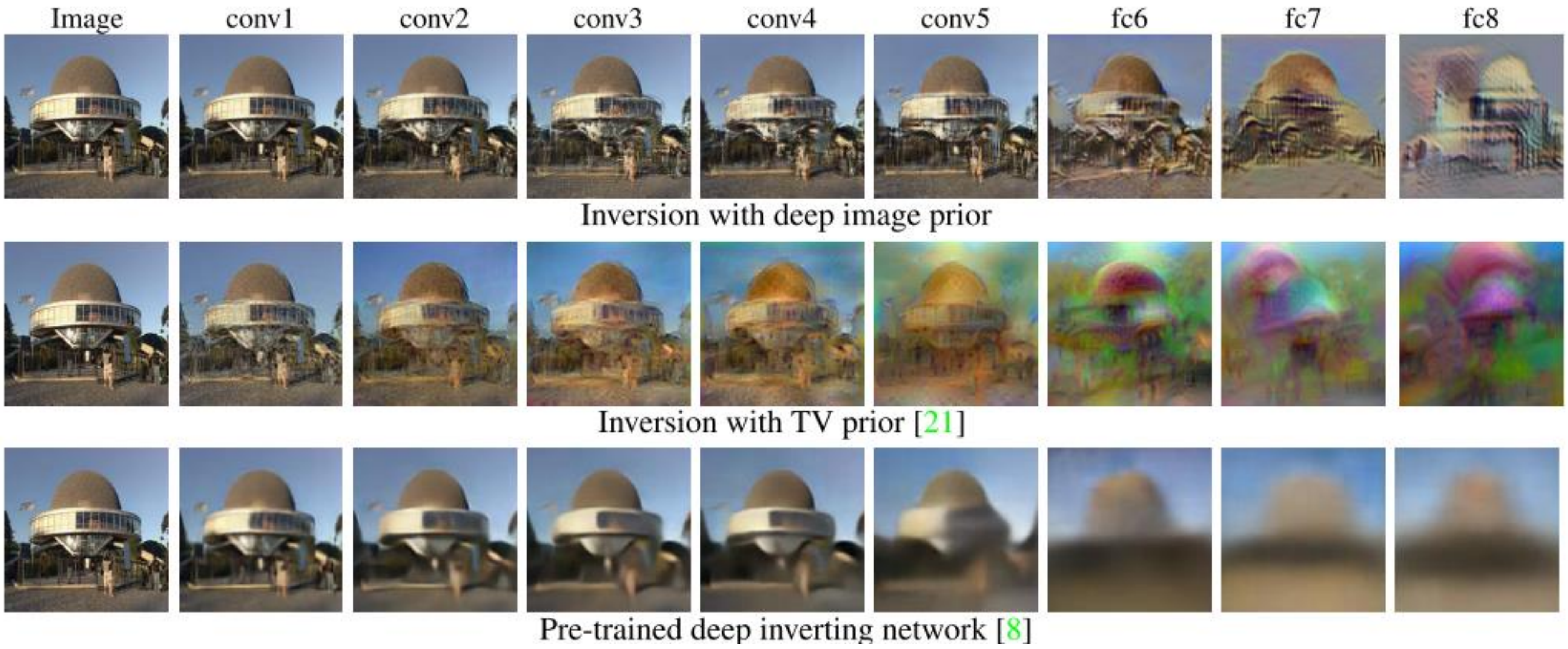
- Convnets trained for different tasks find different kinds of information useful to keep

Recognition Head Pose Emotion

Image: inversion of the last 3 convolutional layers of VGG-Face, VGG-Face pretrained for head pose and emotion

Regularization matters!

