Visualizing and Understanding Convolutional Networks [1]

Yu Wu

2017.10.31

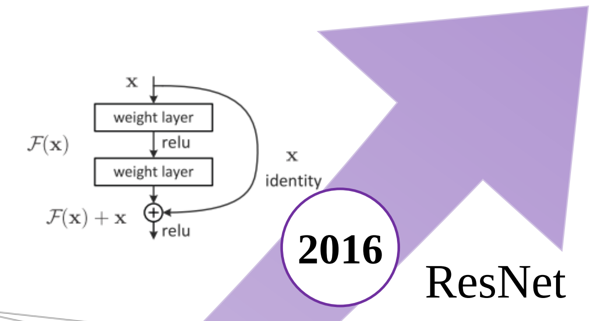
[1] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]. European conference on computer vision. 2014: 818-833.

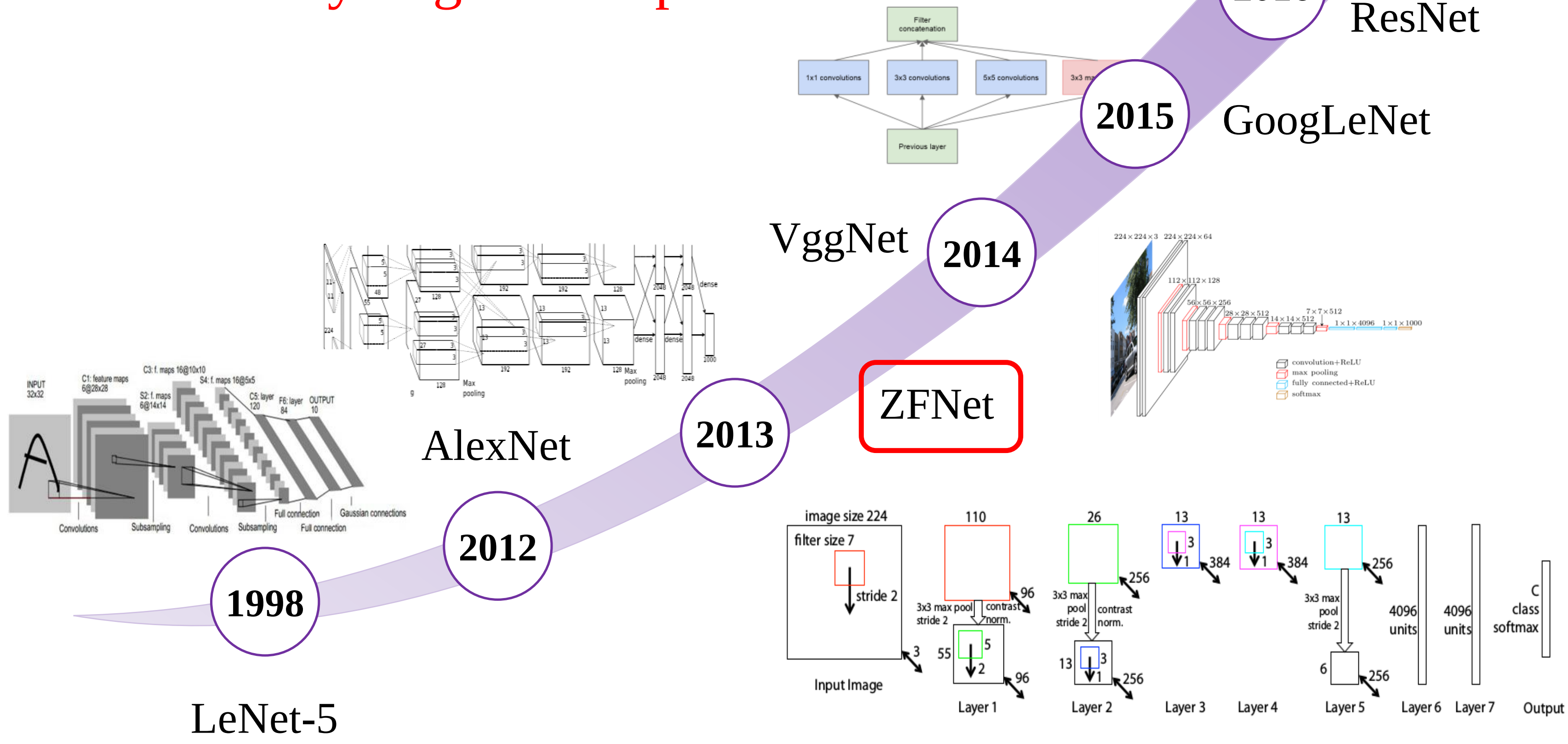
Outline

- Visualization Approaches
- Convnet Visualization
- Experiments
- My Experiments

Convnets

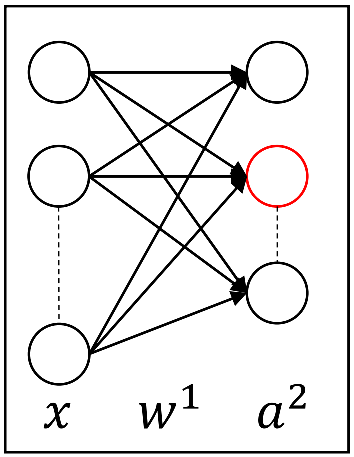
1. Why they perform so well?
2. How they might be improved?





Visualization Approaches

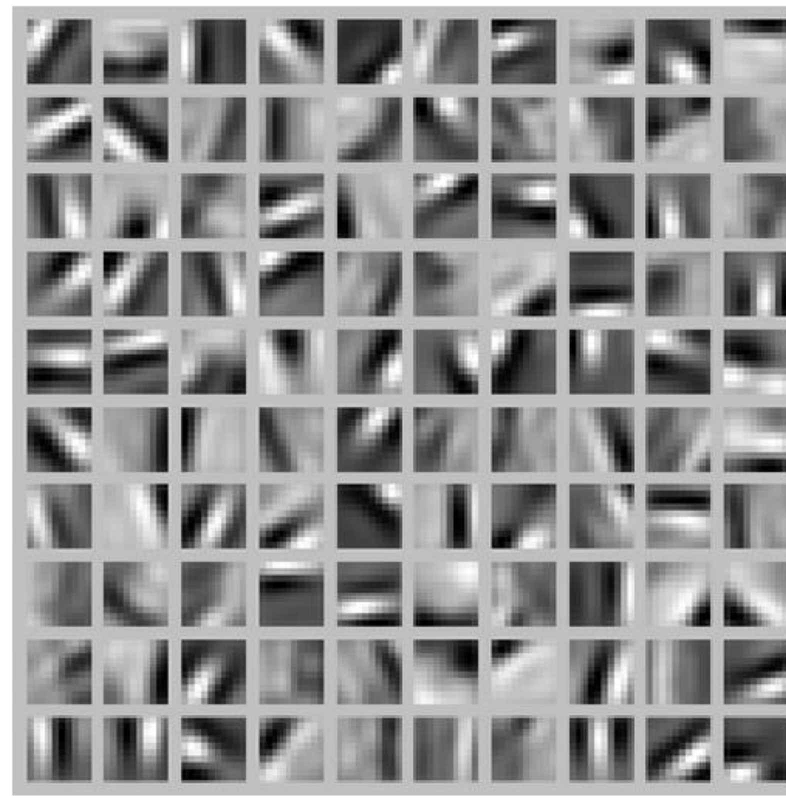
What input image x cause a_i^2 to be maximally activated?



$$a_i^2 = f\left(\sum_{j=1}^{n} W_{ij}^1 x_j\right)$$

$$\begin{cases} \max \sum_{j=1}^{n} W_{ij}^1 x_j \\ s.t. \sum_{j=1}^{n} x_i^2 \leq 10 \end{cases}$$

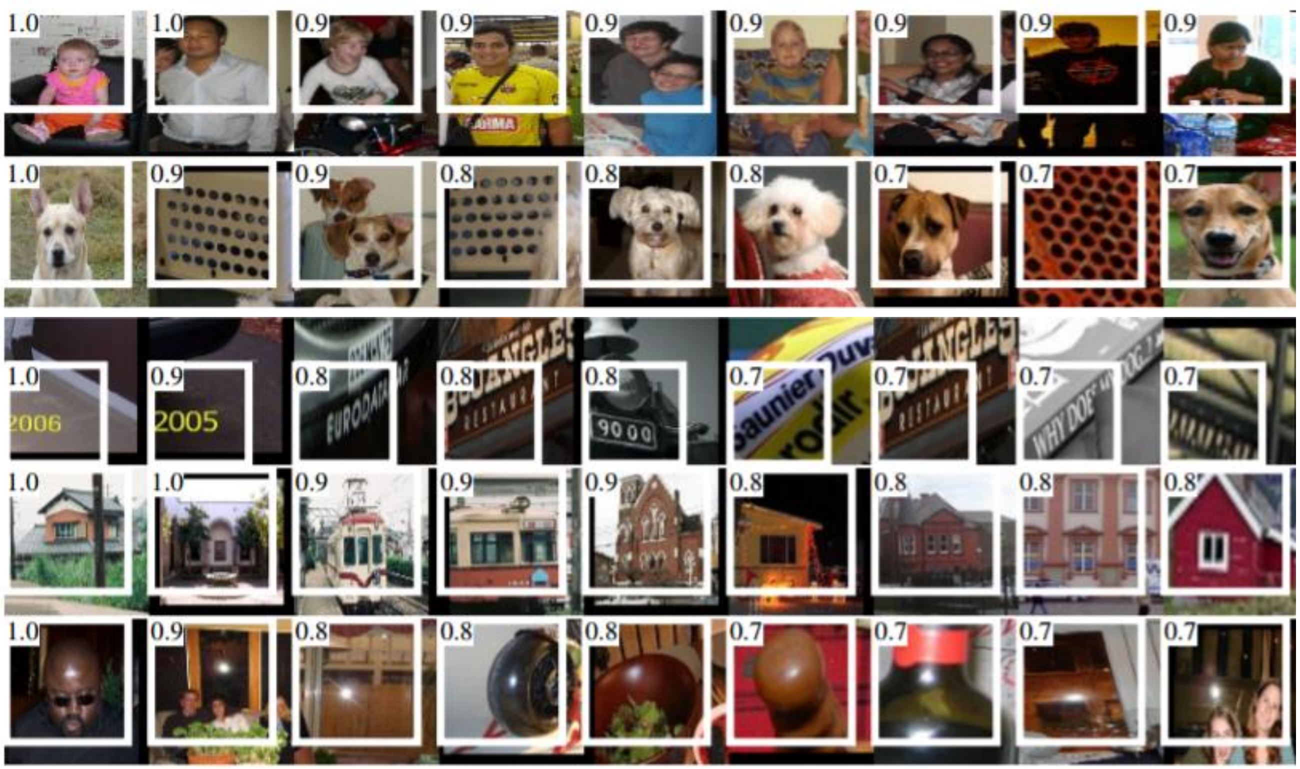
$$x_{\mathrm{j}} = \frac{W_{\mathrm{ij}}^1}{\sqrt{\sum_{j=1}^{n}\left(W_{ij}^1\right)^2}}$$

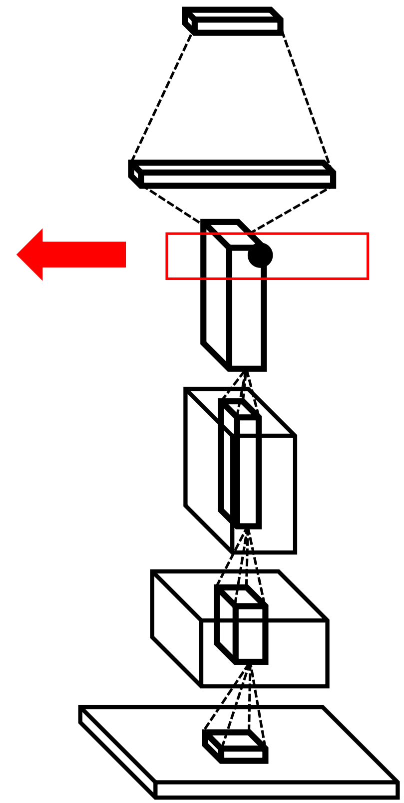


Edges at different positions and orientations

Visualization Approaches

To visualize images that maximally activate certain units [2]





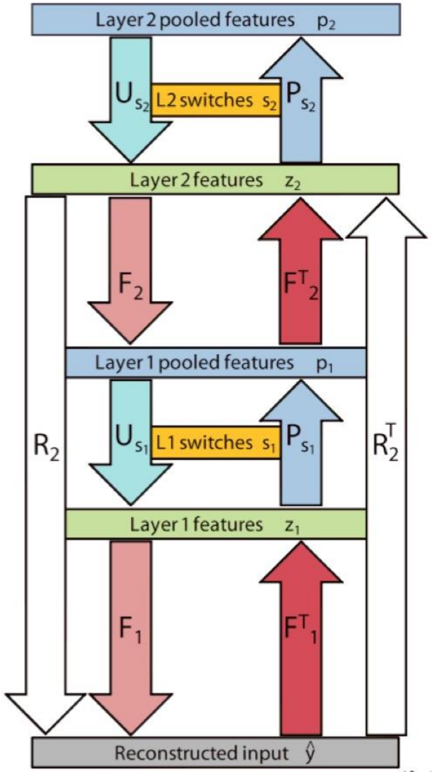
ConvNet

[2] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014: 580-587.

