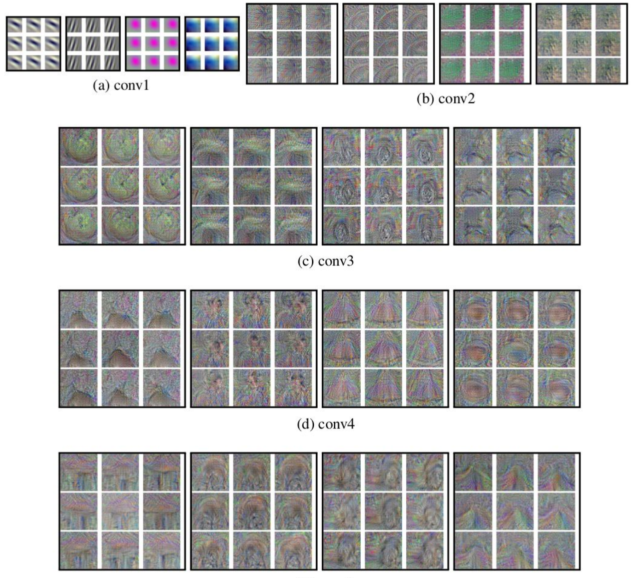
Visualizing and Understanding Convolutional Networks

Speaker: Aleksandrov Nikita

Paper authors: Matthew D. Zeiler, Rob Fergus

What will we learn from this presentation?

- We will learn how to get images like below from your convolutional network:



The image is taken from this article

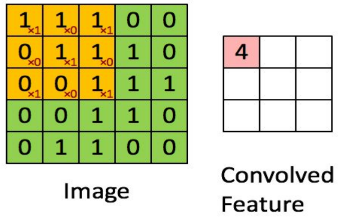
The plan of the presentation

Introduction

1. How convolutional neural networks works
2. High-level overview of the article

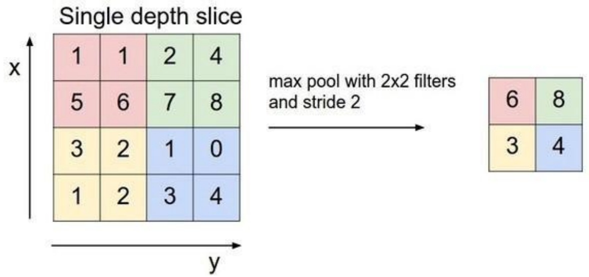
How convolutional neural networks works

Convolution mechanism



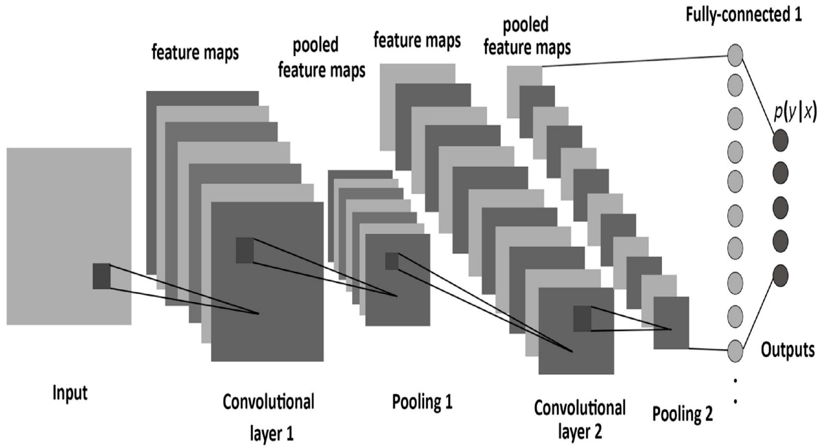
The picture is in public property

Max pooling filter mechanism



The image is taken from this blogpost

CNN architecture high-level overview



The image is taken from this article

High-level overview of the article

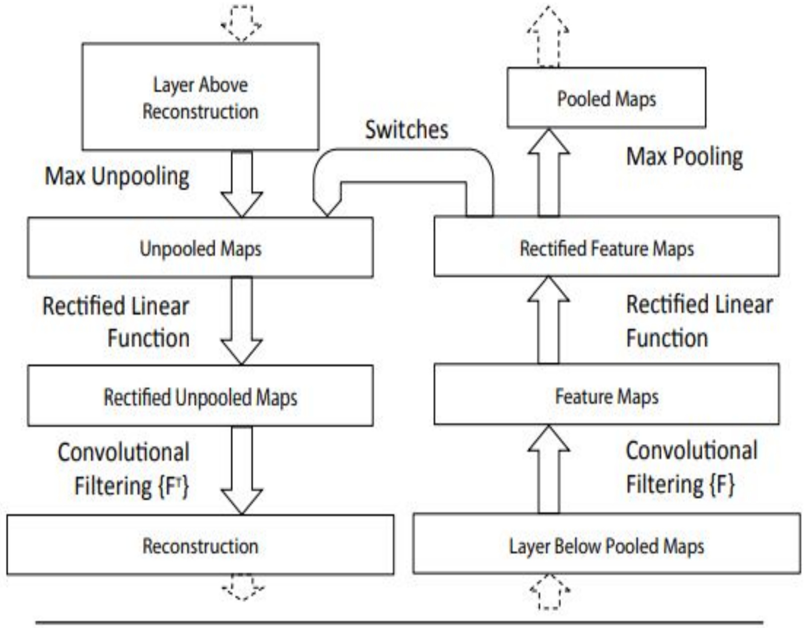
1. CNNs show great results. However there is no clear understanding of why they perform so well, or how they might be improved.
2. Authors introduce a novel visualization technique that gives insight into the function of intermediate feature layers. They do visualization with deconvolutional neural network.
3. Used in a diagnostic role, these visualizations allowed authors to find model architecture that outperform state of the art solution on ImageNet dataset.
4. Their ImageNet model generalizes well to other datasets: when the softmax classifier is retrained, it convincingly beats state-of-the-art results on Caltech-101 and Caltech-256 datasets.

Deconvolutional neural networks

1. High-level overview of deconvolutional network
2. Unpooling
3. Rectification
4. Filtering

High-level overview of deconvolutional network

- To examine a convnet, a deconvnet is attached to each of its layers, as illustrated on figure on the left providing a continuous path back to image pixels.



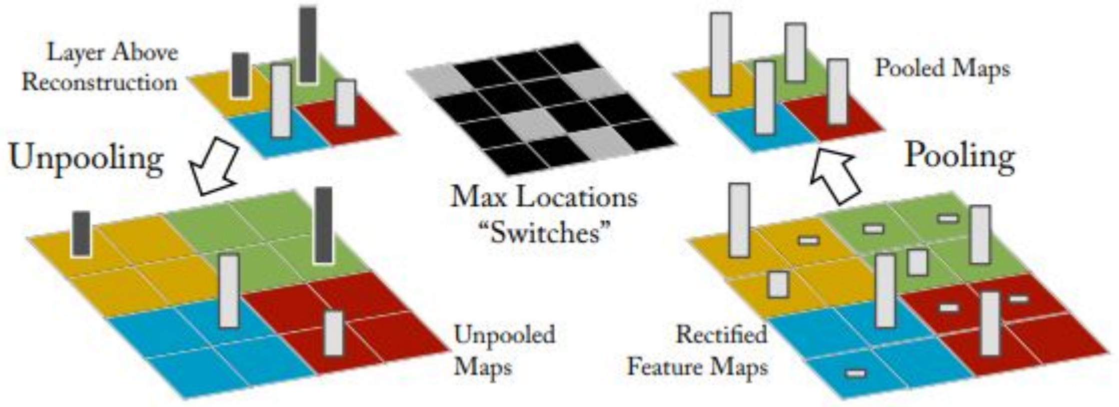
Part of Figure 1 in presenting paper

High-level overview of deconvolutional network

1. To examine a given convnet feature map, we set all other activations in the layer to zero and pass the feature maps as input to the attached deconvnet layer.
2. Then we successively rectify, filter and unpool to reconstruct the activity in the layer beneath that gave rise to the chosen activation.
3. This is then repeated until input pixel space is reached.

Unpooling

In the convnet, the max pooling operation is non-invertible, however we can obtain an approximate inverse by recording the locations of the maxima within each pooling region in a set of switch variables.



Part of Figure 1 in presenting article

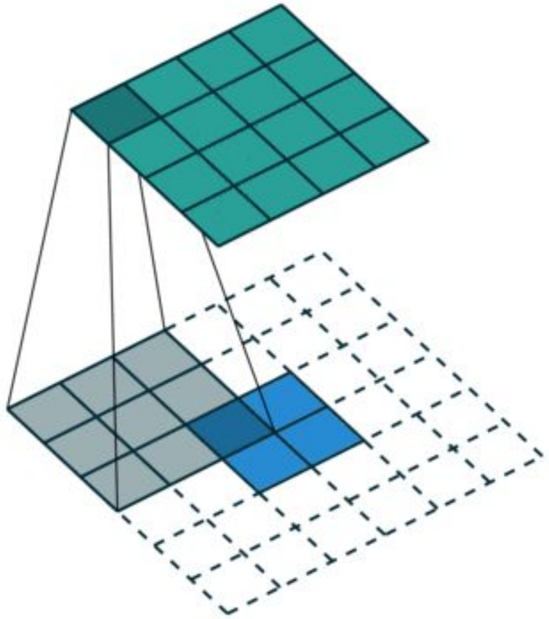
Rectification

- The convnet uses relu non-linearities, which rectify the feature maps thus ensuring the feature maps are always positive.
- We will use ReLU too in order to obtain valid feature reconstructions at each layer.