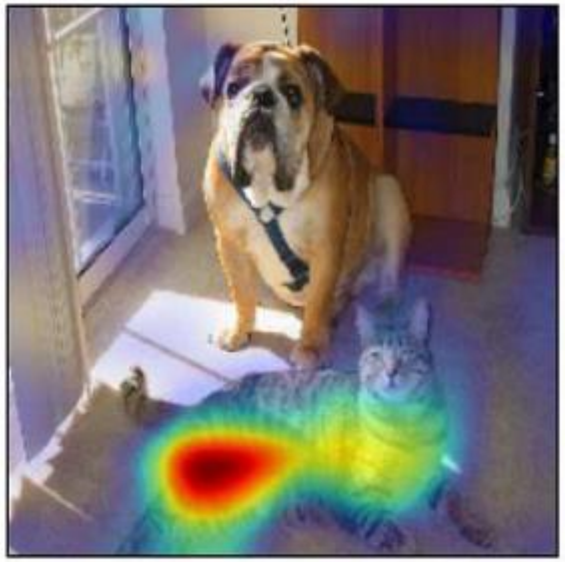
- The visualizations are very sensitive to regularization



Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." *CVPR*. 2018.

Saliency Visualization

- Which part of an input image is important?

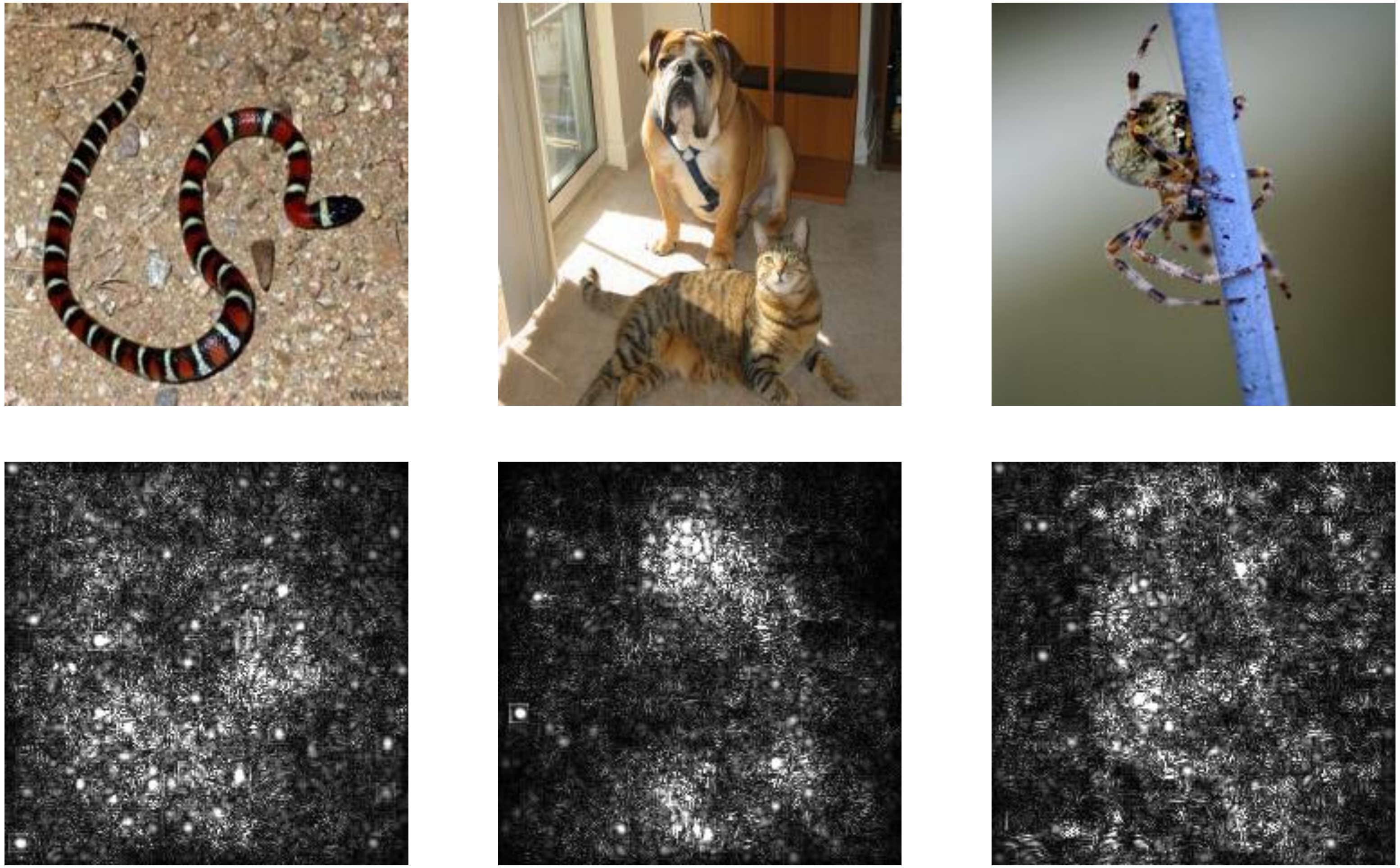


Class is ‘cat’

Class is ‘dog’

Image source: Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *ICCV* 2017.

Visualize magnitude of gradients



pytorch-cnn-visualizations/src at master · utkuozbulak/pytorch-cnn-visualizations (github.com)

Class activation mapping (Grad-CAM)