Visualization Approaches

Convnet: mapping image pixels to feature representation in network

Deconvnet: mapping feature maps back to the input pixel space, unsupervised learning[3]



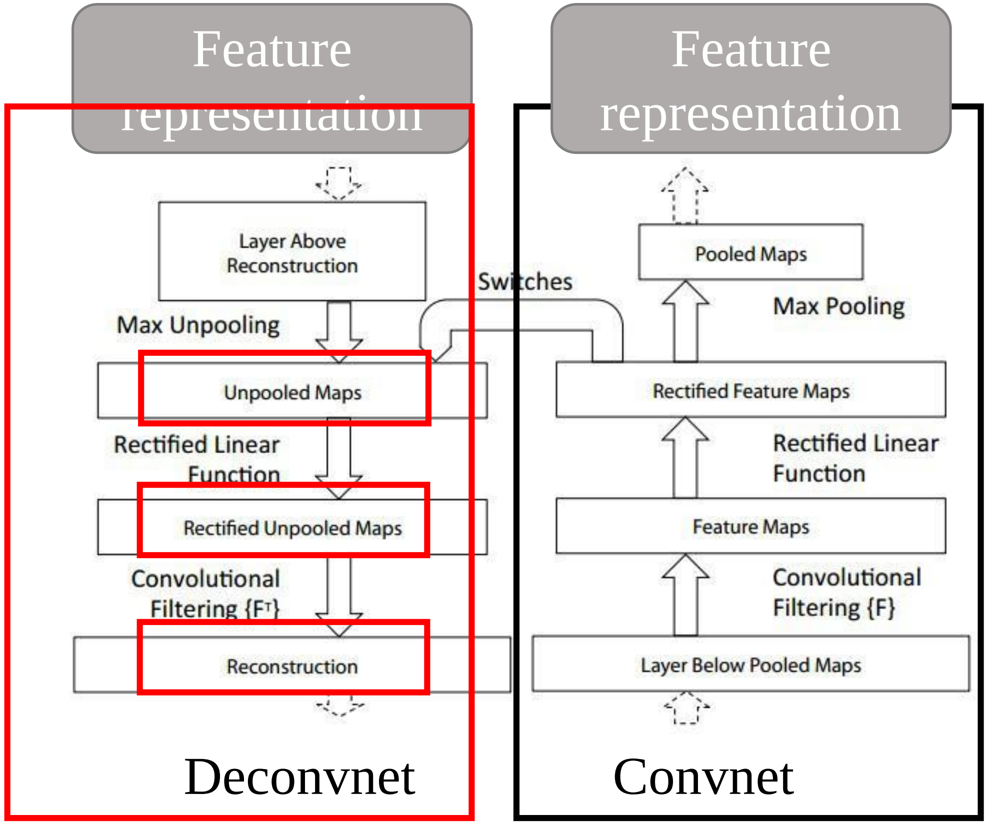
The target for feature map i in the layer l :

$$\underset{f_i^l}{\arg\min} \underbrace{\|y - \hat{y}\|_2^2}_{\text{Reconstruction error}} + \underbrace{\sum_{i=1}^{n_l} \|z_{i,l}\|}_{\text{Sparsity constrain}}$$

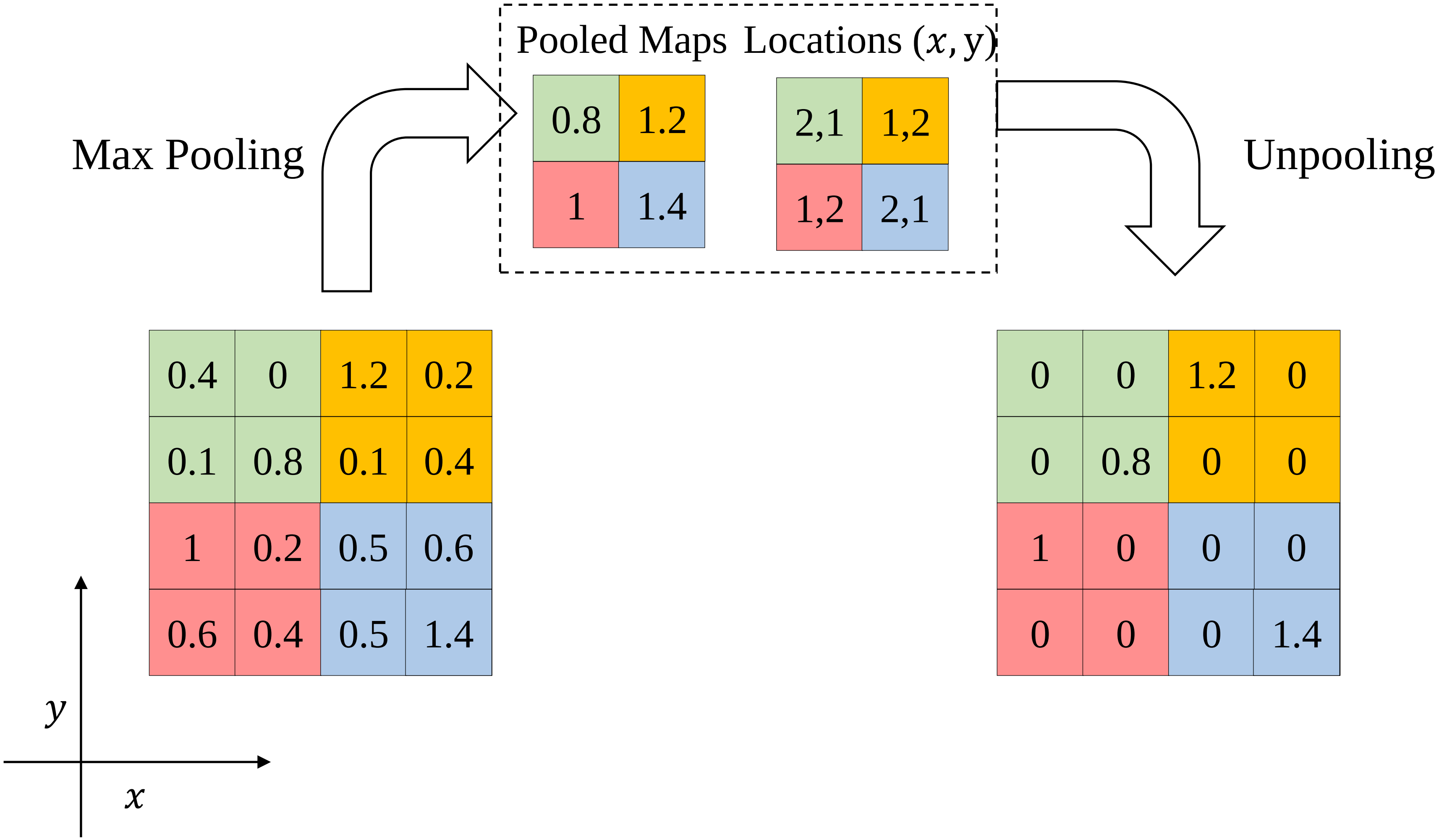
[3] Zeiler, M. D., Taylor, G. W., Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In ICCV, 2011: 2018-2025

Visualization Approaches

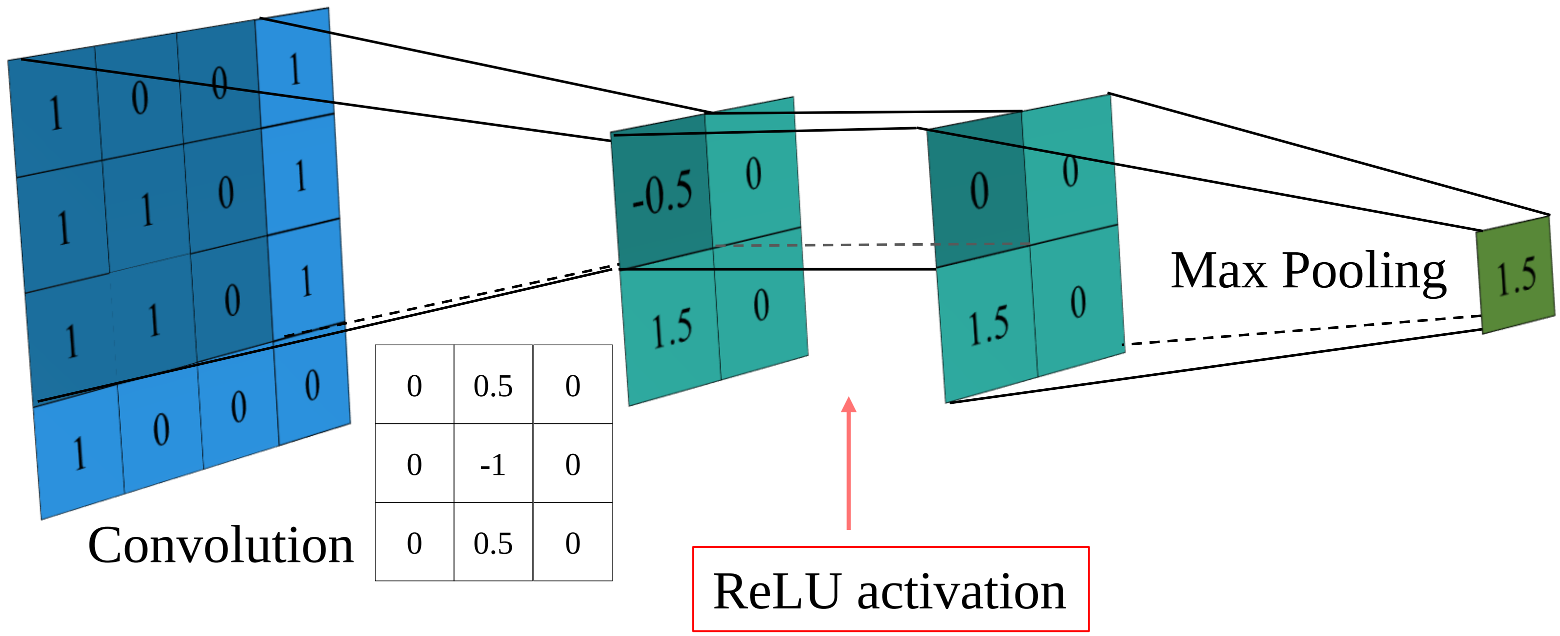
Deconvnets are not used in any learning capacity, just as a probe of an already trained convnet.



