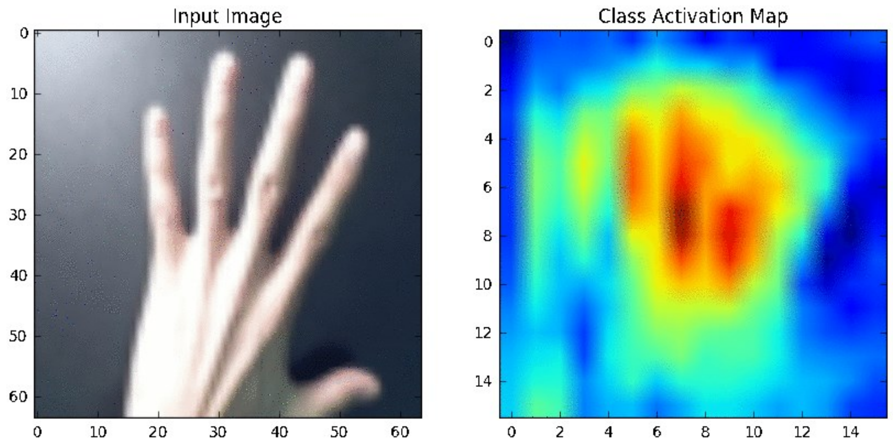
Visualization of CNN

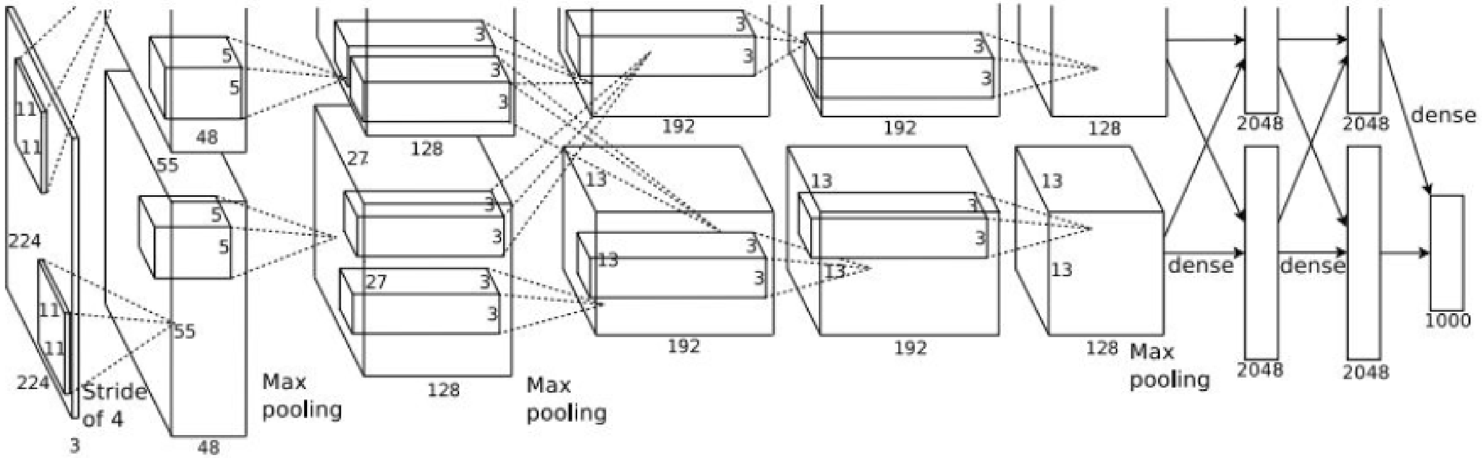


Fast Campus
Start Deep Learning with Tensorflow

What's going on inside CNN?

This image is CC0 public domain

Input Image:
3 x 224 x 224



Class Scores:
1000 numbers

What are the intermediate features looking for?

Slide Credit : Stanford CS231n

Visualize Patches that Maximally Activate Neurons

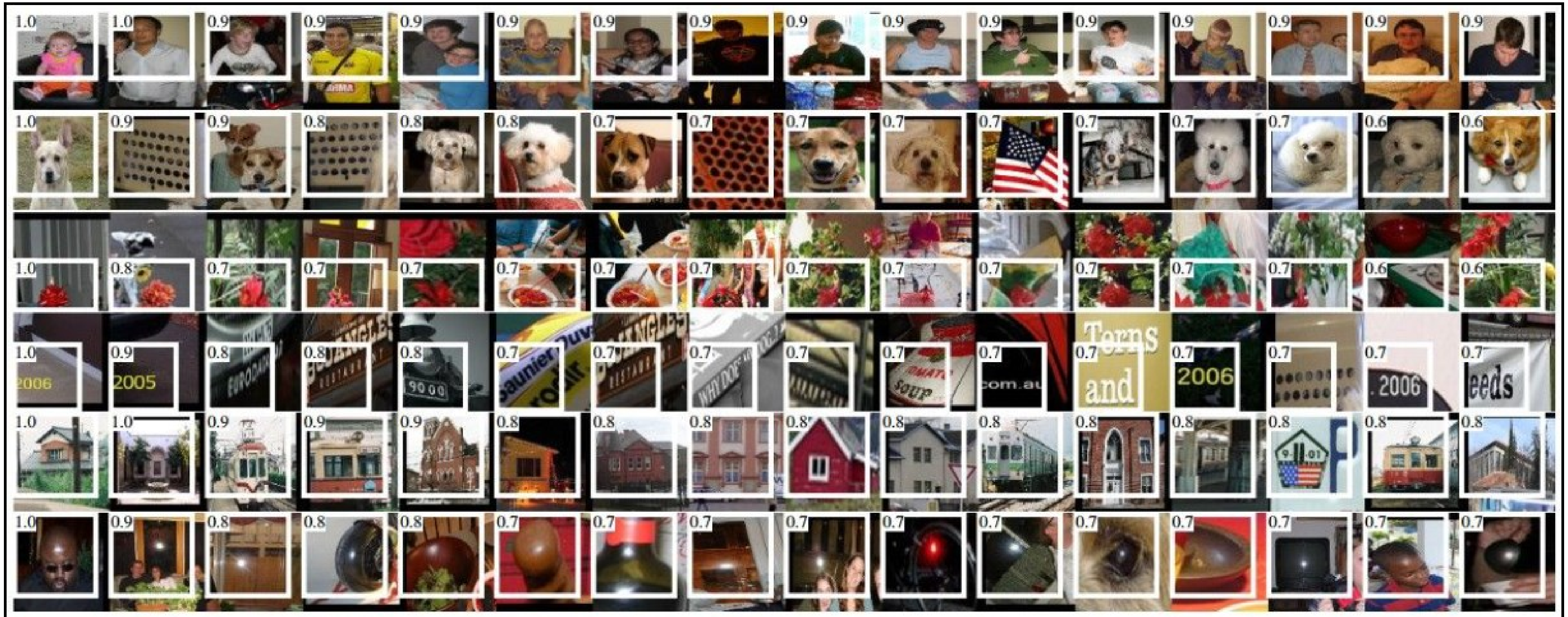
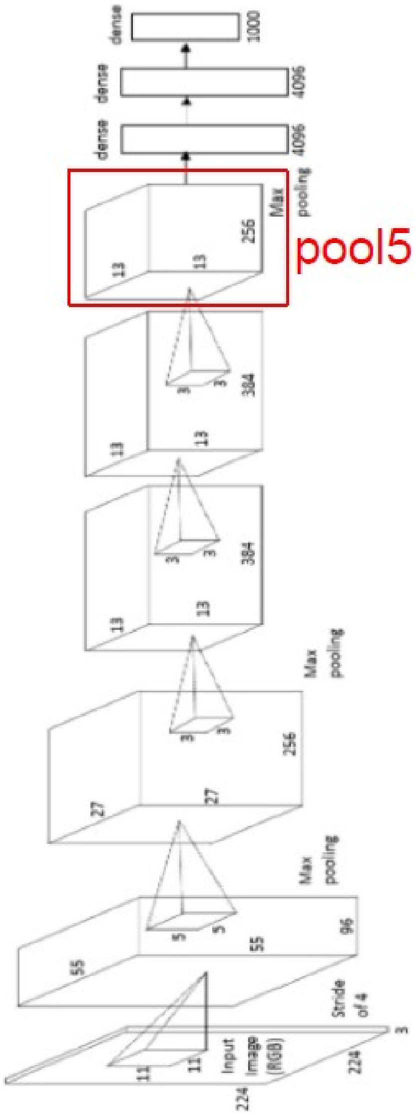


Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

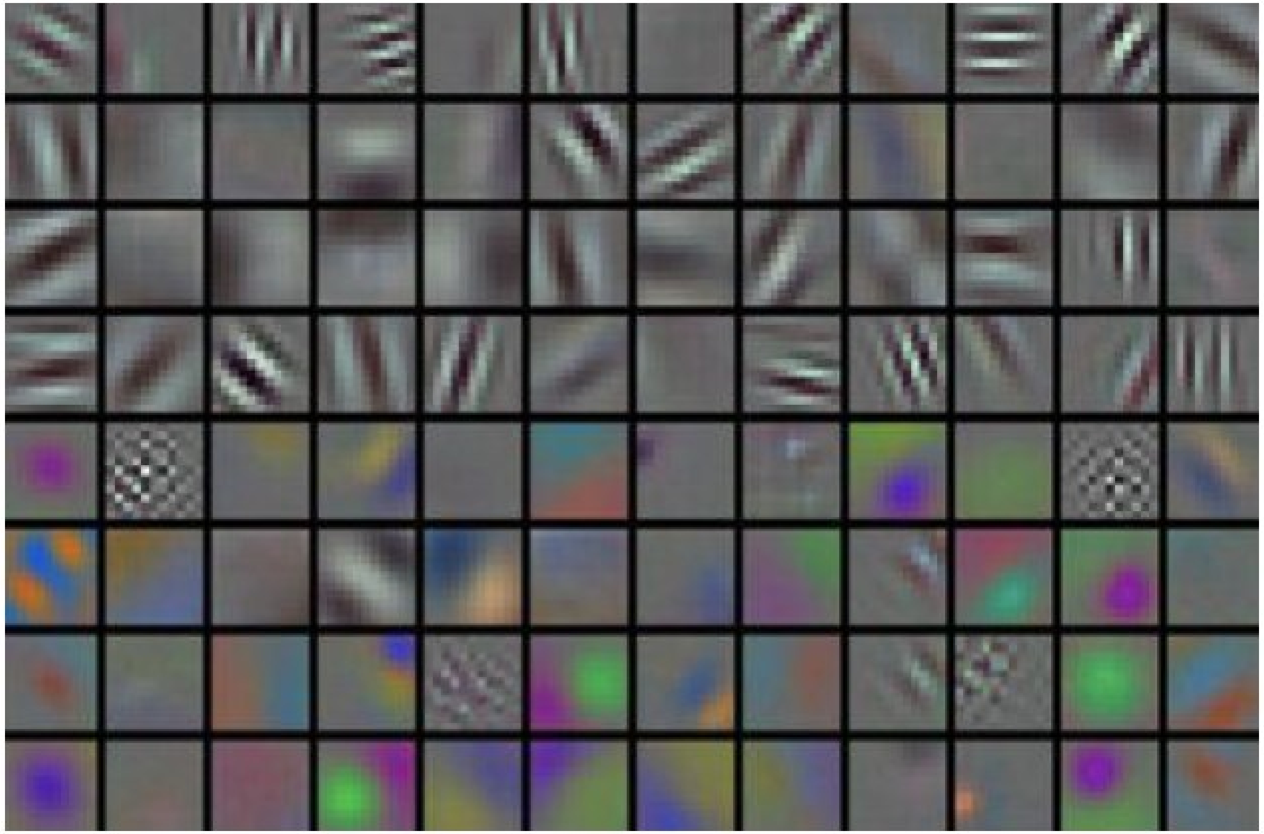


Rich feature hierarchies for accurate object detection and semantic segmentation [Girshick, Donahue, Darrell, Malik]