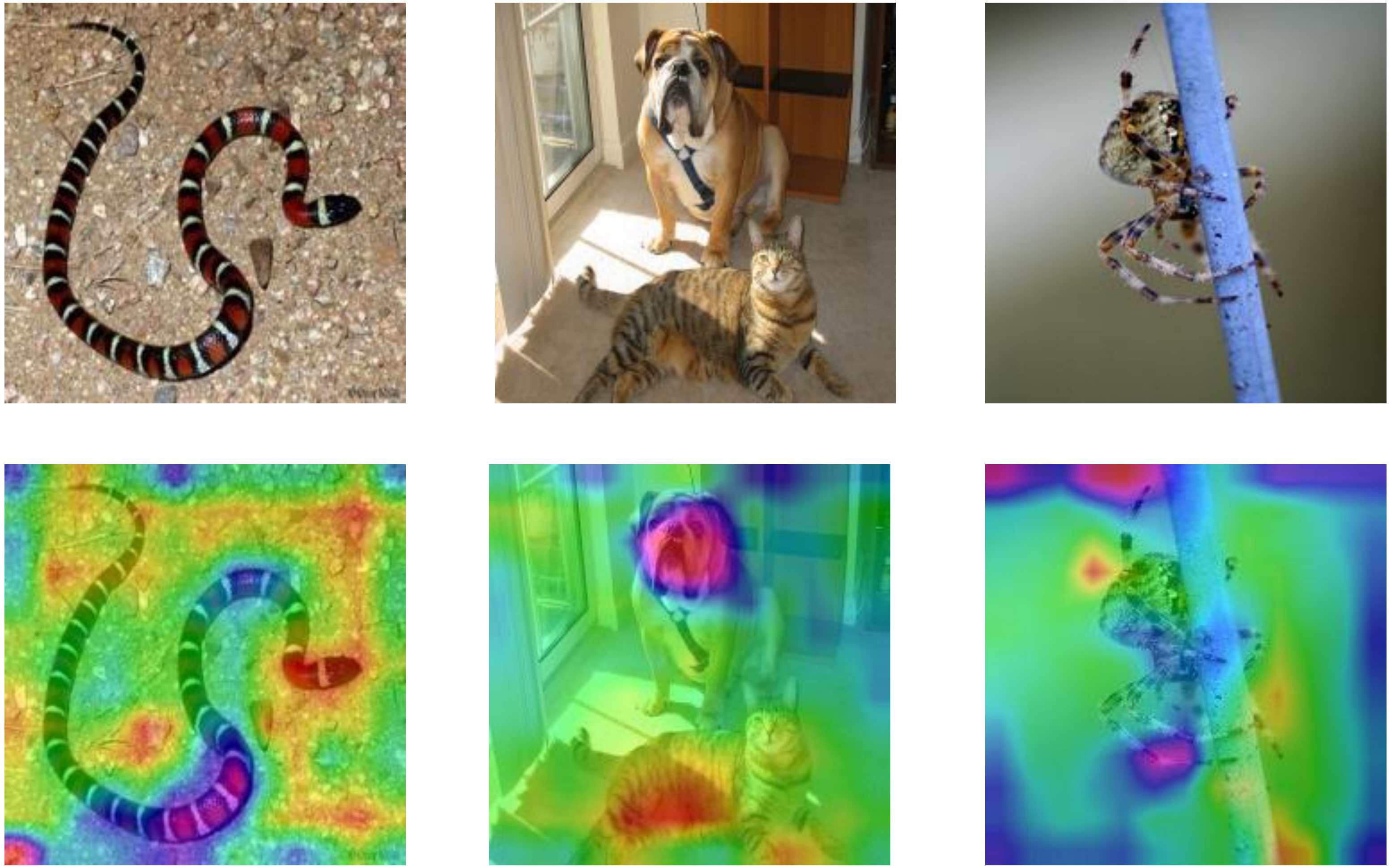
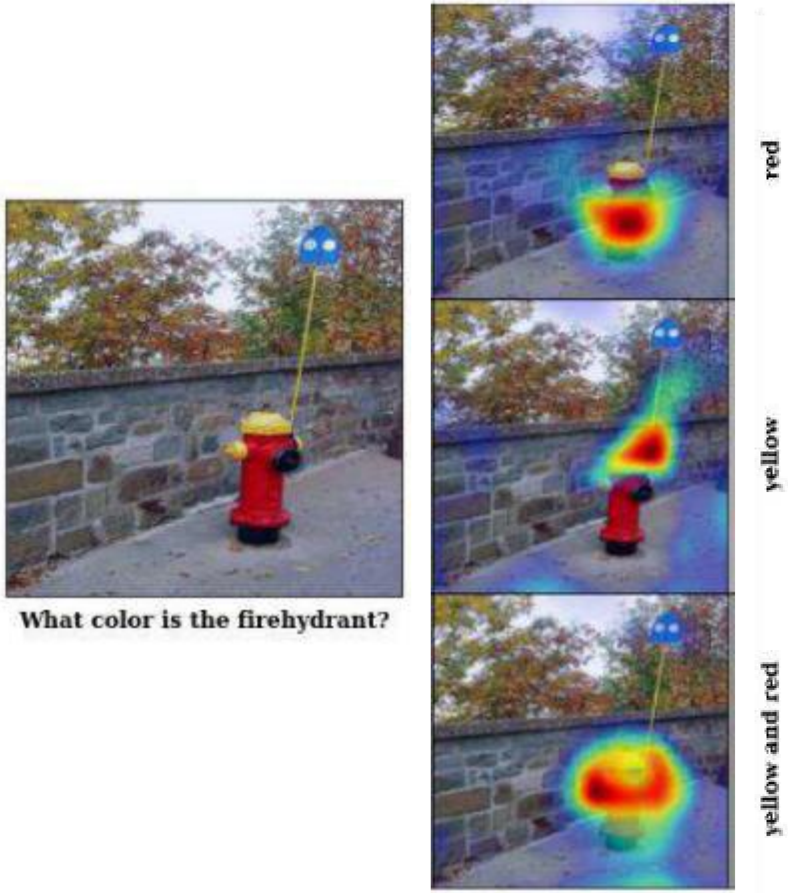


Image source: utkuozbulak/pytorch-cnn-visualizations: Pytorch implementation of convolutional neural network visualization techniques (github.com)

Grad-CAM: How

- Two basic Ideas (partly known earlier are combined)
- CAM: Class Activation Maps
  - Limitation to certain type of architectures
- Role of Gradients
  - e.g., Gradient of score of the class wrt to activation from a layer
  - We are interested in feature maps that have positive impact on the class of interest (negative ones correspond to BG or some other class)