Unpooling


Pooled Maps
Locations (x, y)
Max Pooling
Unpooling
0.8
1.2
1
1.4
2,1
1,2
1,2
2,1
0.4
0
1.2
0.2
0.1
0.8
0.1
0.4
1
0.2
0.5
0.6
0.6
0.4
0.5
1.4
0
0
1.2
0
0
0.8
0
0
1
0
0
0
0
0
0
1.4
y
x

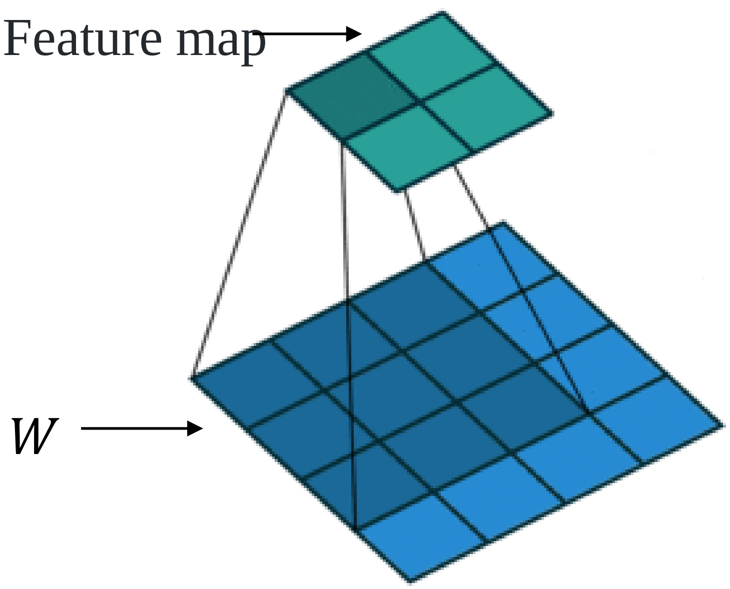
Rectification



Ensuring the feature maps are always positive

Filtering

No padding, no strides



Convet: filter W

Deconvet: transposed filter W^T

Flipping W vertically and horizontally

Convolution

1d CNNs feedforward: no padding, no strides

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \otimes \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

$$z_1 = w_1 x_1 + w_2 x_2 + w_3 x_3$$
$$z_2 = w_1 x_2 + w_2 x_3 + w_3 x_4$$
$$z_3 = w_1 x_3 + w_2 x_4 + w_3 x_5$$

Matrix formulation:

$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Transposed Convolution

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \otimes \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Convolution

Deconvolution:

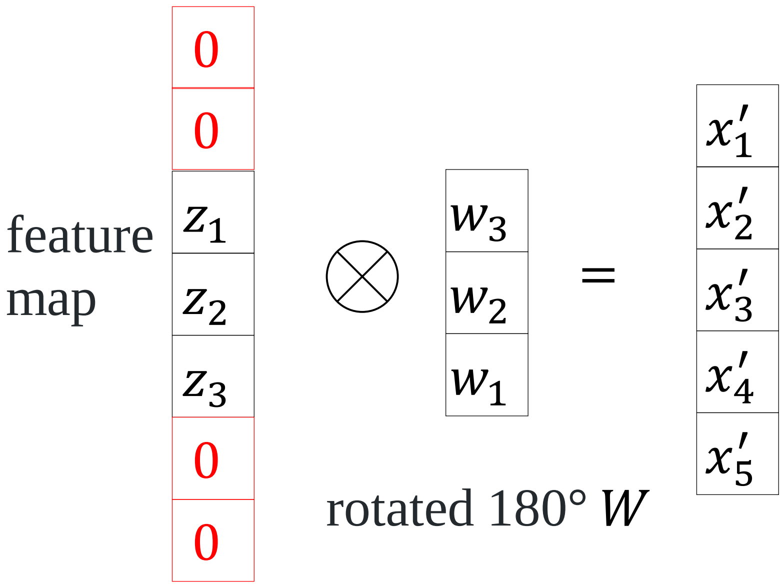
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \otimes \begin{bmatrix} w_3 \\ w_2 \\ w_1 \end{bmatrix} \xrightarrow{?} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ x'_5 \end{bmatrix}$$

feature map   rotated 180° W

$$\begin{bmatrix} 0 \\ 0 \\ z_1 \\ z_2 \\ z_3 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} w_3 \\ w_2 \\ w_1 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ x'_5 \end{bmatrix} \Rightarrow$$

$$\begin{aligned} x'_1 &= w_1 z_1 \\ x'_2 &= w_2 z_1 + w_1 x_2 \\ x'_3 &= w_3 z_1 + w_2 z_2 + w_1 z_1 \\ x'_4 &= w_3 z_2 + w_2 z_3 \\ x'_4 &= w_3 z_3 \end{aligned}$$

Transposed Convolution



$$
\begin{aligned}
x_1' &= w_1 z_1 \\
x_2' &= w_2 z_1 + w_1 x_2 \\
x_3' &= w_3 z_1 + w_2 z_2 + w_1 z_1 \\
x_4' &= w_3 z_2 + w_2 z_3 \\
x_4' &= w_3 z_3
\end{aligned}
$$

Matrix formulation:

$$
\begin{bmatrix} w_1 & 0 & 0 \\ w_2 & w_1 & 0 \\ w_3 & w_2 & w_1 \\ 0 & w_3 & w_2 \\ 0 & 0 & w_3 \end{bmatrix} \times \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \end{bmatrix}
$$

Transposed Convolution

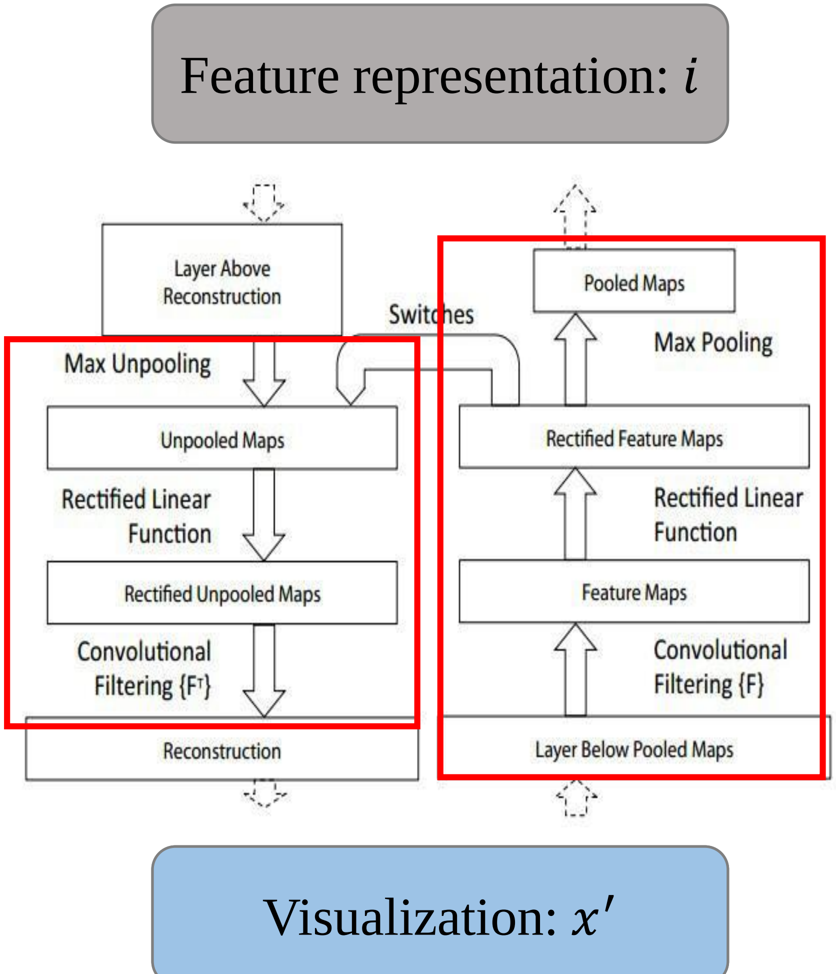
Matrix formulation 1:

$$\boxed{\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Matrix formulation 2:

$$\boxed{\begin{bmatrix} w_1 & 0 & 0 \\ w_2 & w_1 & 0 \\ w_3 & w_2 & w_1 \\ 0 & w_3 & w_2 \\ 0 & 0 & w_3 \end{bmatrix}} \times \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \end{bmatrix}$$

Visualization Approaches



For a given image x and a fully trained model, to visualize the i-th feature map in the layer l:

1, for a given x, feedforward computing feature activities in the layer l

2, set all other activations (not the i-th feature map) in the layer to zero

3, for layer l to the first layer:

if the layer beneath is pooling layer:

use **Unpooling** operation

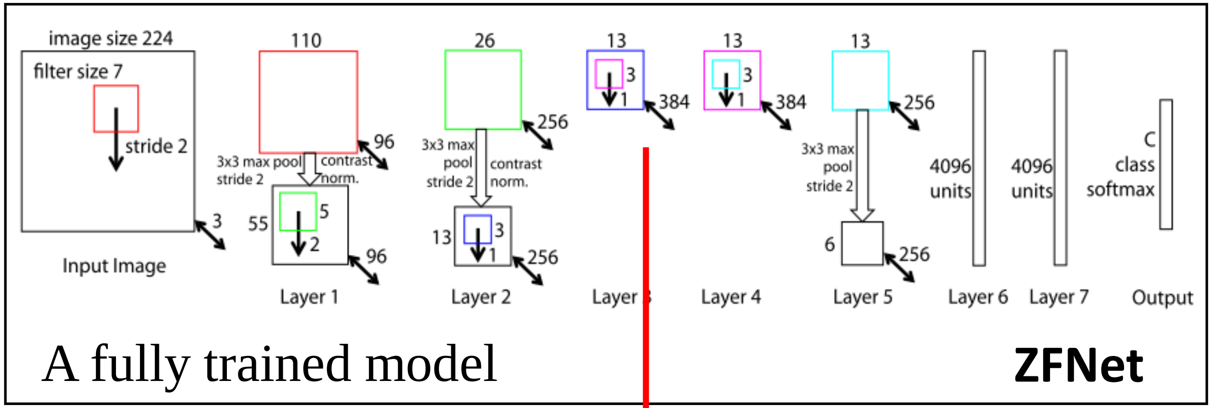
else if the layer beneath is convolutional layer:

use **Rectify** first **and Transposed filters** operation

Outline

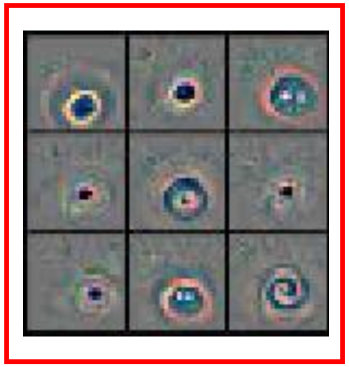
- Visualization Approaches
- Convnet Visualization
- Experiments
- My Experiments