Slide Credit : Stanford CS231n

Visualize Filters

- Only interpretable on the first layer

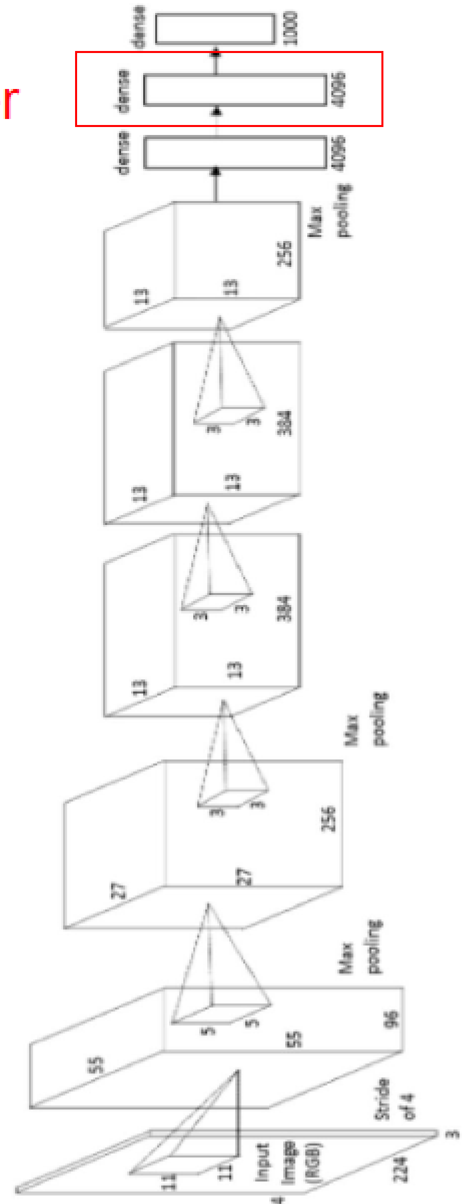


- http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

Slide Credit : Stanford CS231n

Visualizing the Representation

fc7 layer



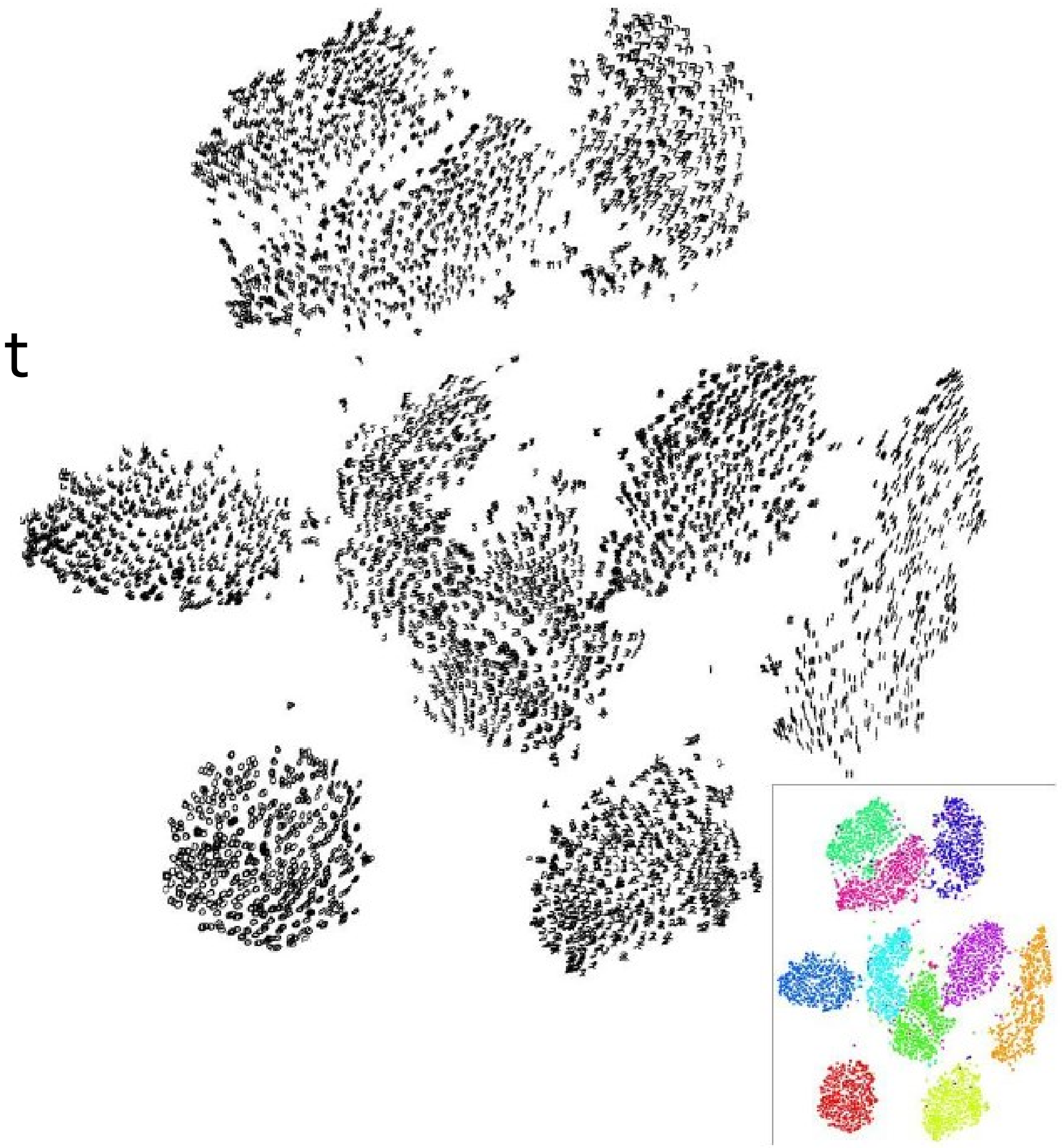
4096-dimensional "code" for an image
(layer immediately before the classifier)

can collect the code for many images

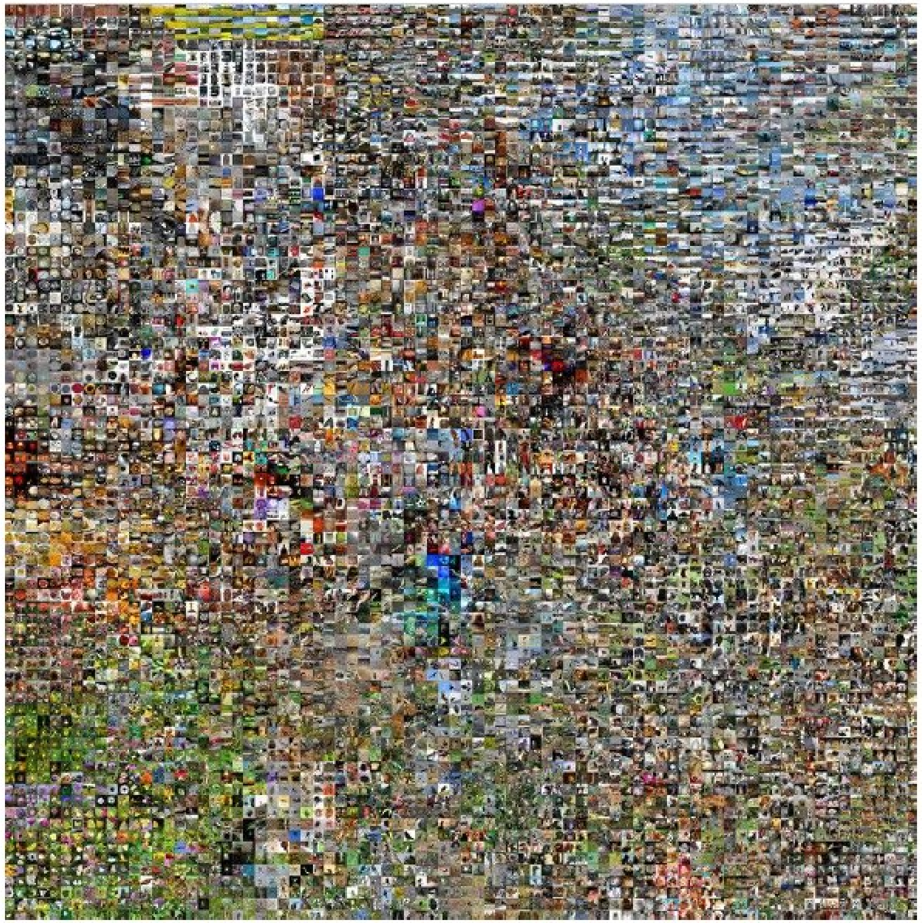
Slide Credit : Stanford CS231n

Last Layer: Dimensionality Reduction

- Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions
- Simple algorithm: Principle Component Analysis(PCA)
- More complex: t-SNE