GradCAM in Visual Question Answering



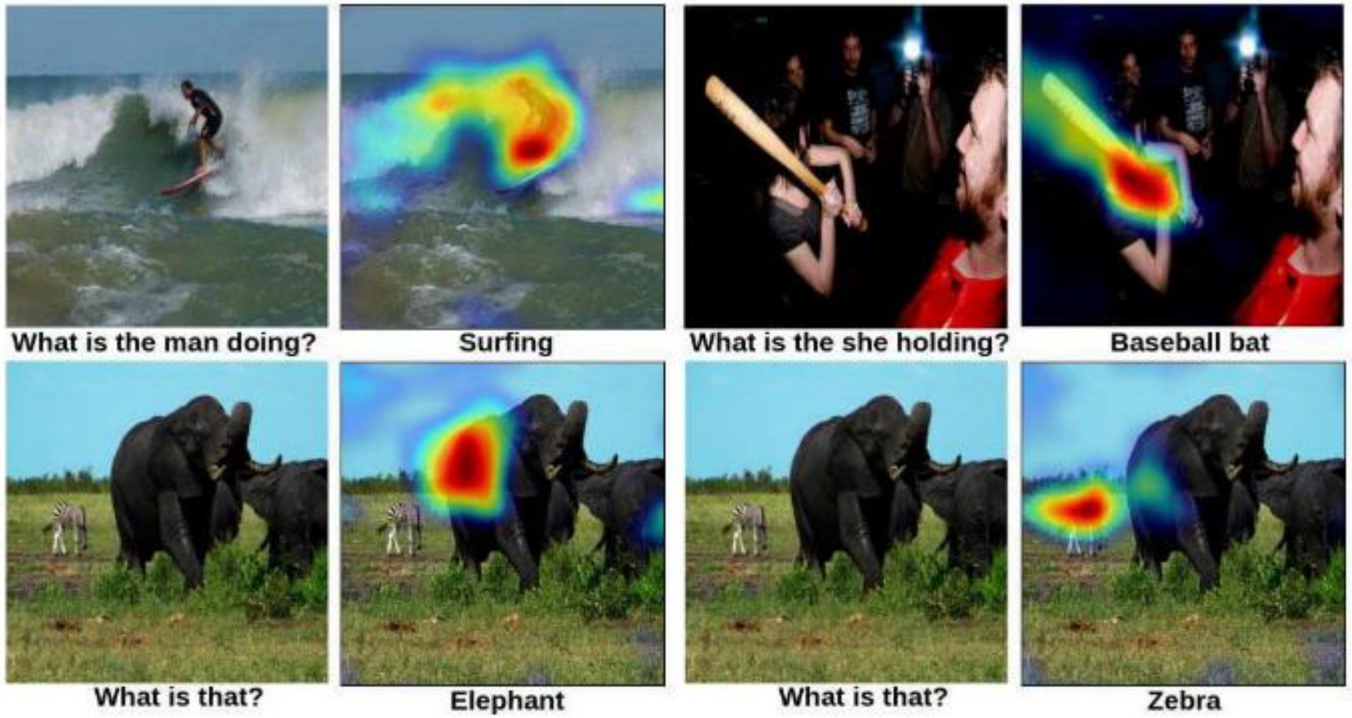


Image source: Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *ICCV* 2017.

Grad-CAM in image captioning

A group of people flying kites on a beach   A man is sitting at a table with a pizza

A house with a green roof   Sheep grazing in field   A house with a roof

Image source: Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *ICCV* 2017.