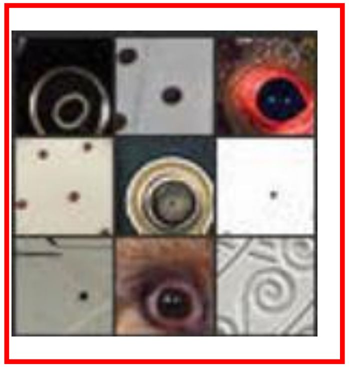
Feature Visualization



Choosing a feature map and
recording the top 9 activities

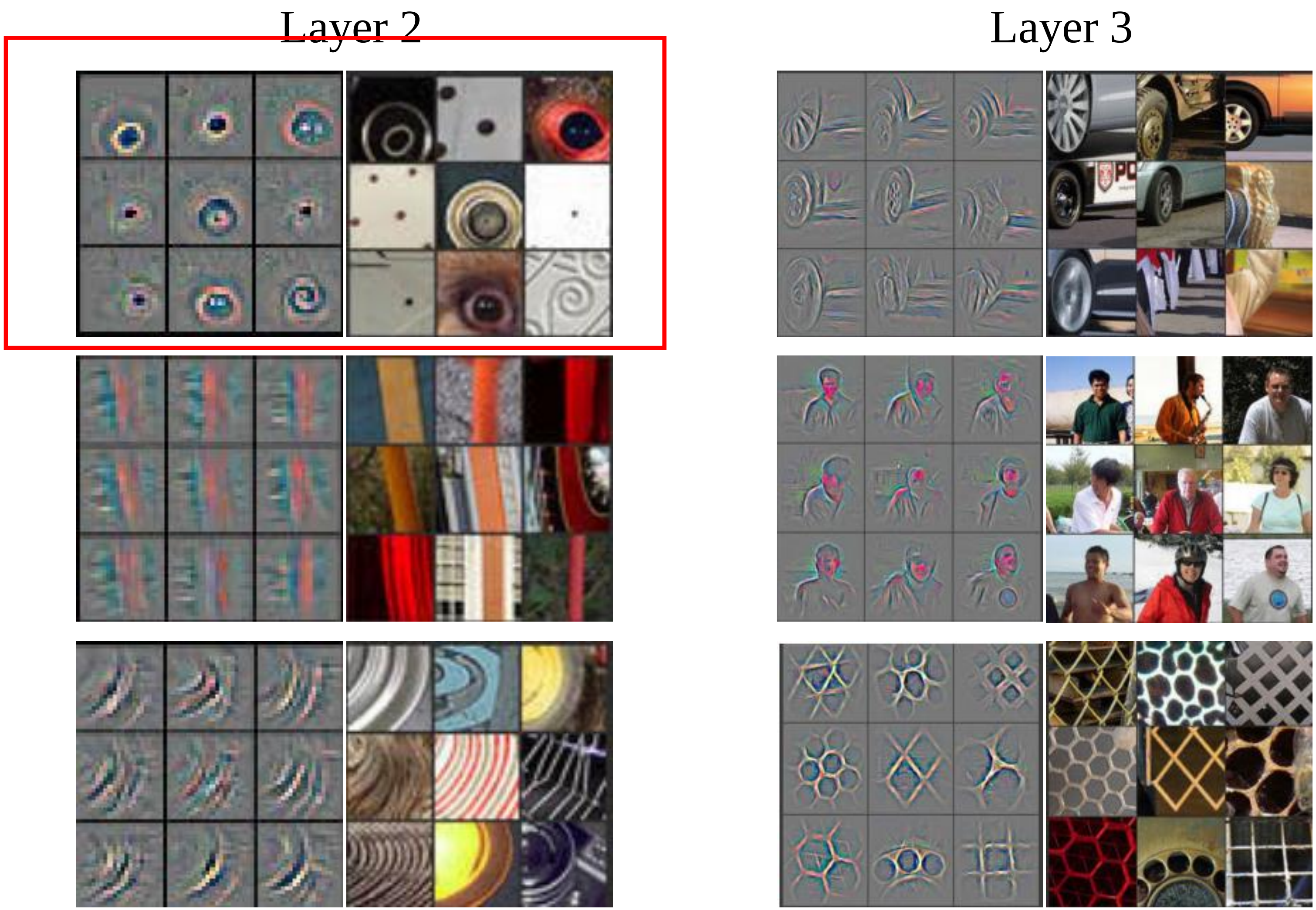
Mapping features back
to the input pixel space