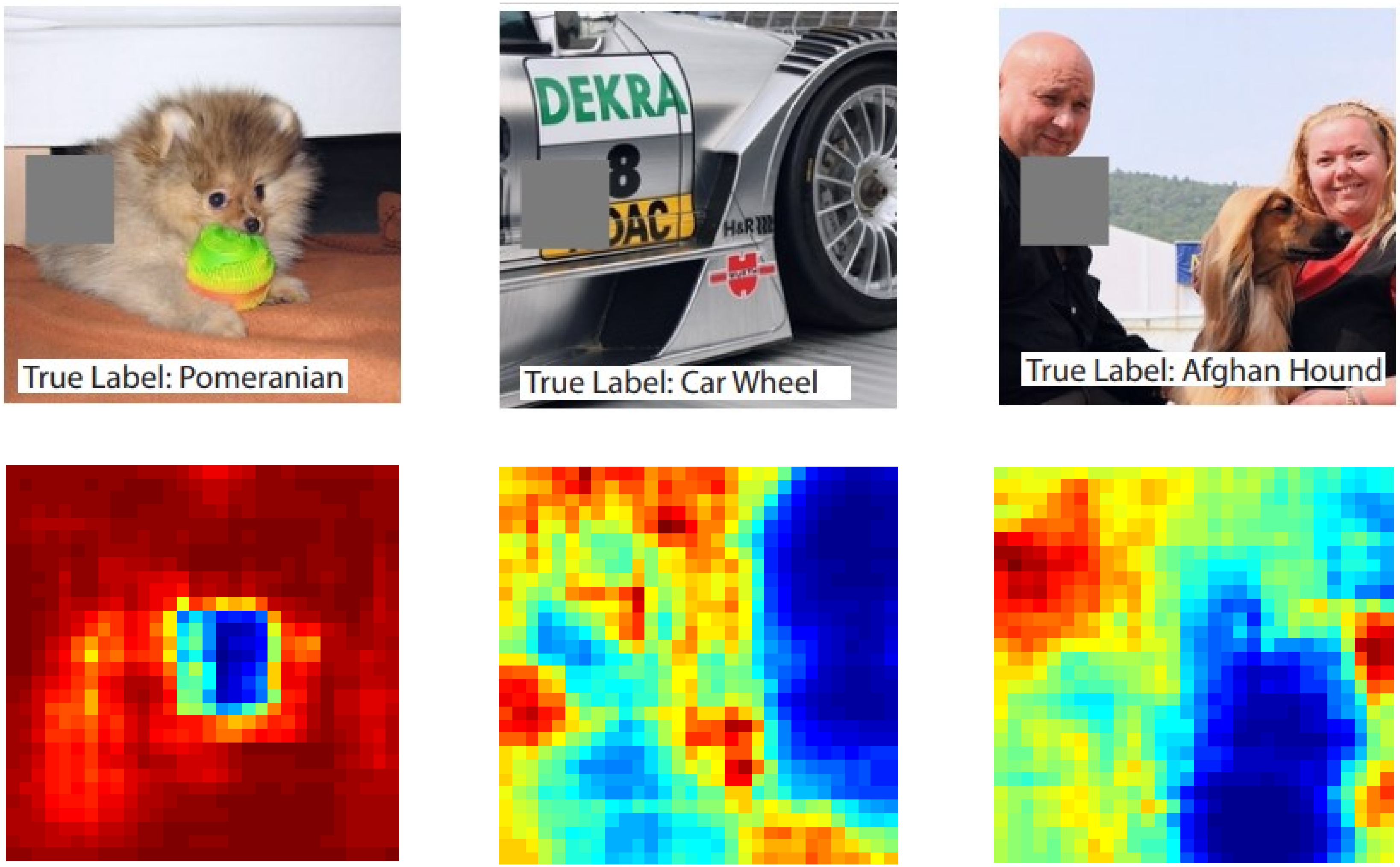


Last Layer: Dimensionality Reduction



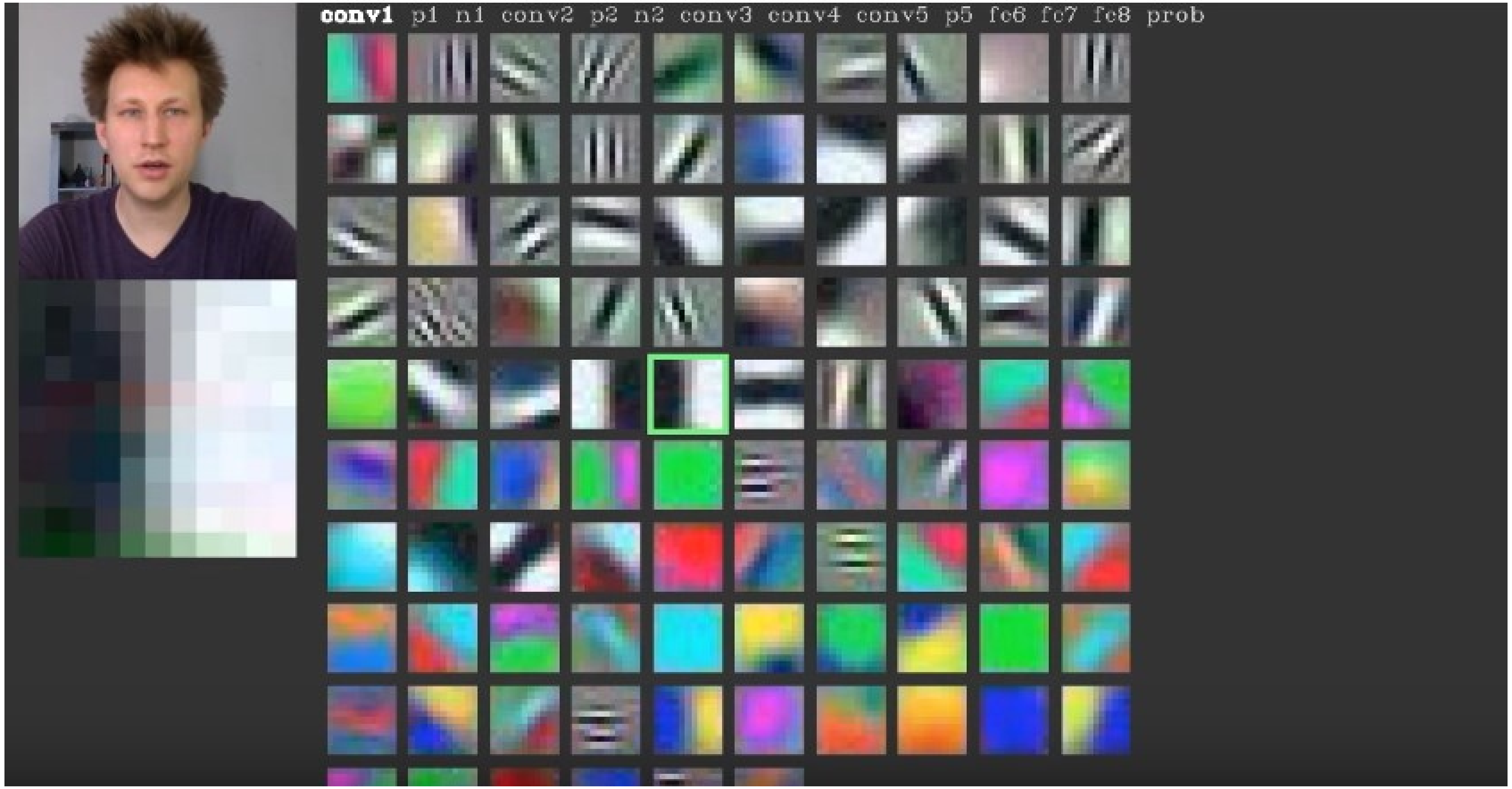
- http://cs.stanford.edu/people/karpathy/cnnembed/