Another: Occlusion map

- Systematically occlude parts of the image and record the drop in classification confidence

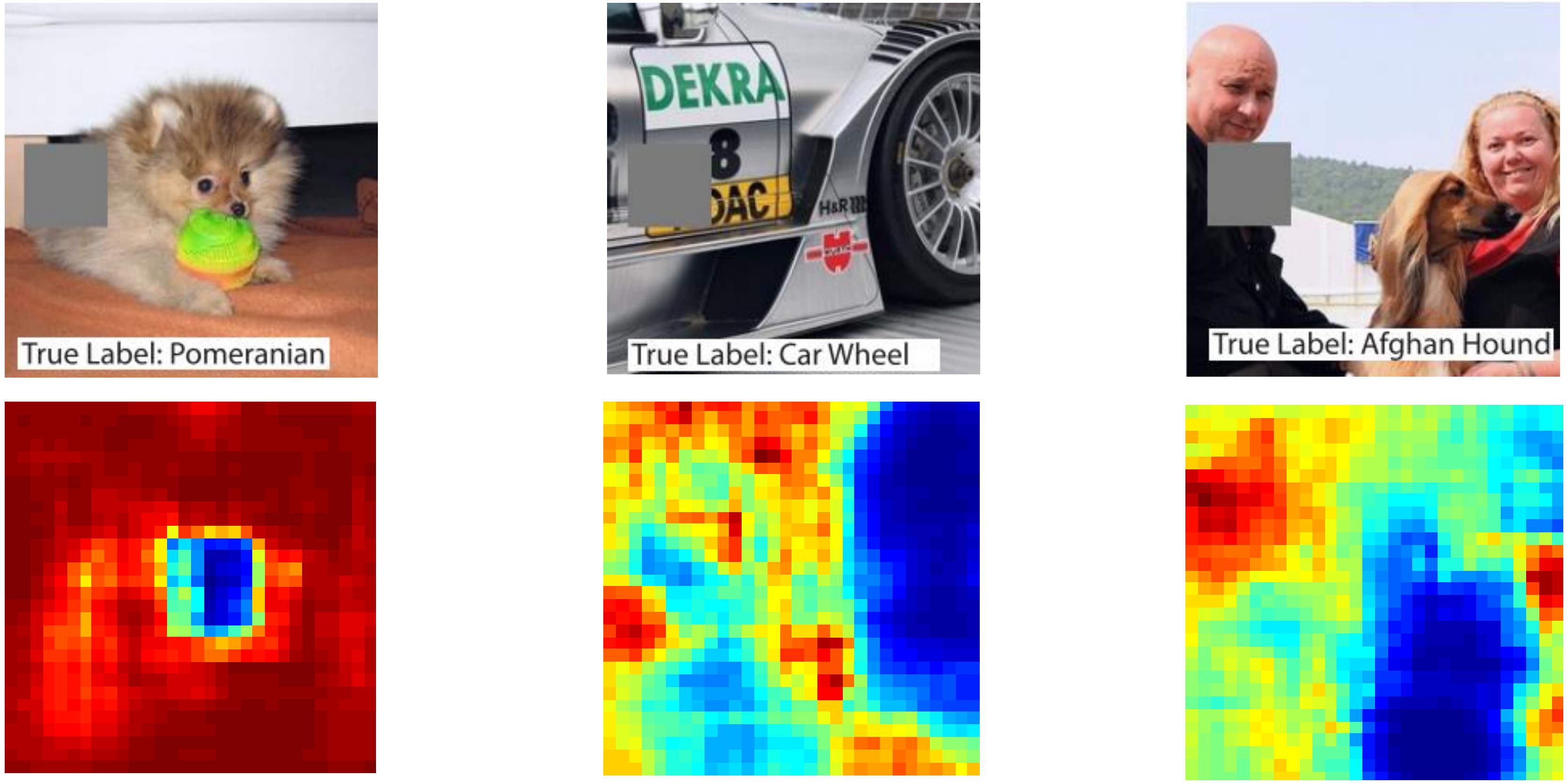
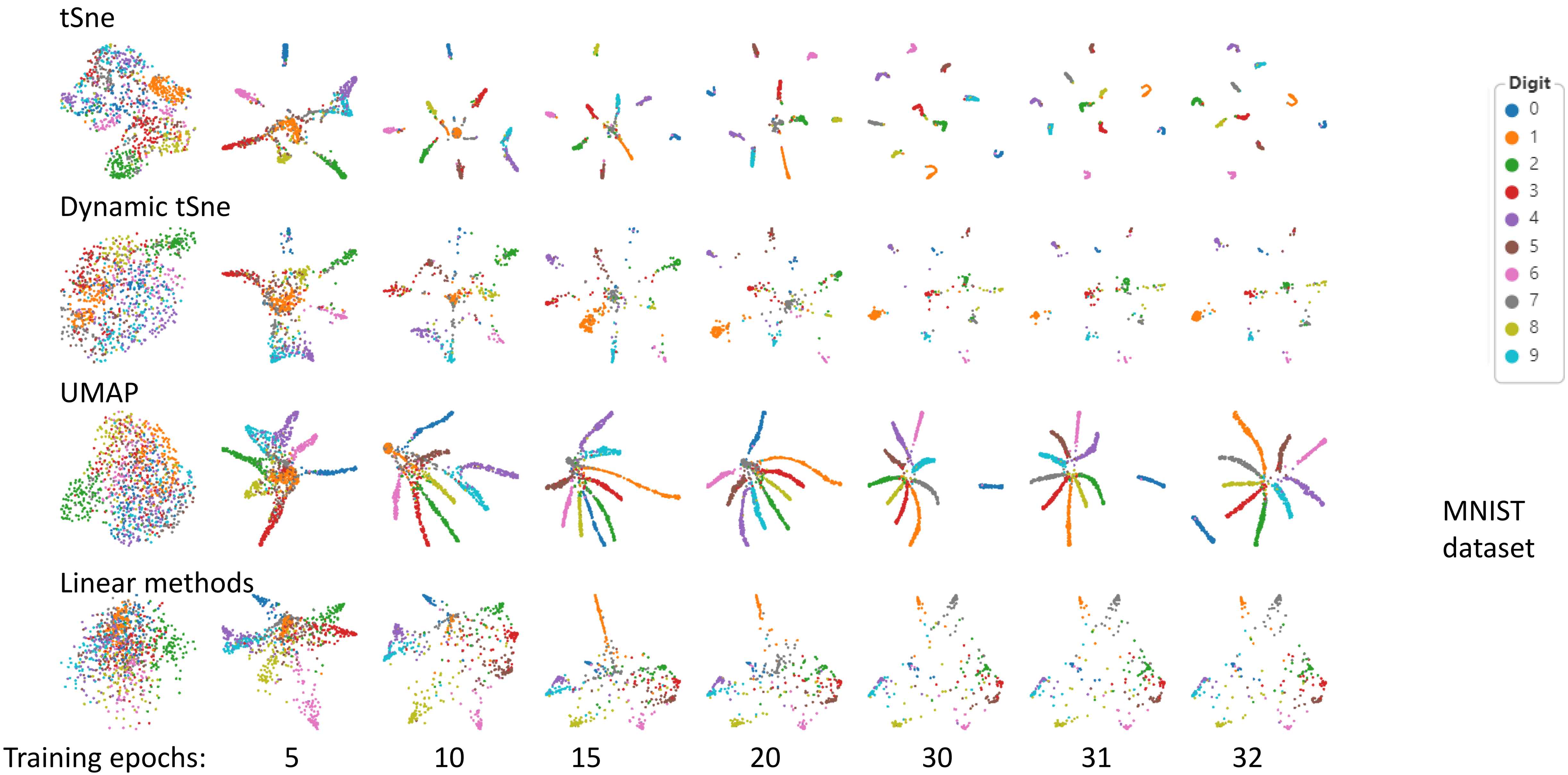


Image source:
Zeiler, Matthew D., and Rob Fergus. 2014. "Visualizing and Understanding Convolutional Networks. *ECCV 2014*,

At output: Monitor training dynamics 2D embedding



MNIST dataset

Summary

- Visualizing different aspects of CNNs
- Interpreting for better understanding
- Better explanations
- Better design/refinements

Thanks!!

Questions?