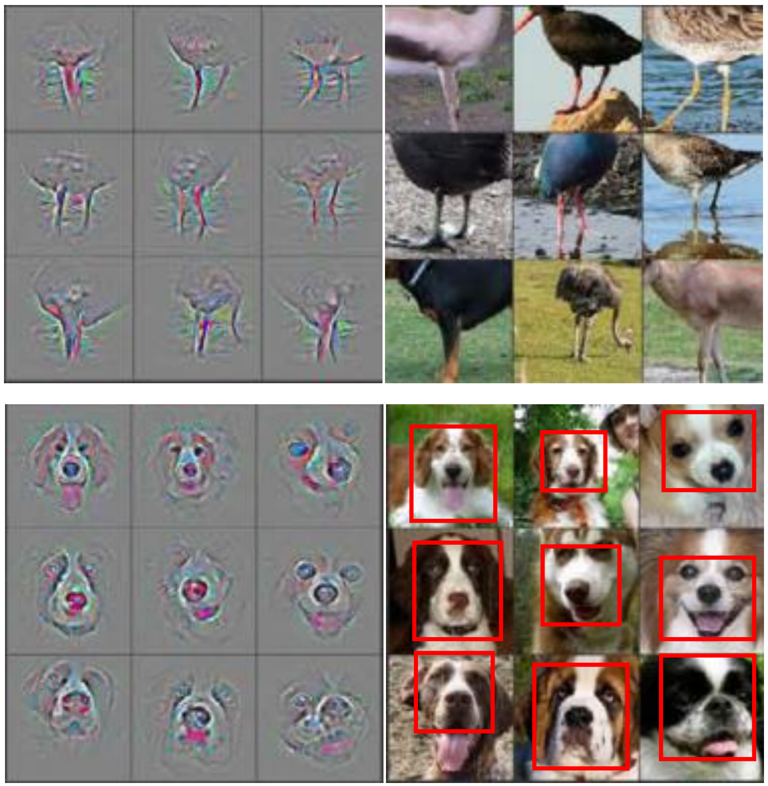
Corresponding
original images

Feature Visualization

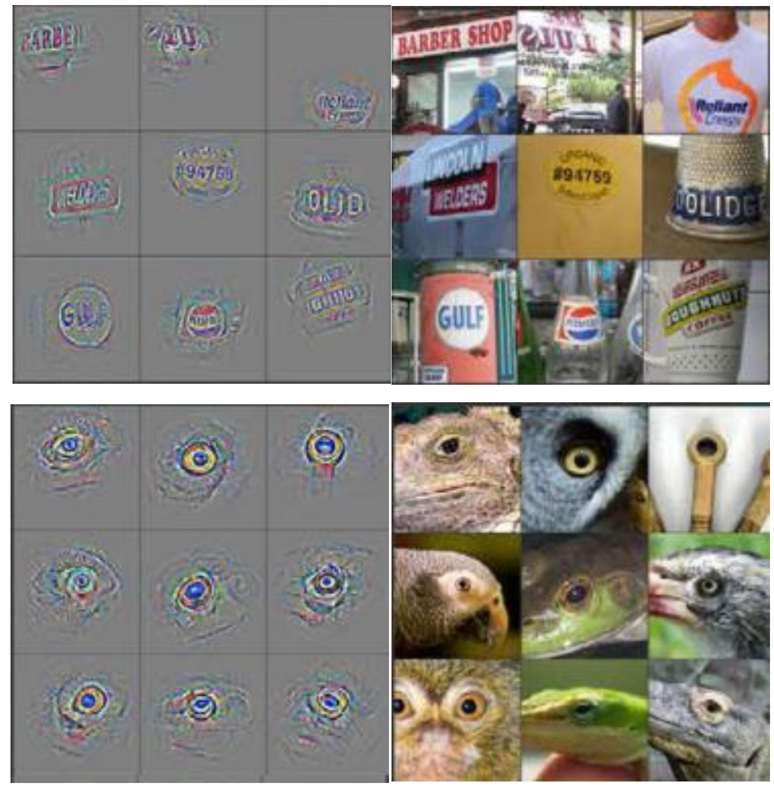


Feature Visualization

Layer 4



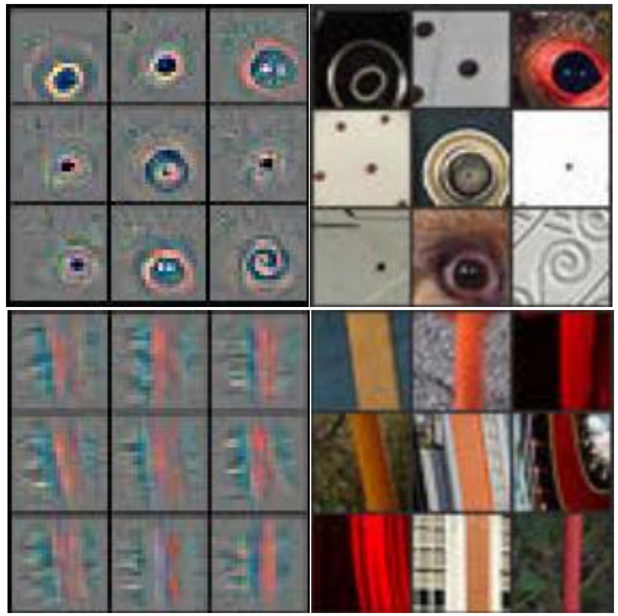
Layer 5



The strong grouping within each feature map

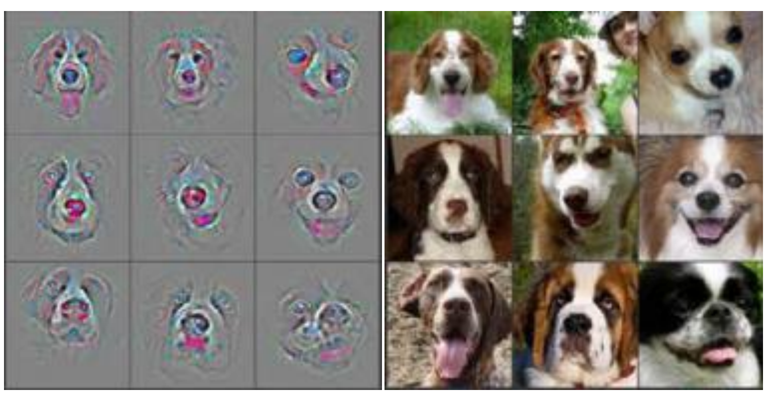
Feature Visualization

Layer 1

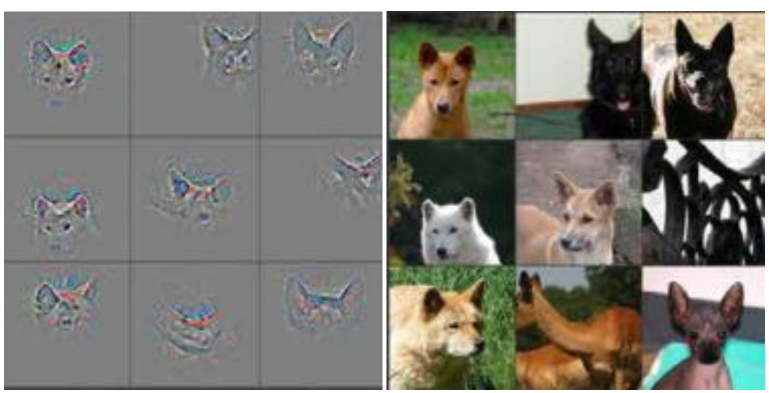


Layer 2 edge/color

Layer 3 texture

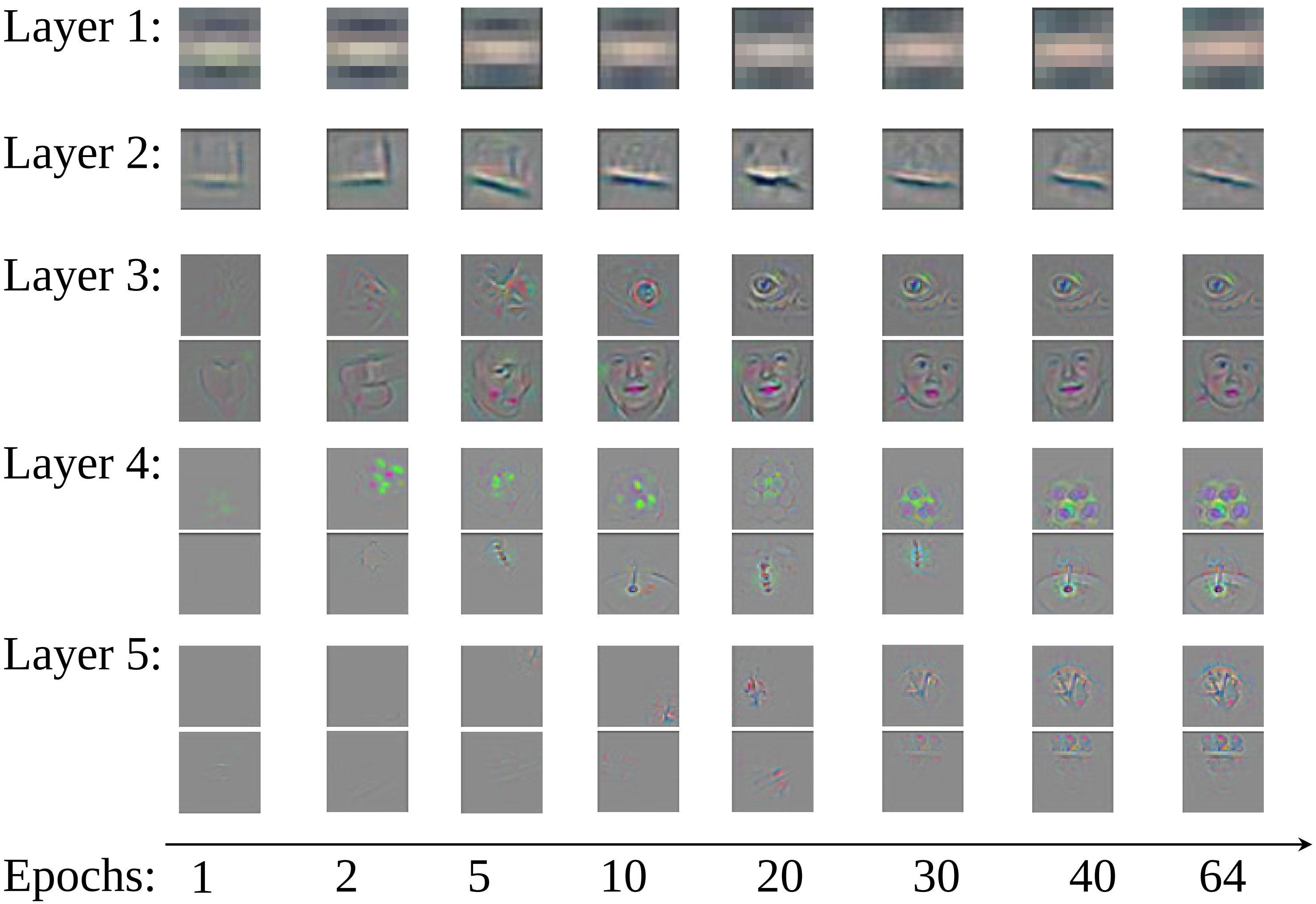


Layer 4

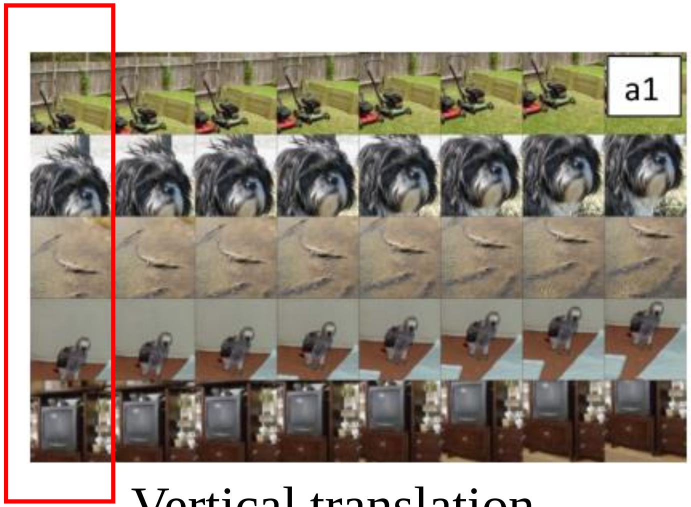


Layer 5

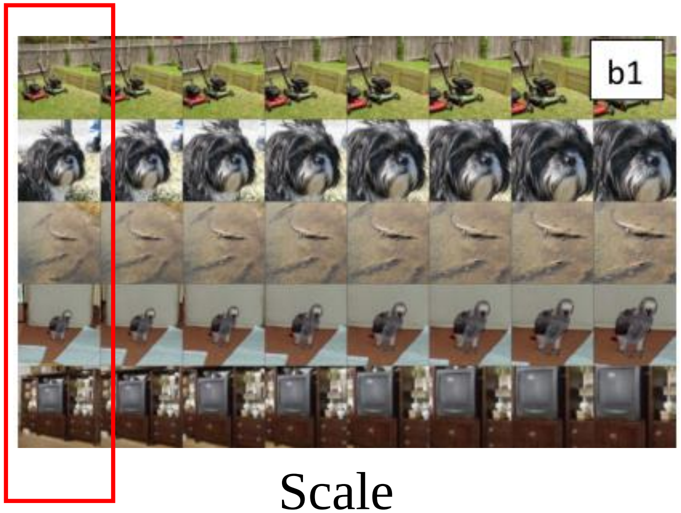
Feature Evolution



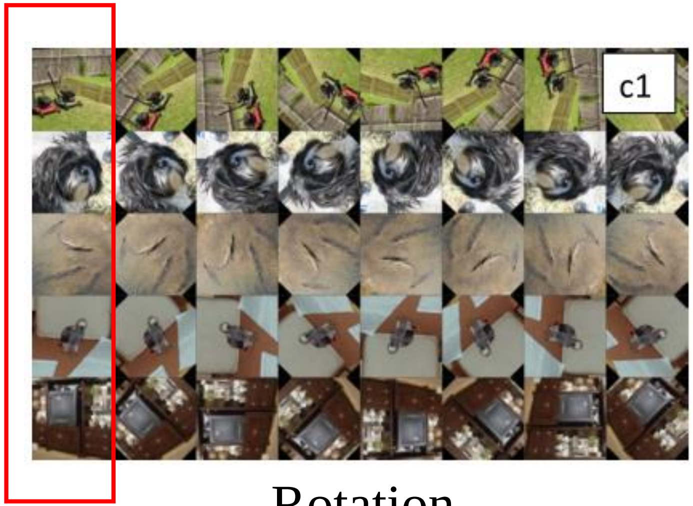
Feature Invariance



Vertical translation



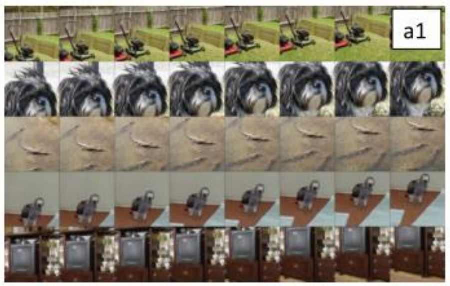
Scale



Rotation

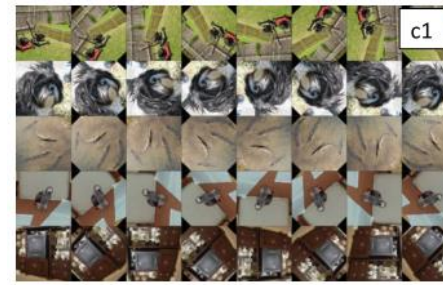
Red boxes display 5 original images

Feature Invariance

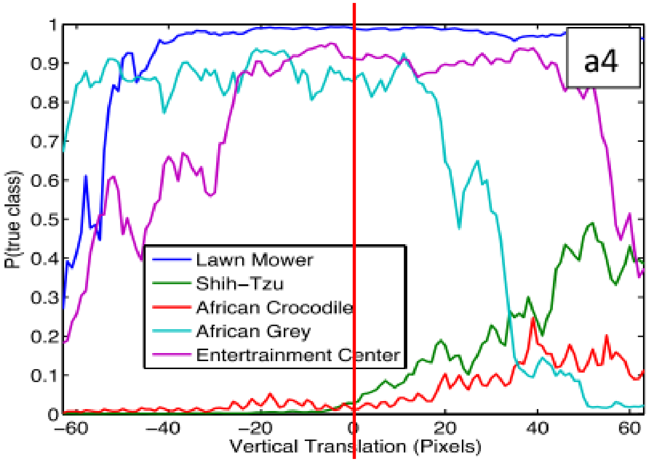


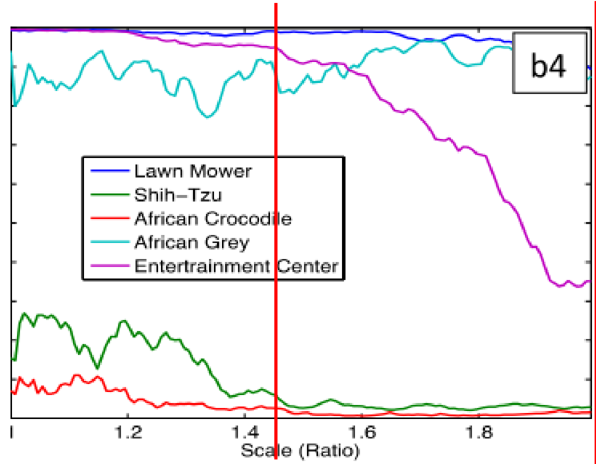
Vertical translation

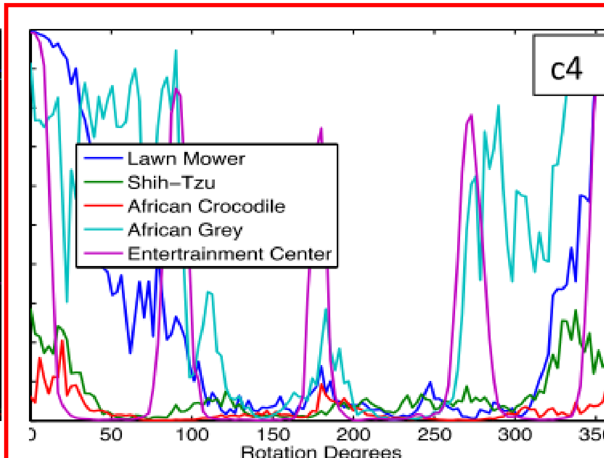
Scale



Rotation

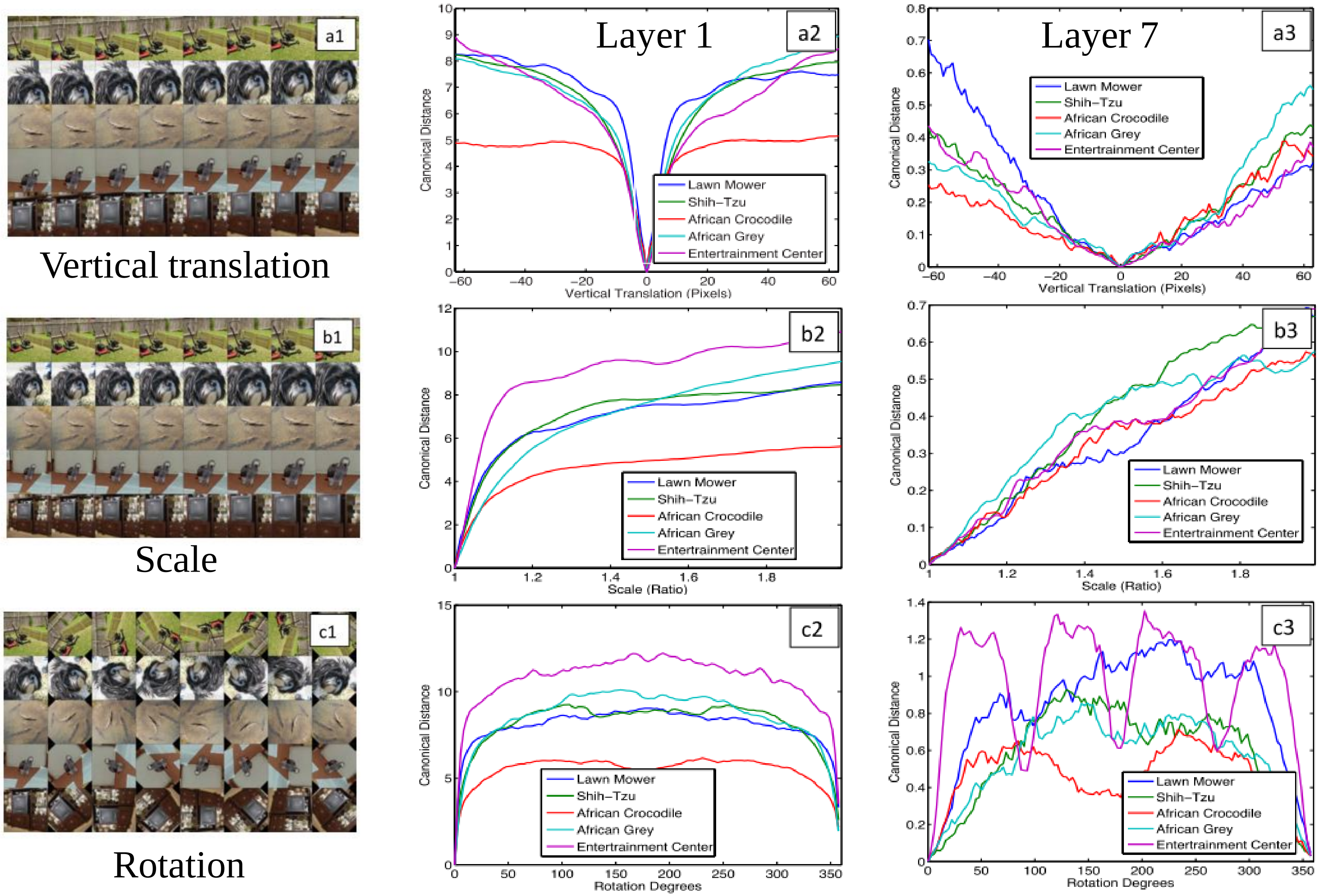






The probability of the true label for each image

Feature Invariance



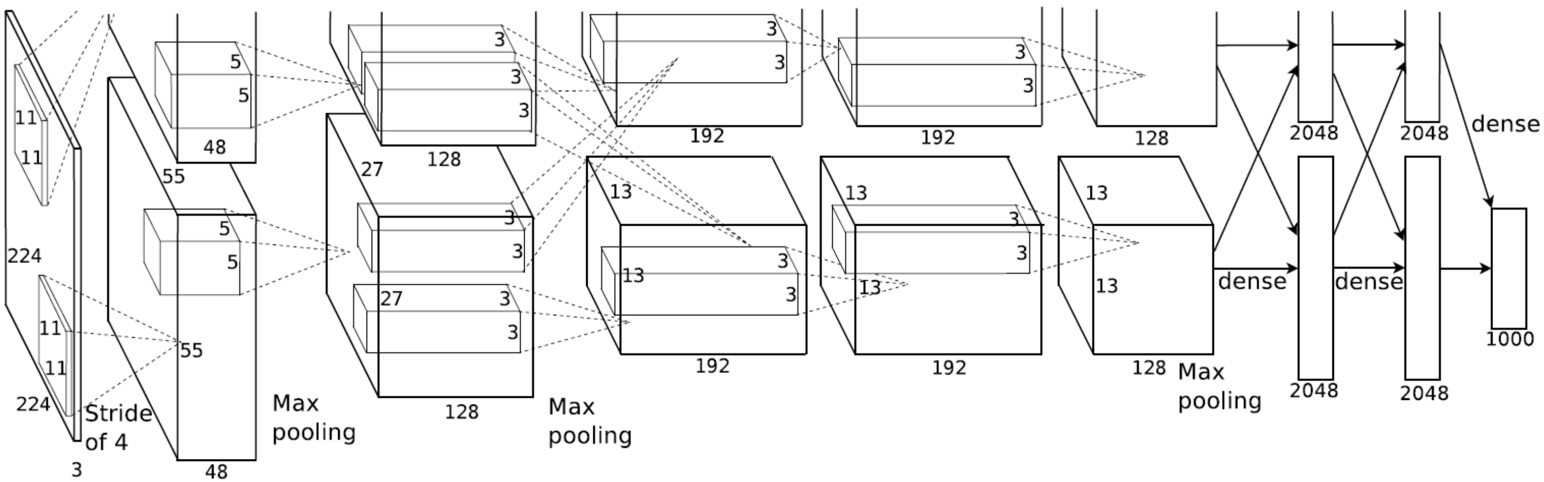
Feature Visualization

Why they perform so well?