Occlusion Experiments



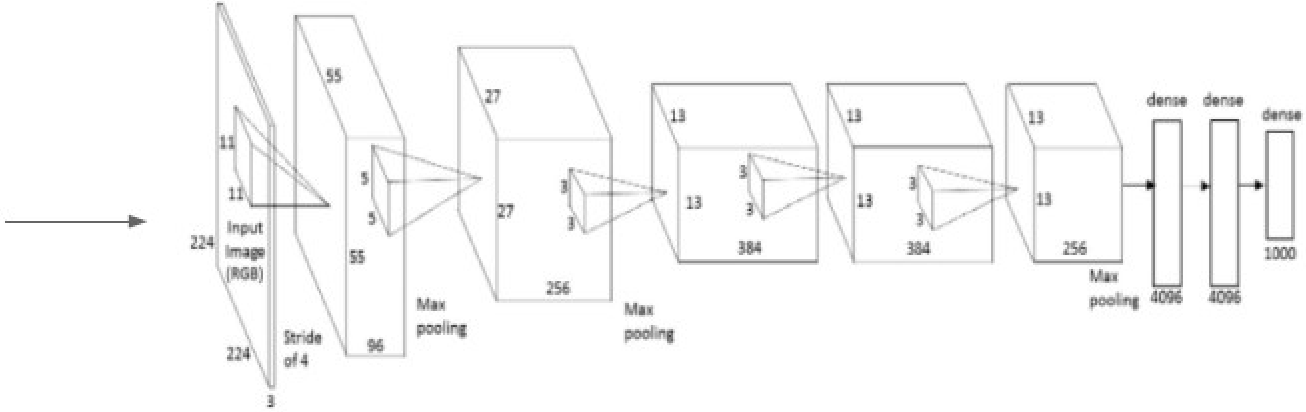
Visualizing Activations

- https://www.youtube.com/watch?v=AgkfIQ4IGaM

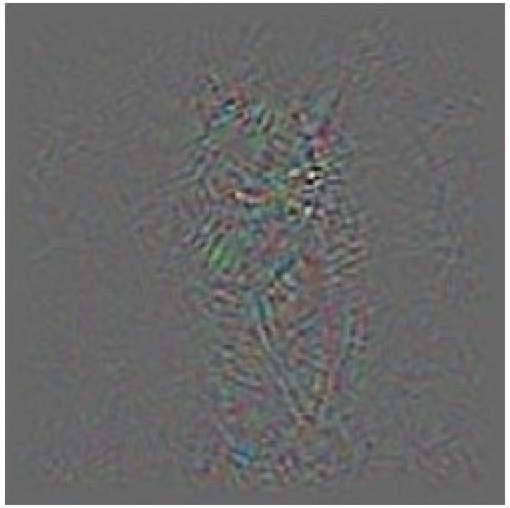


Deconv Approaches

1. Feed image into net

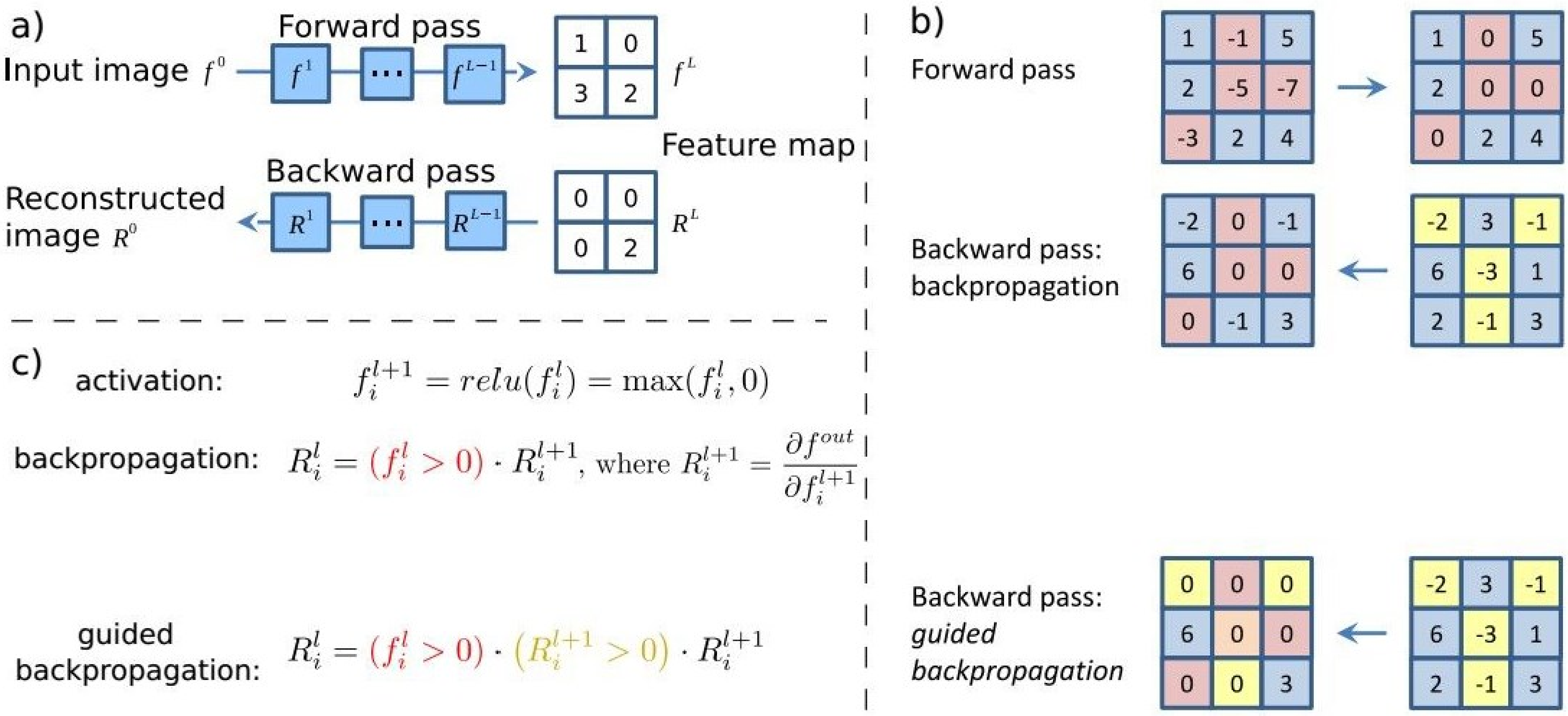


2. Pick a layer, set the gradient there to be all zero except for one 1 for some neuron of interest
3. Backprop to image:



“Guided backpropagation:” instead

Deconv Approaches

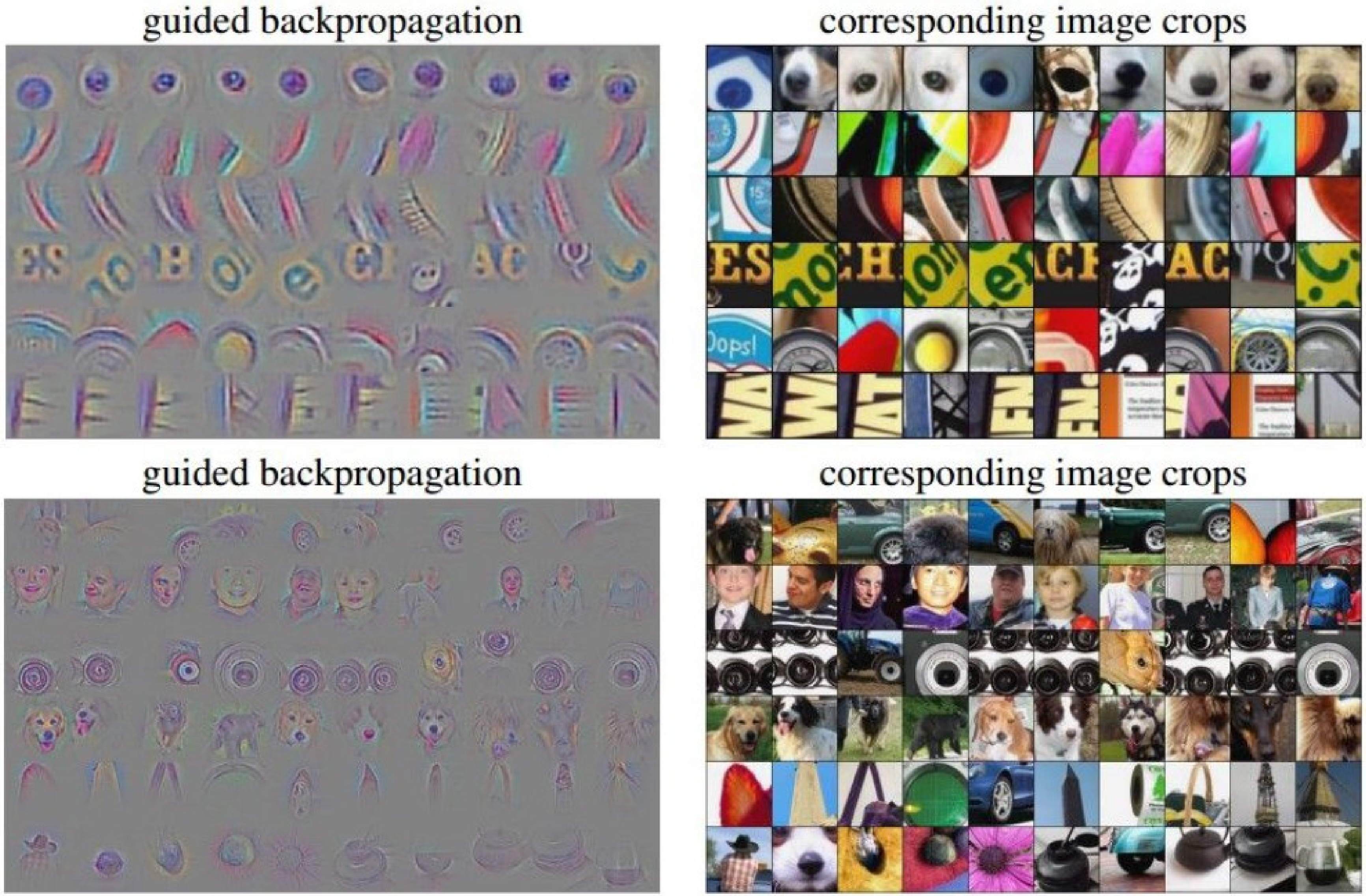


Deconv Approaches

Visualization of patterns learned by the layer **conv6** (top) and layer **conv9** (bottom) of the network trained on ImageNet.

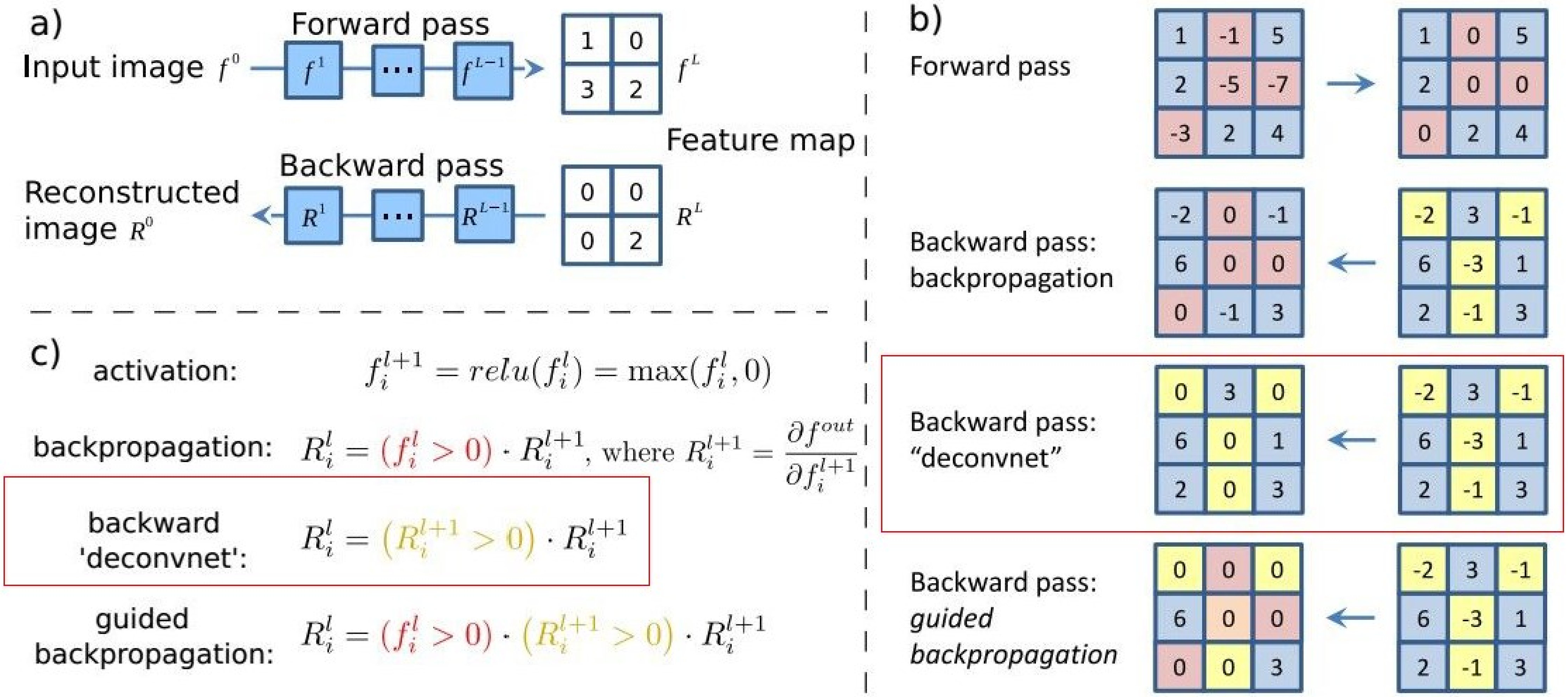
Each row corresponds to one filter.

The visualization using “guided backpropagation” is based on the top 10 image patches activating this filter taken from the ImageNet dataset.



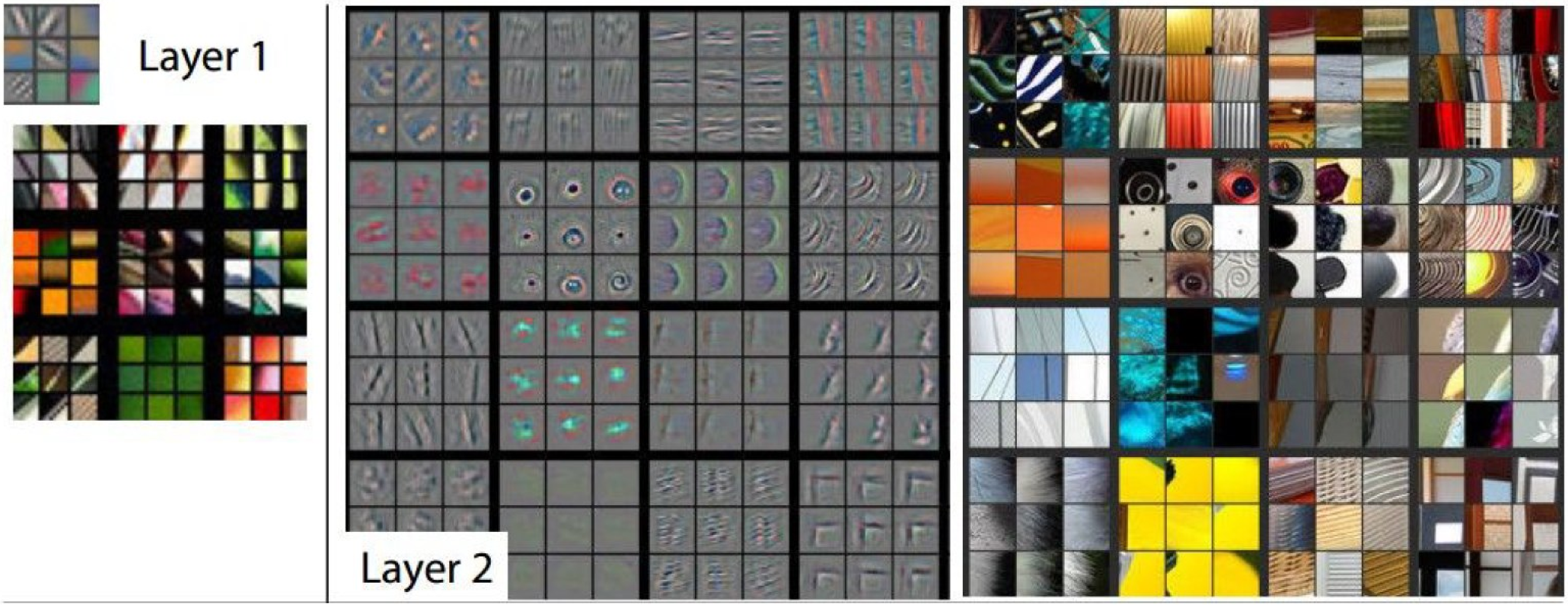
[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]