- the availability of much larger training sets, with millions of labeled examples;
- powerful GPU implementations, making the training of very large models practical
- better model regularization strategies, such as Dropout
- deep layers with more abstract feature representations
- greater invariance at higher layers

Architecture Selection

How they might be improved?

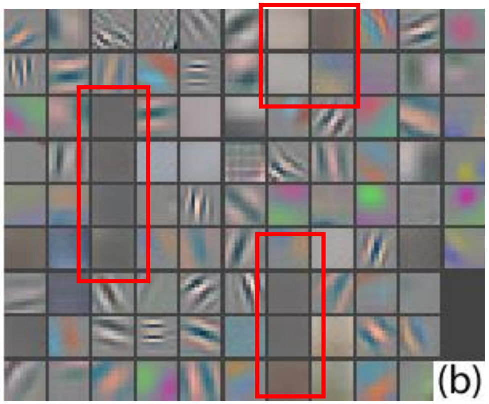


AlexNet [4]

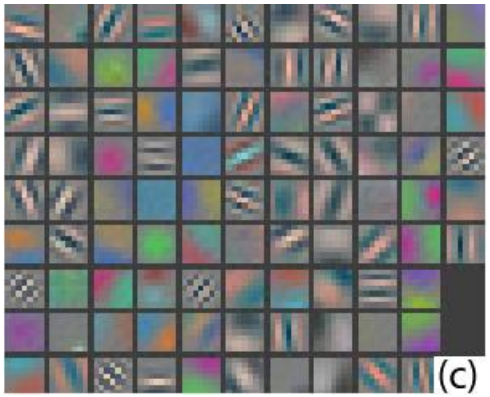
[4] Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. 2012: 1097-1105.

Architecture Selection

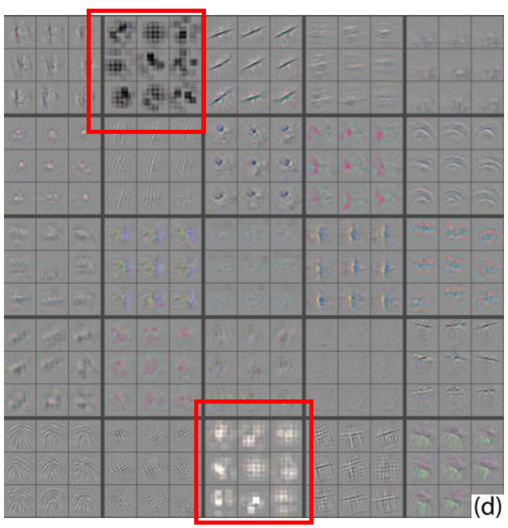
Layer 1



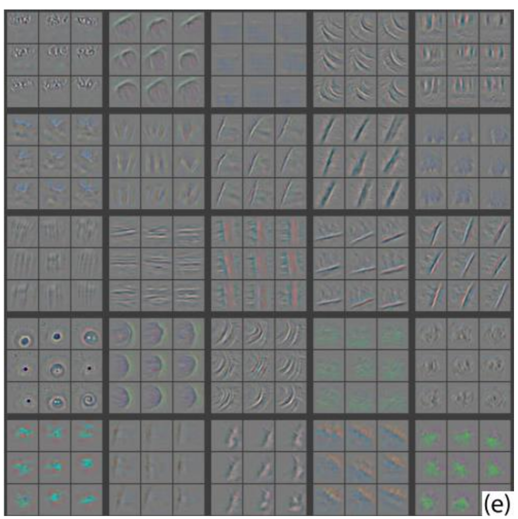
smaller filters:
11×11 to 7×7



Layer 2



smaller stride:
4 to 2

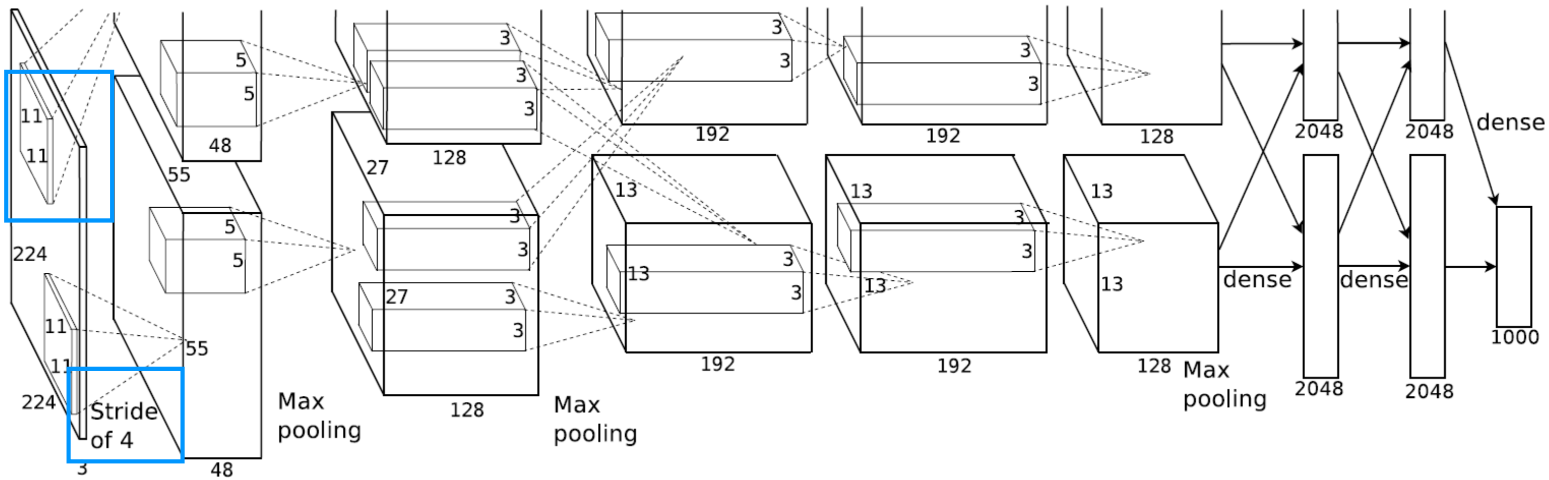


AlexNet

ZFNet

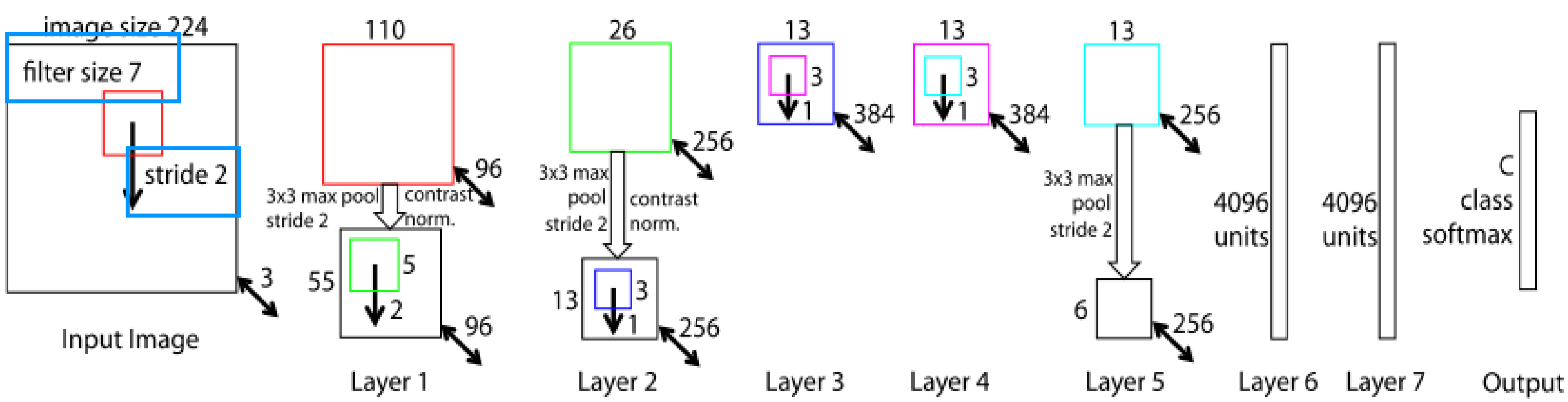
Architecture Selection

AlexNet



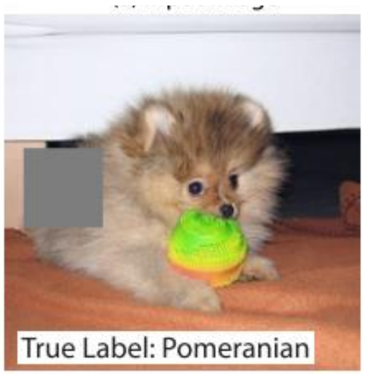
Smaller filters 11x11 to 7x7 and Smaller stride 4 to 2

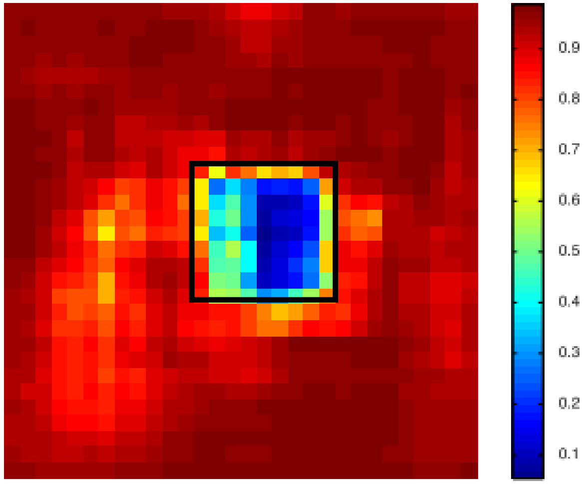
ZFNet

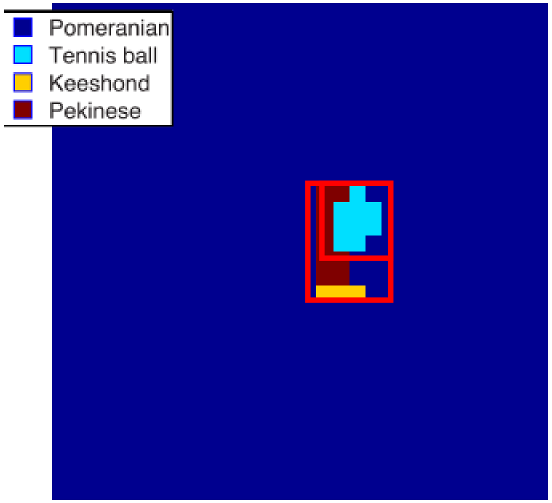


Occlusion Sensitivity

If the model is truly identifying the location of the object in the image, or just using the surrounding context.





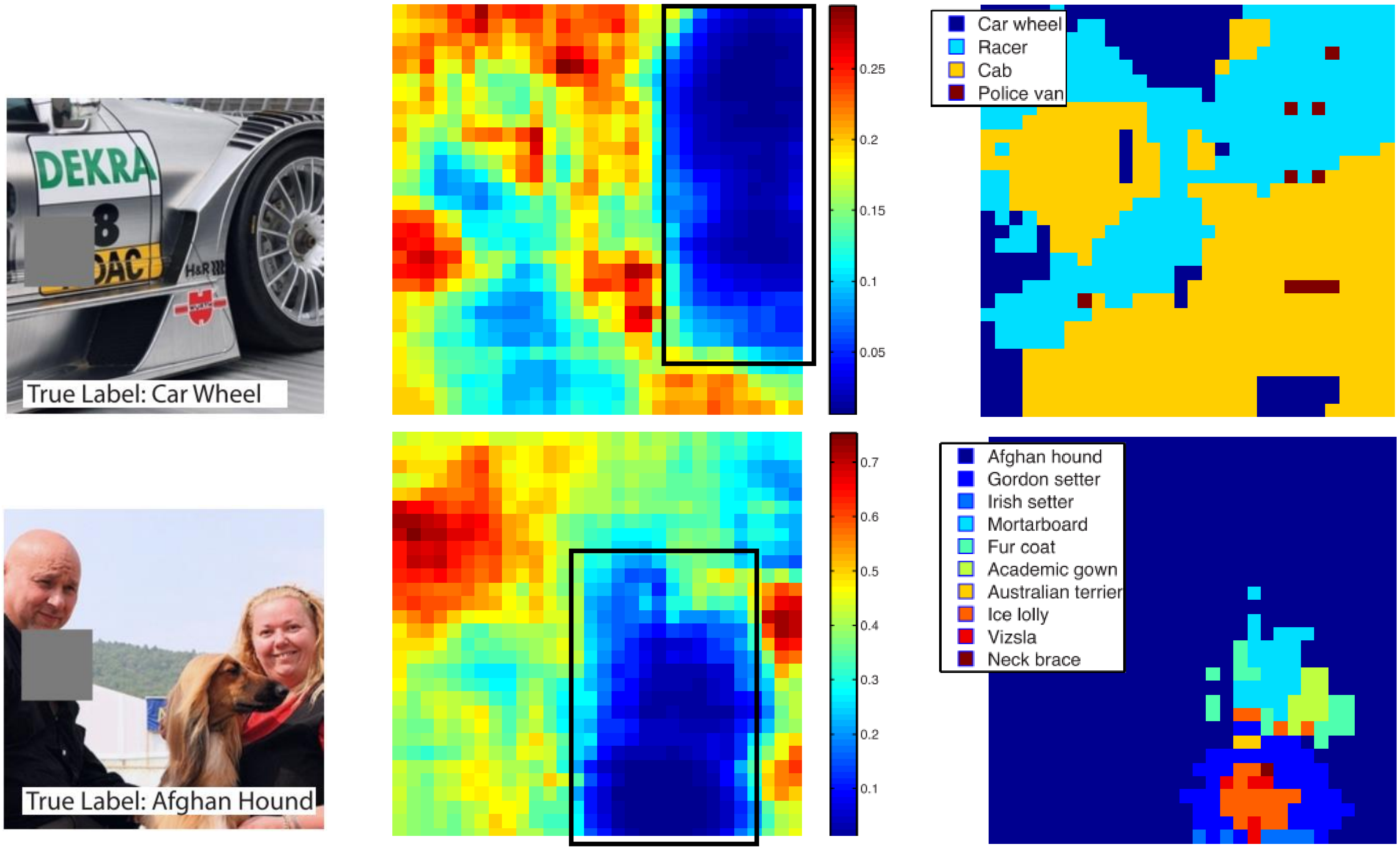


Input image

Classifier, probability of correct class

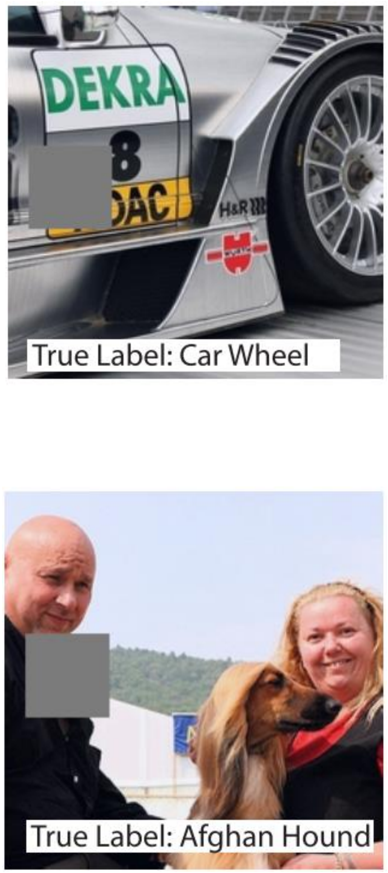
Classifier, most probable class

Occlusion Sensitivity



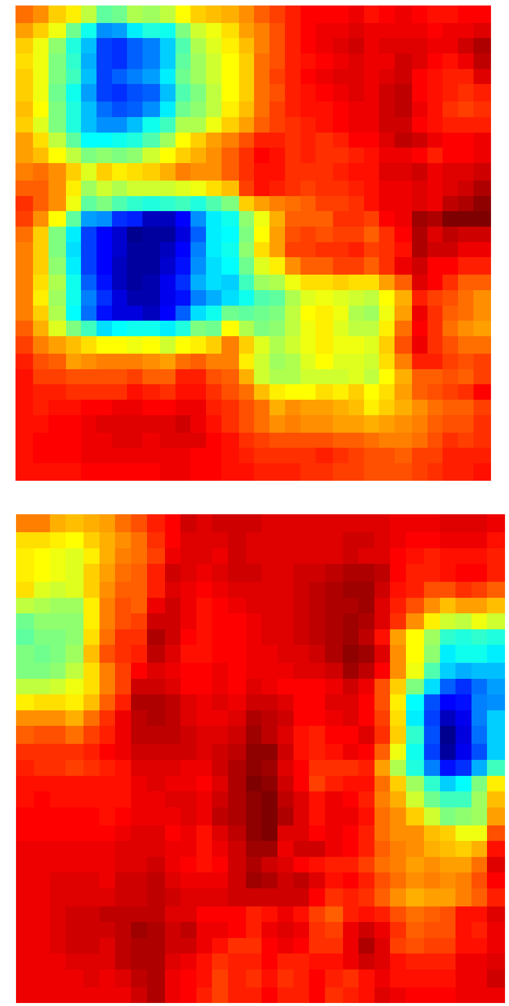
Input image　　Probability of correct class　　Most probable class

Occlusion Sensitivity



Input image

Strongest feature map projections



Strongest feature map

Outline

- Visualization Approaches
- Convnet Visualization
- Experiments
- My Experiments

Experimental Results

Error %	Train Top-1	Val Top-1	Val Top-5
Alexnt, 1 convnet	40.7	18.2	– –
Alexnt, 5 convnets	38.1	16.4	16.4
Alexnt*, 1 convnet	39.0	16.6	– –
Alexnt*, 7 convnets	36.7	15.4	15.3

Our replication of AlexNet, 1convent	40.5	18.1	– –
ZFNet, 1 convnet	38.4	16.5	– –
ZFNet, 5 convnets –(a)	36.7	15.3	15.3
ZFNet, 1 convnet but with layers 3, 4, 5: 512, 1024, 512 maps –(b)	37.5	16	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

ImageNet 2012 classification error rates (ACC %)

Experimental Results

Error %	Train Top-1	Val Top-1	Val Top-5
The replication of AlexNet	35.1	40.5	18.1
Removed layers 3,4	41.8	45.4	22.1
Removed layers 6,7	27.4	44.8	22.4
Removed layers 3,4,6,7	71.1	71.3	50.1
Adjust layers 6,7: 2048 units	40.3	41.7	18.8
Adjust layers 6,7: 8192 units	26.8	40	18.1

ZFNet	33.1	38.4	16.5
Adjust layers 6,7: 2048 units	38.2	40.2	17.6
Adjust layers 6,7: 8192 units	22	38.8	17
Adjust layers 3,4,5: 512, 1024, 512 maps	18.8	**37.5**	**16**
Adjust layers 6,7: 8192 units and layers 3,4,5: 512, 1024, 512 maps	10	38.3	16.9

ImageNet 2012 classification error rates (ACC %)

Feature Generalization

#Methods	15/class	30/class
Jianchaoetal.,2009	73.2	84.3
Non-pretrained ZFNet	22.8±1.5	46.5±1.7
ImageNet-pretrained ZFNet	**83.8±0.5**	**86.5±0.5**

Caltech-101 classification accuracies (ACC %)

#Methods	15/class	30/class	45/class	60/class
Boetal.,2013	40.5±0.4	48.0±0.2	51.9±0.2	55.2±0.3
Non-pretrained	9.0±1.4	22.5±0.7	31.2±0.5	38.8±1.4
ImageNet-pretrained	**65.7±0.2**	**70.6±0.2**	**72.7±0.4**	**74.2±0.3**

Caltech-256 classification accuracies (ACC %)

Feature Analysis

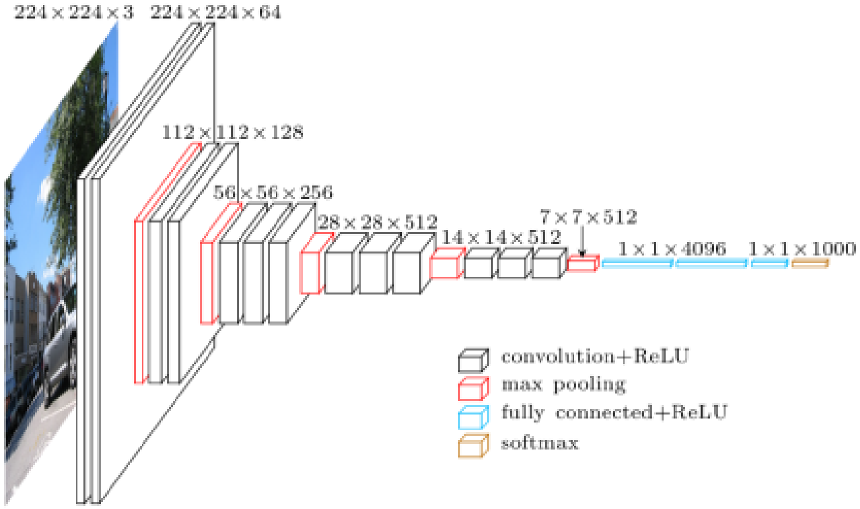
	Caltech-101 (30/class)	**Caltech-256 (60/class)**
SVM (1)	44.8±0.7	24.6±0.4
SVM (2)	66.2±0.5	39.6±0.3
SVM (3)	72.3±0.4	46.0±0.3
SVM (4)	76.6±0.4	51.3±0.1
SVM (5)	**86.2±0.8**	65.6±0.3
SVM (7)	85.5±0.4	71.7±0.2
Softmax (5)	82.9±0.4	65.7±0.5
Softmax (7)	85.4±0.4	**72.6±0.1**