Deconv Approaches

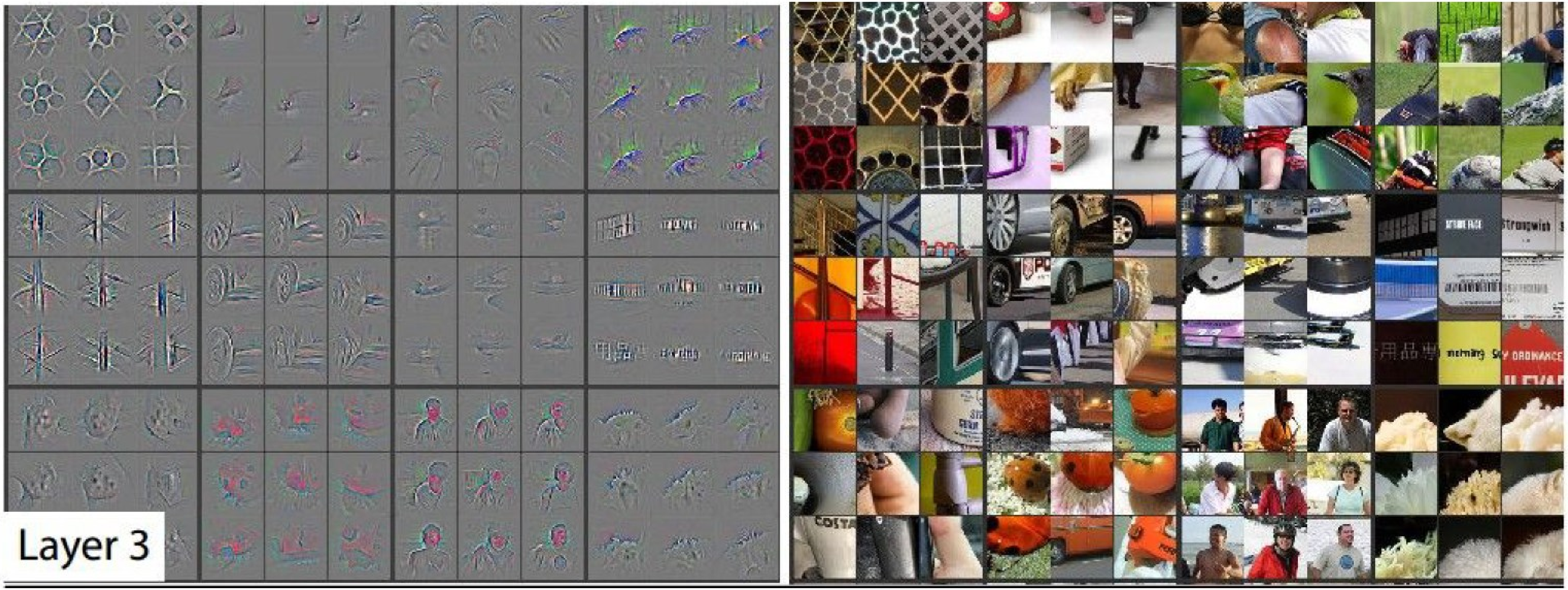


bit weird

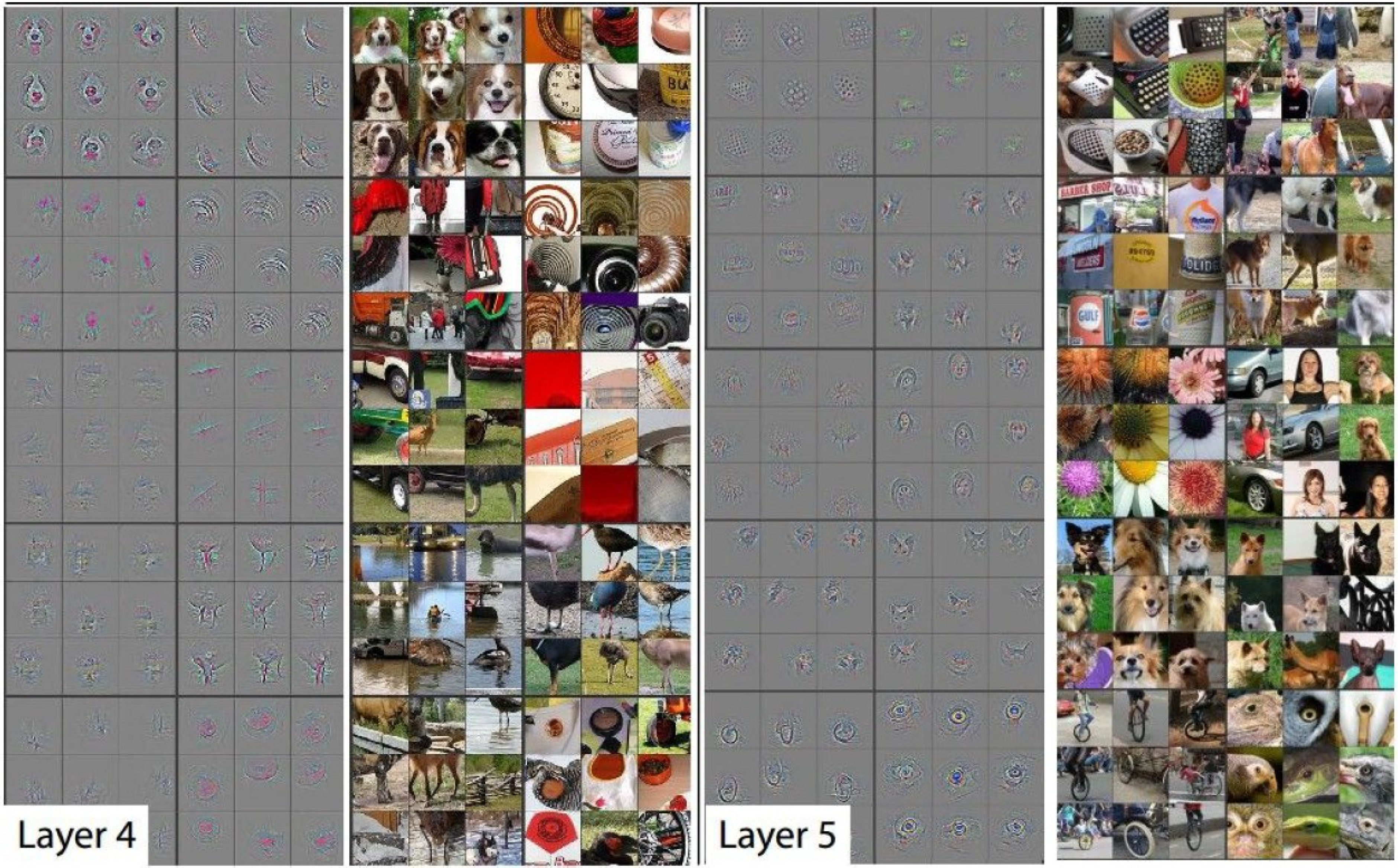
Visualizing Arbitrary Neurons



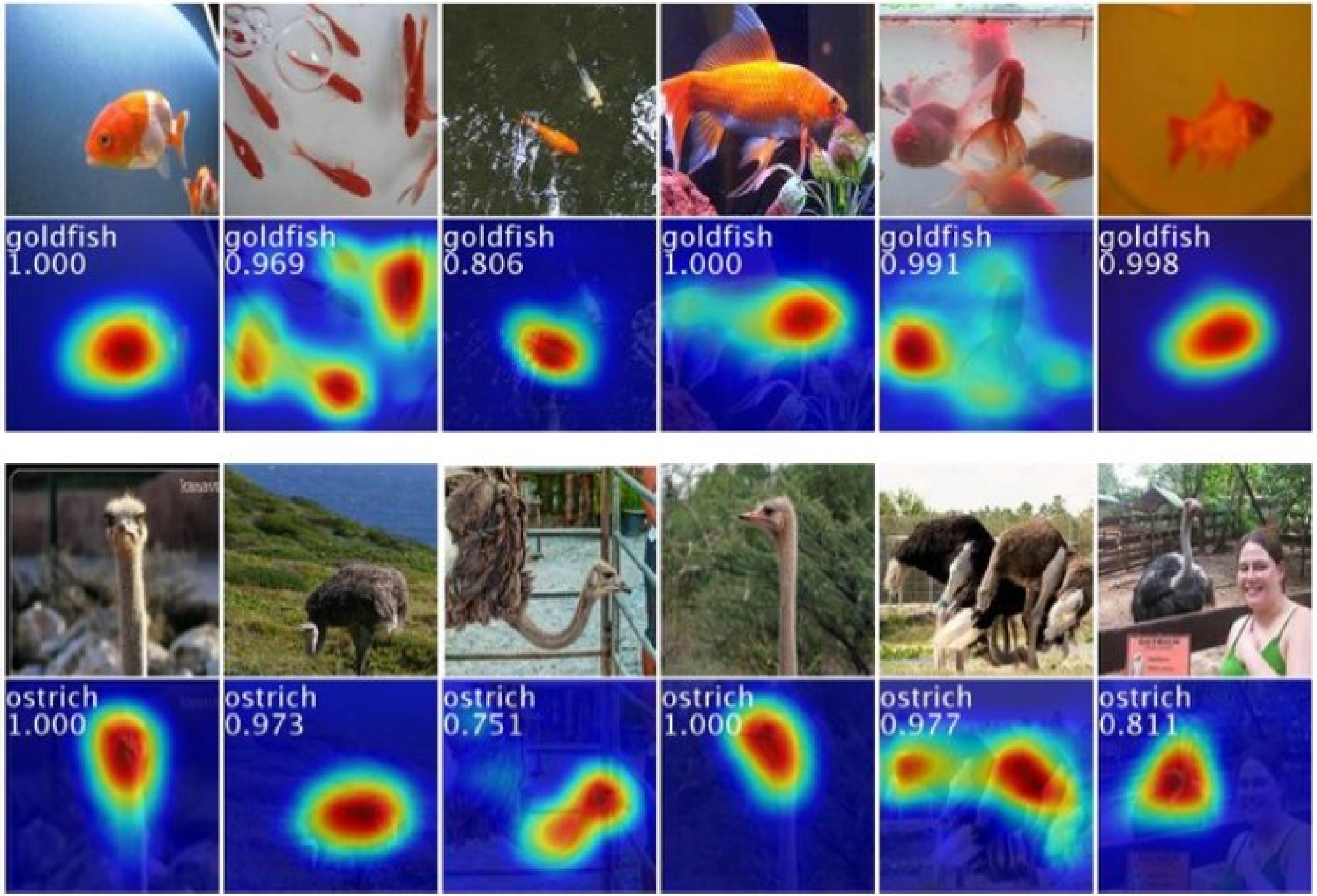
Visualizing Arbitrary Neurons



Visualizing Arbitrary Neurons



Weakly Supervised Learning

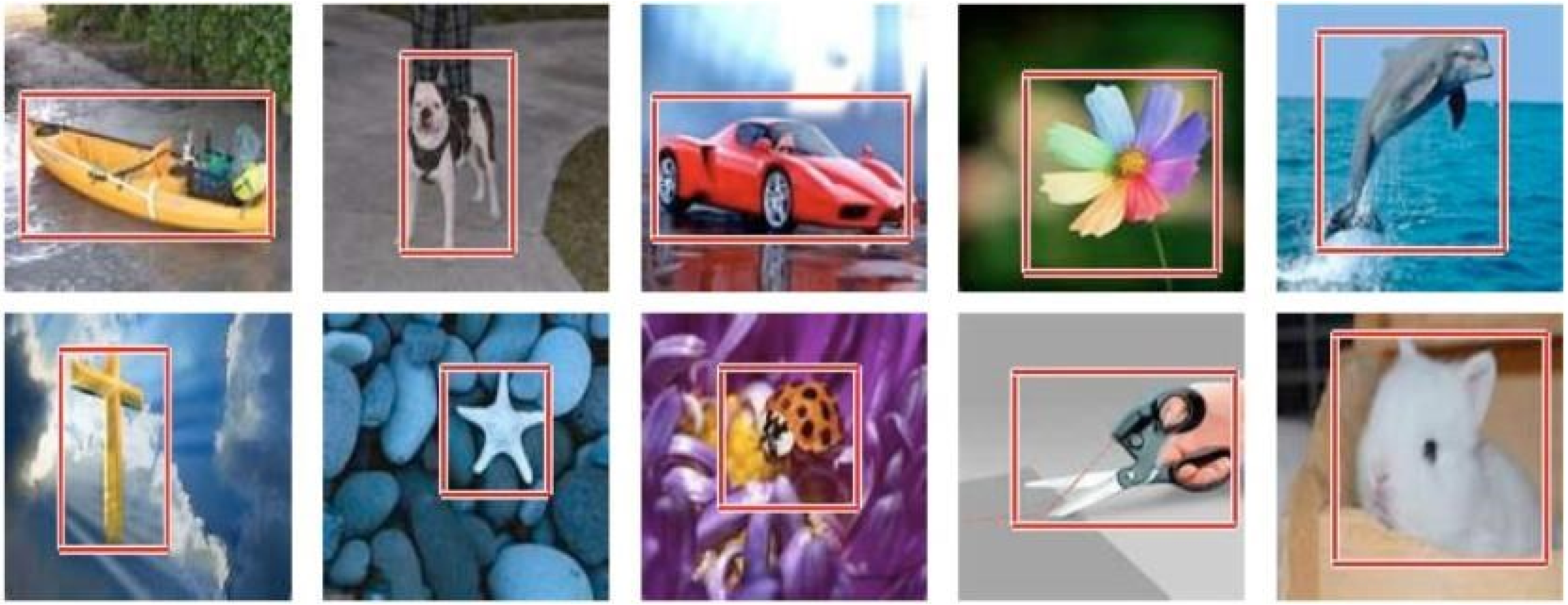


Introduction

Learning Deep Features for Discriminative Localization – CVPR2016

◆ **Weakly supervised object localization**

- Only trained with **class label** on image
- Yet able to localize object very well
- Close to fully supervised learned AlexNet

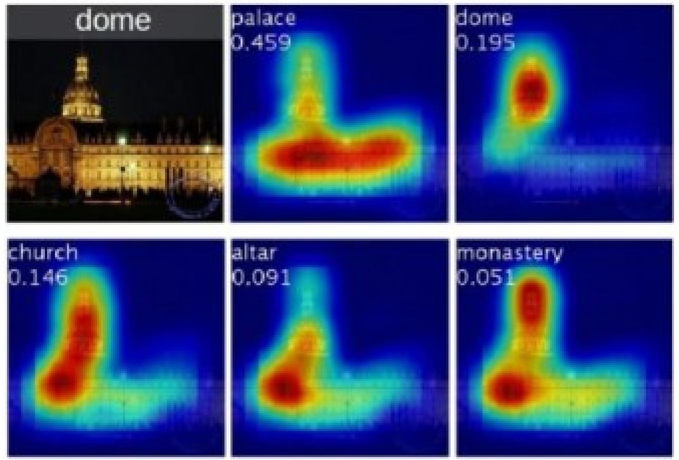


Introduction

Learning Deep Features for Discriminative Localization – CVPR2016

◆ **Visualizing the internal representation of CNNs.**

- Localization by **Class Activation Map (CAM)**
- Units activated by some visual pattern within its receptive field
- Visualize what activates for the output
- in One CNN forward pass.



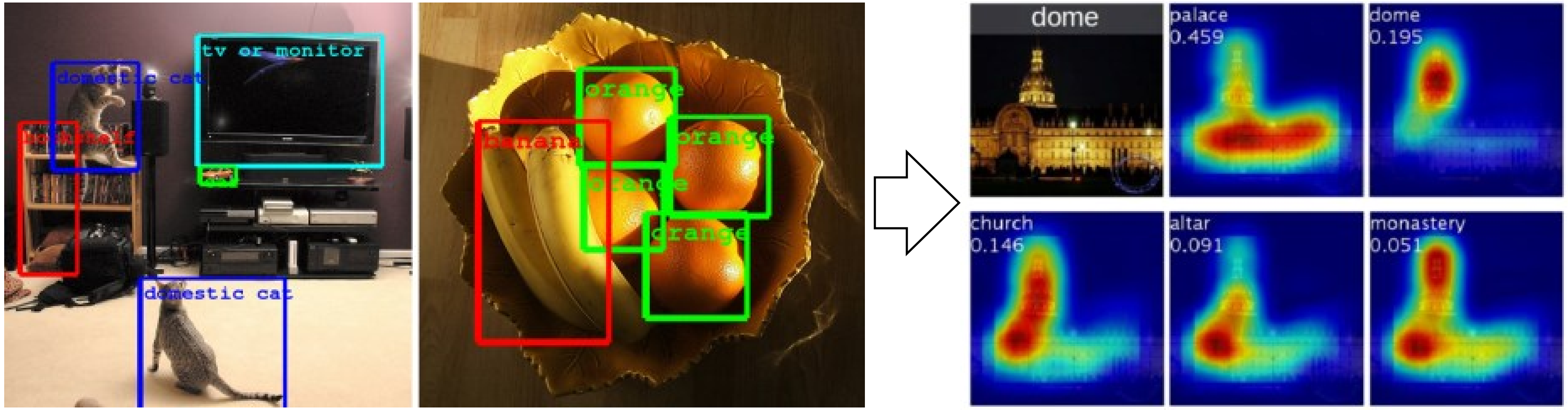
Class activation maps of top 5 predictions

Class activation maps for one object class

Weakly Supervised Object Localization

Usually **supervised learning** of localization is **annotated with bounding box**

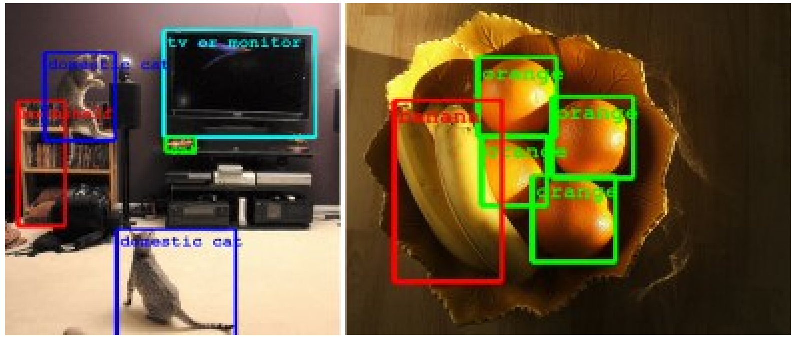
What if **localization is possible with image label** without bounding box annotations?