Outline

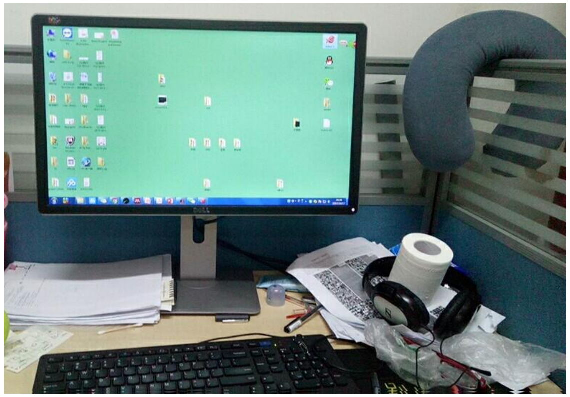
- Visualization Approaches
- Convnet Visualization
- Experiments
- My Experiments

My Experiment

- Find a trained model and weights
- Take an image



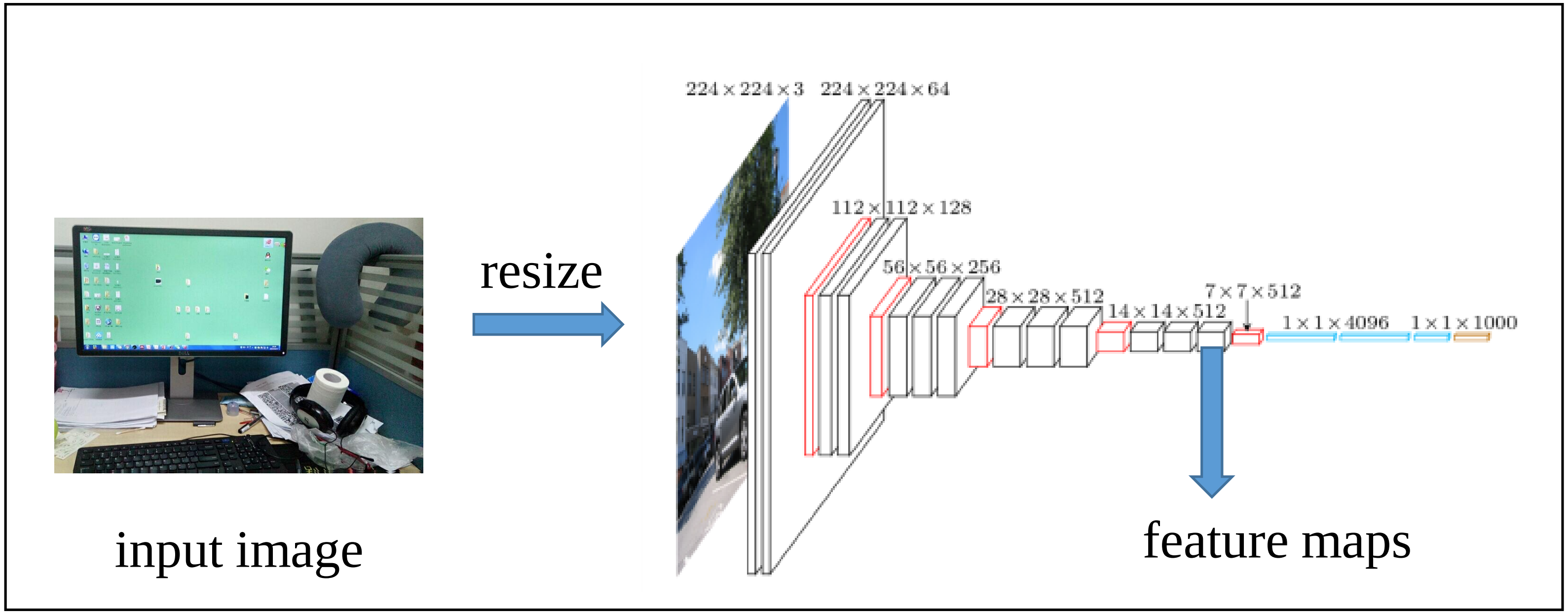
ImageNet-pretrained VGG-16



My computer

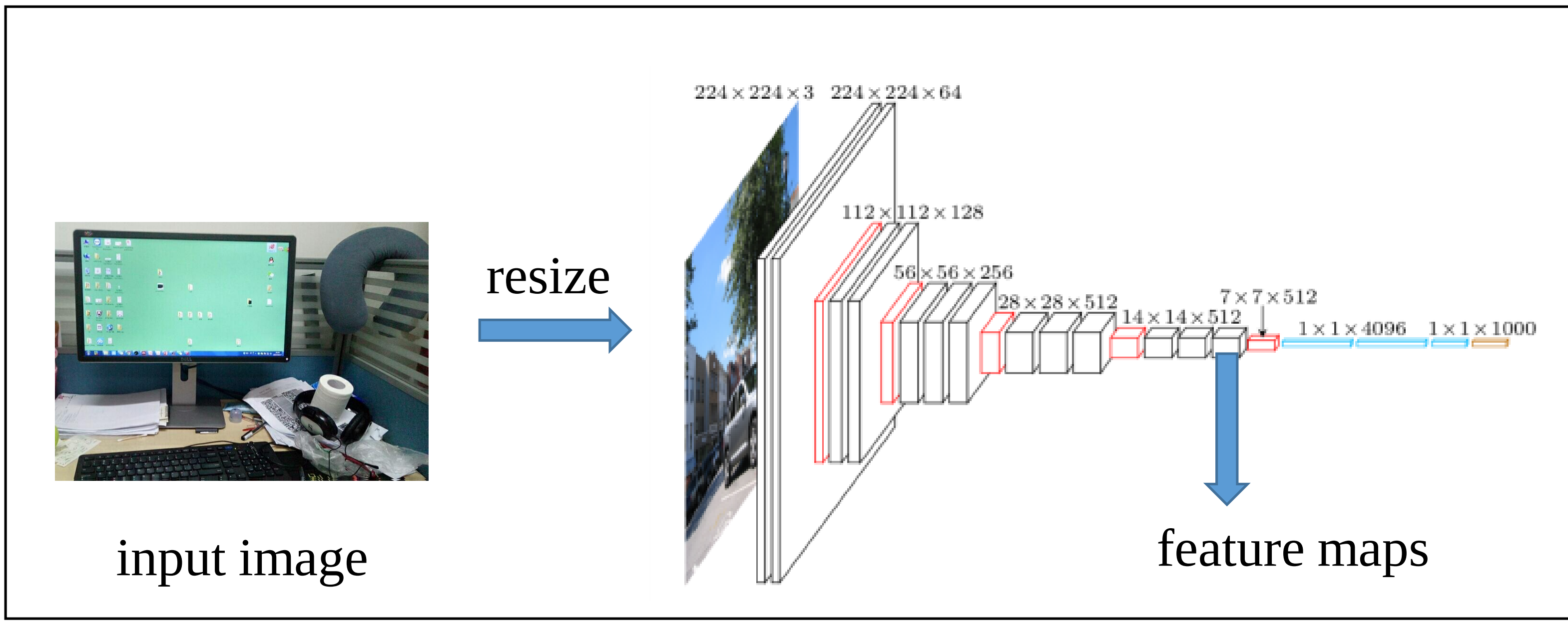
My Experiment

- The image is presented to VGG-16 and features computed throughout the layers



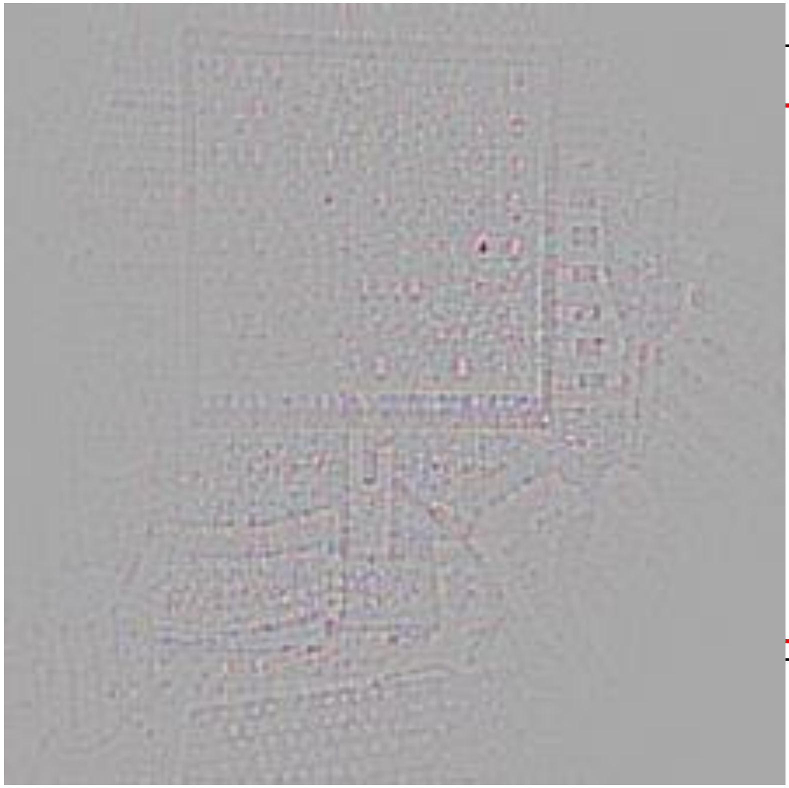
My Experiment

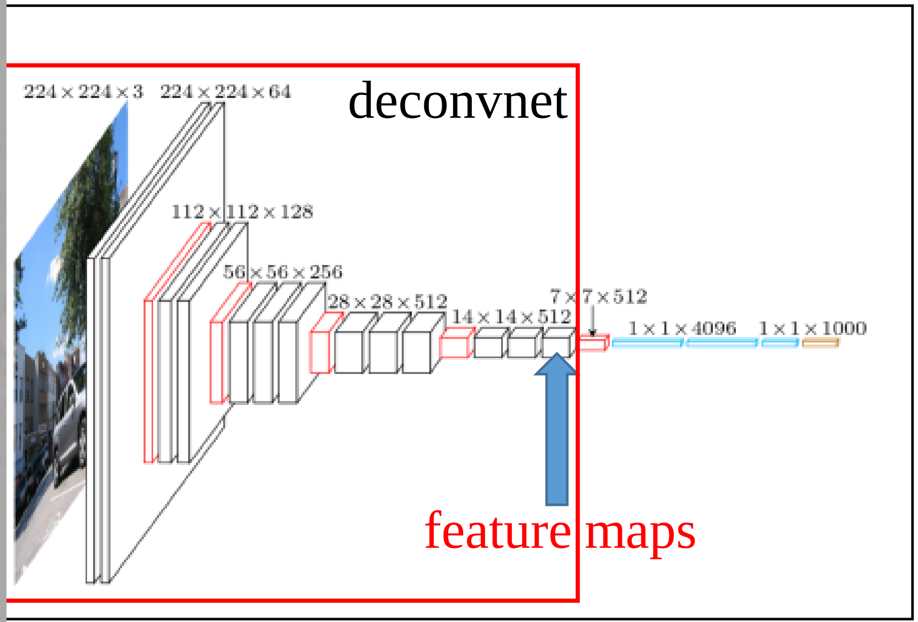
- keep concerned feature activity and set other activities to zero



My Experiment

- put features as the input to a deconvnet





Thanks!