Today's seminar: Learning Deep Features for Discriminative Localization
1512.04150v1 Zhou *et al*. 2015 CVPR2016

Localization task (ILSVRC)

- Classification and **localize** its position
- ILSVRC LOC: 1000 classes and each object annotated with bounding box
  - Predict 5 class labels and 5 bounding boxes for each class label.

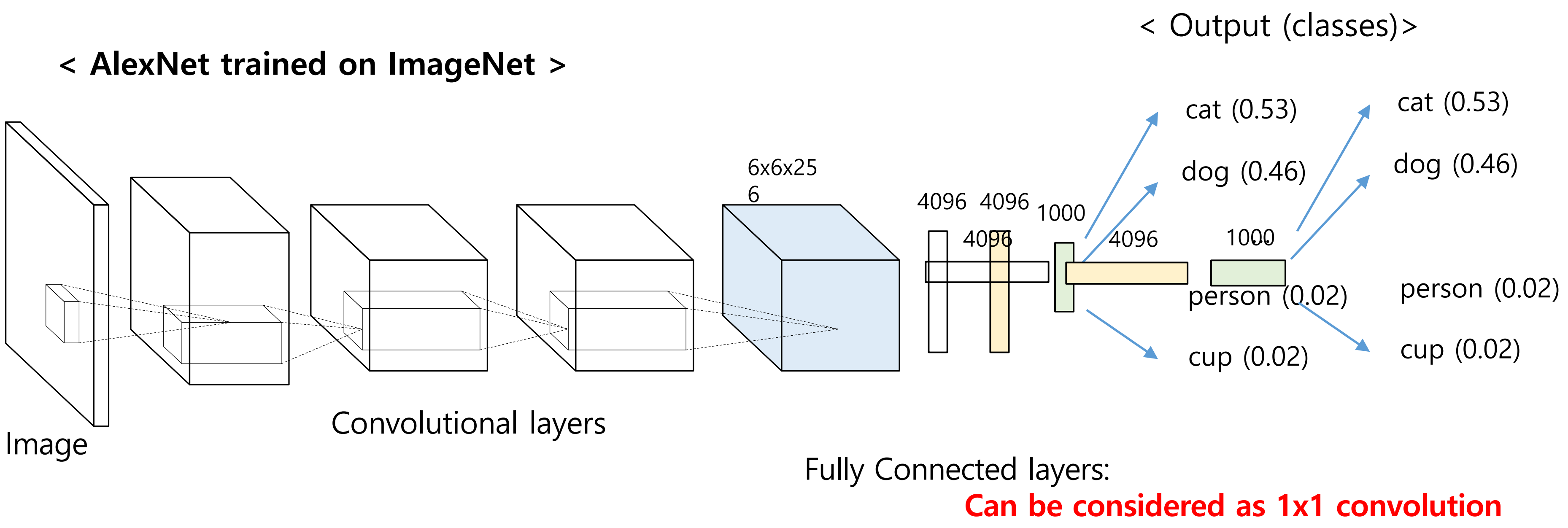


Supervised localization (VGGnet, Deep Residual Net)

- **Per-class regression for each class.**
  - learn a bounding box regressor (4D vector: x, y, w, h) for each class
- **Train image-level classifier** to predict class labels of an image.
  - **then localization** by predicting bounding boxes based on the predicted classes.
- Pre-train networks for ImageNet classification and then fine-tune for localization.

Easy approach: localization by classification

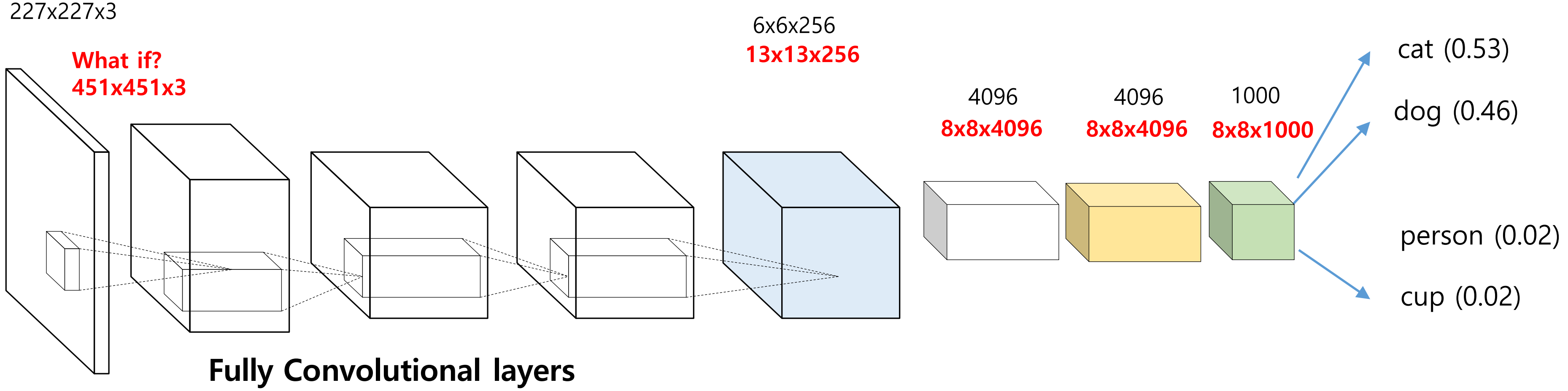
- Localization by **classification model** (AlexNet)
- Classification map with **Fully Convolutional Network** instead single classification.



Easy approach: localization by classification

< AlexNet trained on ImageNet >

< Output (classes)> **each 8x8 pixel classified as**



Fully Convolutional layers

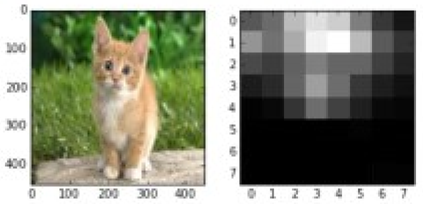
Blob(Tensor) sizes

```
In [5]: [(k, v.data.shape) for k, v in net_full_conv.blobs.items()]
Out[5]: [('data', (1, 3, 451, 451)),
         ('conv1', (1, 96, 111, 111)),
         ('pool1', (1, 96, 55, 55)),
         ('norm1', (1, 96, 55, 55)),
         ('conv2', (1, 256, 55, 55)),
         ('pool2', (1, 256, 27, 27)),
         ('norm2', (1, 256, 27, 27)),
         ('conv3', (1, 384, 27, 27)),
         ('conv4', (1, 384, 27, 27)),
         ('conv5', (1, 256, 27, 27)),
         ('pool5', (1, 256, 13, 13)),
         ('fc6-conv', (1, 4096, 8, 8)),
         ('fc7-conv', (1, 4096, 8, 8)),
         ('fc8-conv', (1, 1000, 8, 8)),
         ('prob', (1, 1000, 8, 8))]
```

```
out = net_full_conv.forward_all(data=np.asarray([transformer.preprocess('data', im)]))
print out['prob'][0].argmax(axis=0)
# show net input and confidence map (probability of the top prediction at each location)
plt.subplot(1, 2, 1)
plt.imshow(transformer.deprocess('data', net_full_conv.blobs['data'].data[0]))
plt.subplot(1, 2, 2)
plt.imshow(out['prob'][0,281])
```

```
[[282 282 281 281 281 281 277 282]
 [281 283 283 281 281 281 281 282]
 [283 283 283 283 283 283 287 282]
 [283 283 283 281 283 283 283 259]
 [283 283 283 283 283 283 283 259]
 [283 283 283 283 283 283 259 259]
 [283 283 283 283 259 259 259 277]
 [335 335 283 259 263 263 263 277]]
Out[11]: <matplotlib.image.AxesImage at 0x12379a690>
```

281 tiger cat
282 tabby
283 persian



Thus classification CNN already able to localize

Class activation map (CAM)

- **Identify important image regions** by projecting back the weights of output layer to convolutional feature maps.
- CAMs can be generated for each class in single image.
- Regions for each categories are different in given image.
  - palace, dome, church …

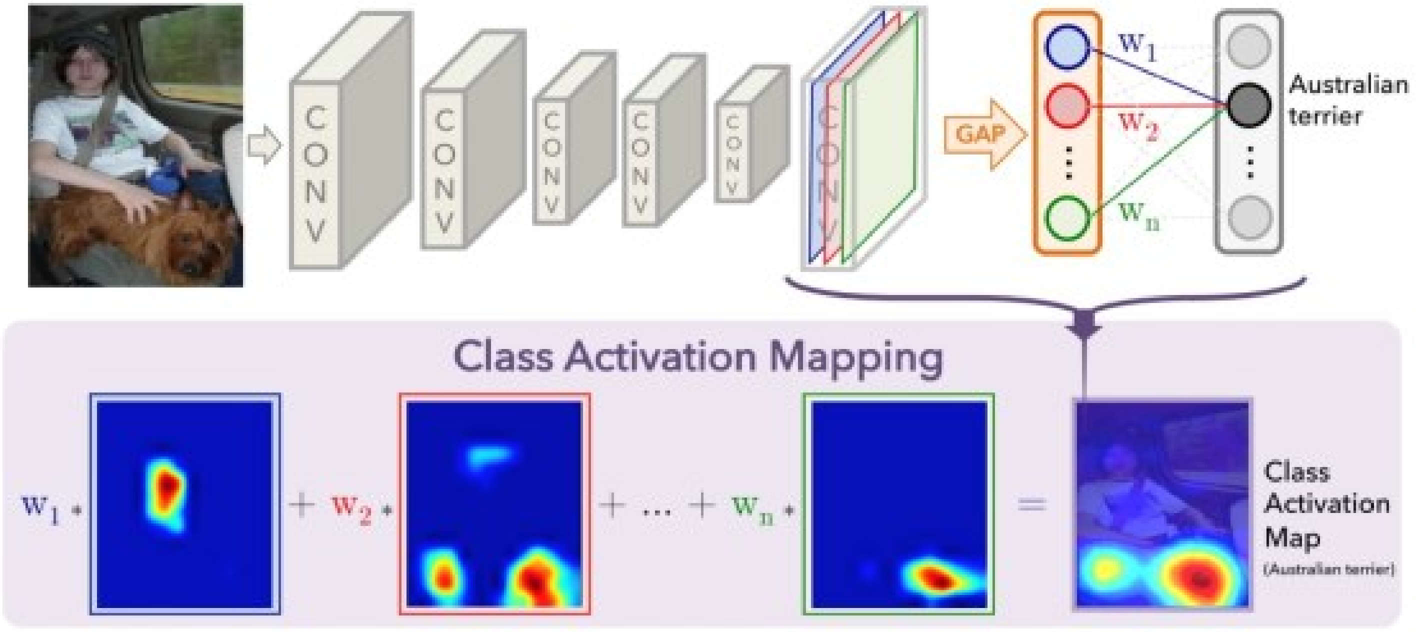


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Results

- CAM on top 5 predictions on an image
- CAM for one object class in images

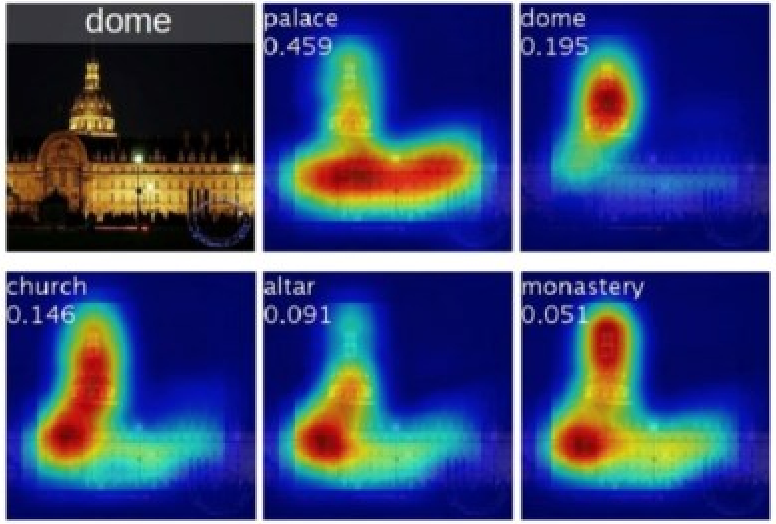


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper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

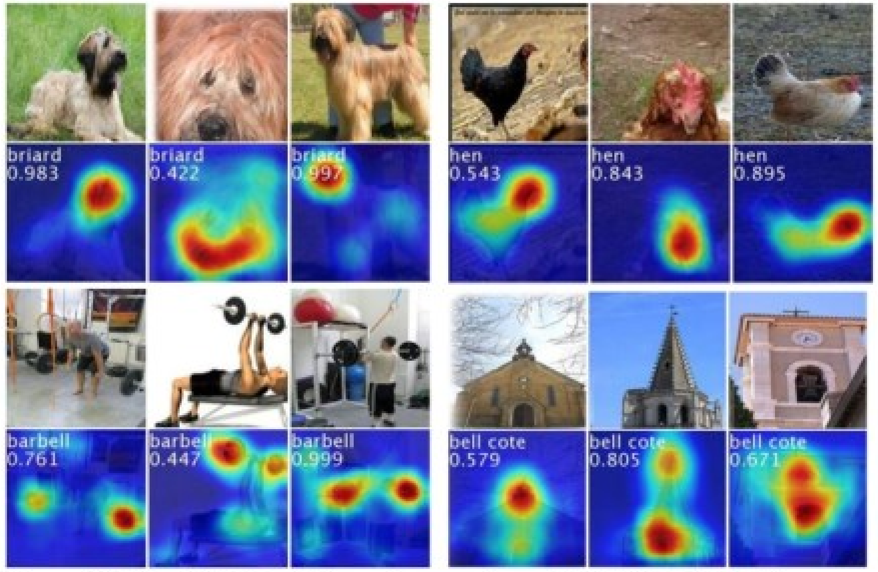
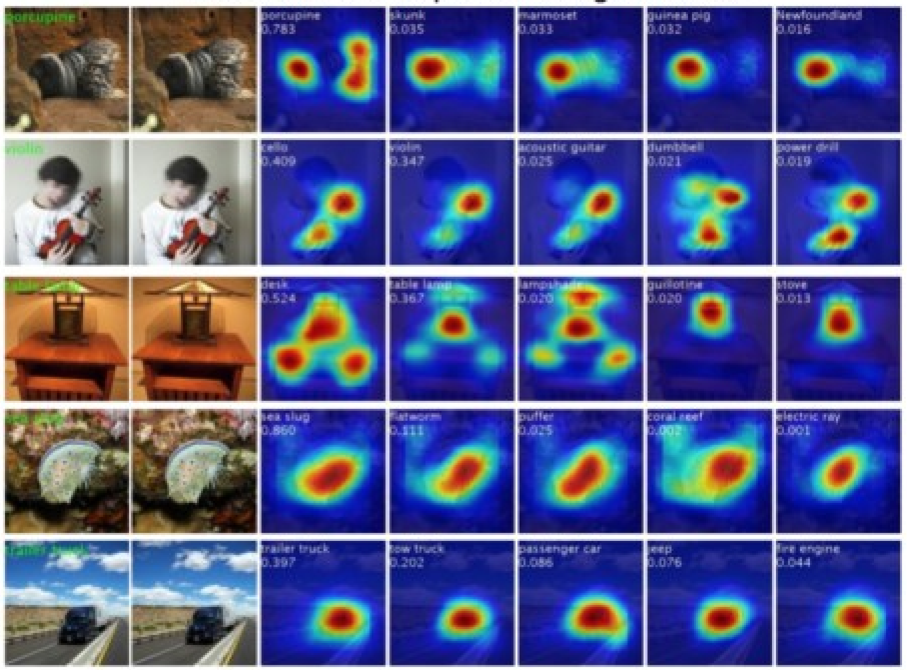


Figure 3. The CAMs of four classes from ILSVRC [20]. The maps highlight the discriminative image regions used for image classification e.g., the head of the animal for *briard* and *hen*, the plates in *barbell*, and the bell in *bell cote*.

GAP & GMP

- GAP (upper) vs GMP (lower)
- GAP outperforms GMP
- GAP highlights more **complete** object regions and less background noise.
- Loss for average pooling benefits when the network identifies ***all* discriminative** regions of an object

Class activation maps from GoogLeNet-GAP



Class activation maps from GoogLeNet-GMP

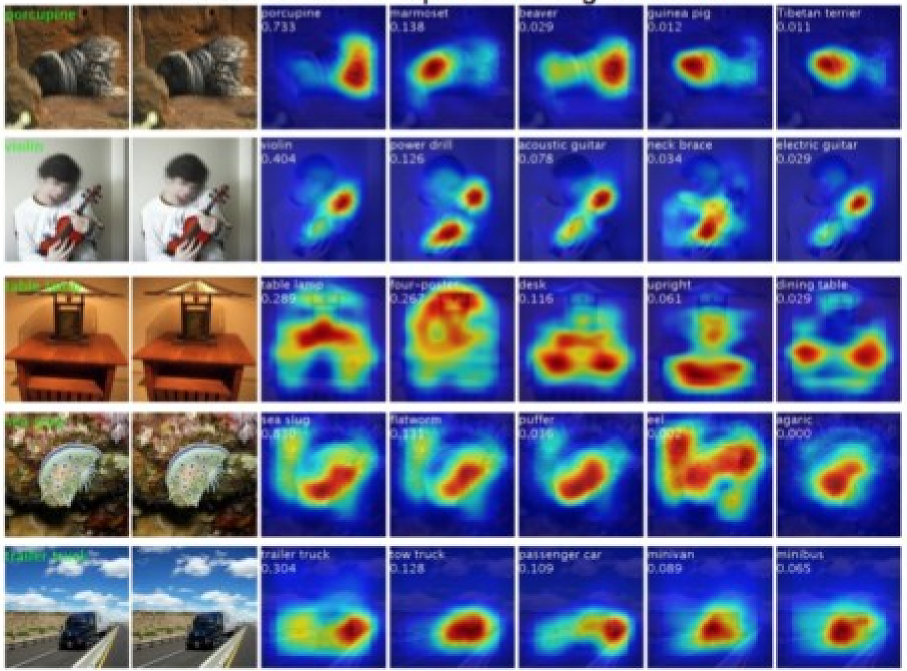


Table 1. Classification error on the ILSVRC validation

Networks	top-1 val. error	top-5 val. error
VGGnet-GAP	33.4	12.2
GoogLeNet-GAP	35.0	13.2
AlexNet*-GAP	44.9	20.9
AlexNet-GAP	51.1	26.3
GoogLeNet	31.9	11.3
VGGnet	31.2	11.4
AlexNet	42.6	19.5
NIN	41.9	19.6
GoogLeNet-GMP	35.6	13.9

Table 2. Localization error on the ILSVRC validation set. *Backprop* refers to using [22] for localization instead of CAM.

Method	top-1 val.error	top-5 val. error
GoogLeNet-GAP	**56.40**	**43.00**
VGGnet-GAP	57.20	45.14
GoogLeNet	60.09	49.34
AlexNet*-GAP	63.75	49.53
AlexNet-GAP	67.19	52.16
NIN	65.47	54.19
Backprop on GoogLeNet	61.31	50.55
Backprop on VGGnet	61.12	51.46
Backprop on AlexNet	65.17	52.64
GoogLeNet-GMP	57.78	45.26