Filtering

1. We use the same filters which was learnt by convolutional network
2. To invert those filters, we transpose them and apply to rectified feature map (reconstructed feature map from the above), not the output of the layer beneath.

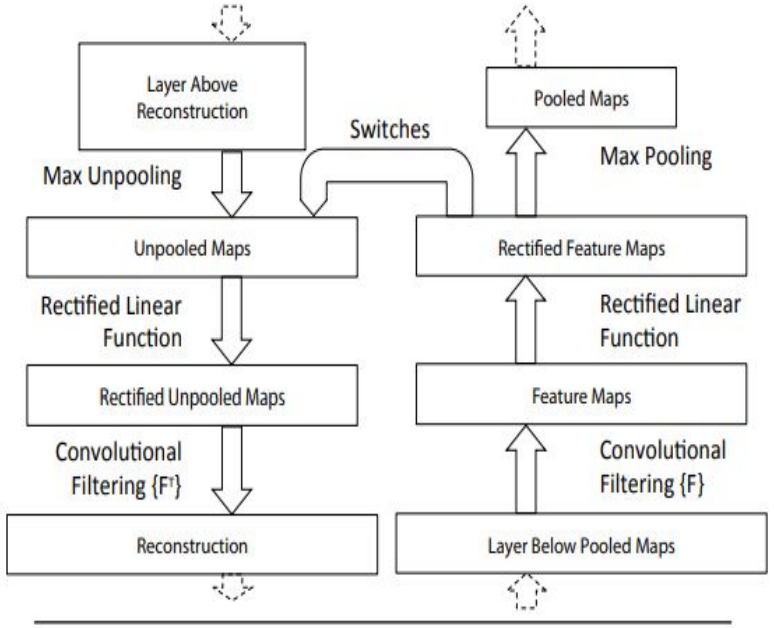
Visualization of simplest filtering



The image is taken from this repository

To sum up

- Projecting down from higher layers uses the switch settings generated by the max pooling in the convnet on the way up.
- As these switch settings are peculiar to a given input image, the reconstruction obtained from a single activation thus resembles a small piece of the original input image, with structures weighted according to their contribution toward to the feature activation.



Part of Figure 1 in presenting article

Author’s model architecture

- Here we have 8 layers.
- Cropped pictures 224x224 are used as input.
- On the first layer we have 96 convolutions, every have 7x7 size and step 2. On layers 2, 3, 4, 5 authors do similar operation.
- Last two layers are fully connected layers with softmax classifier on the output.

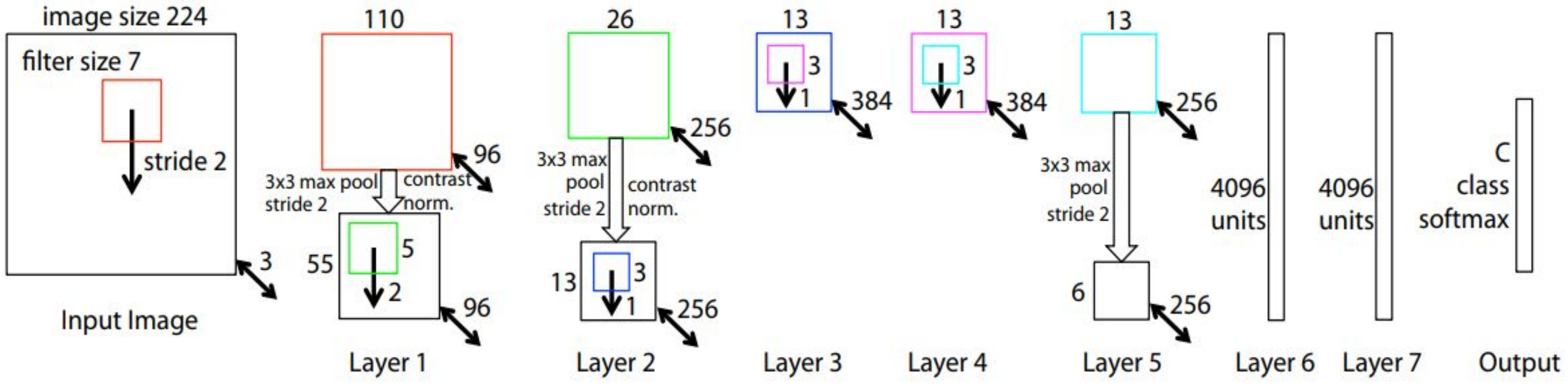


Figure 3 in presenting article

Convnet visualization

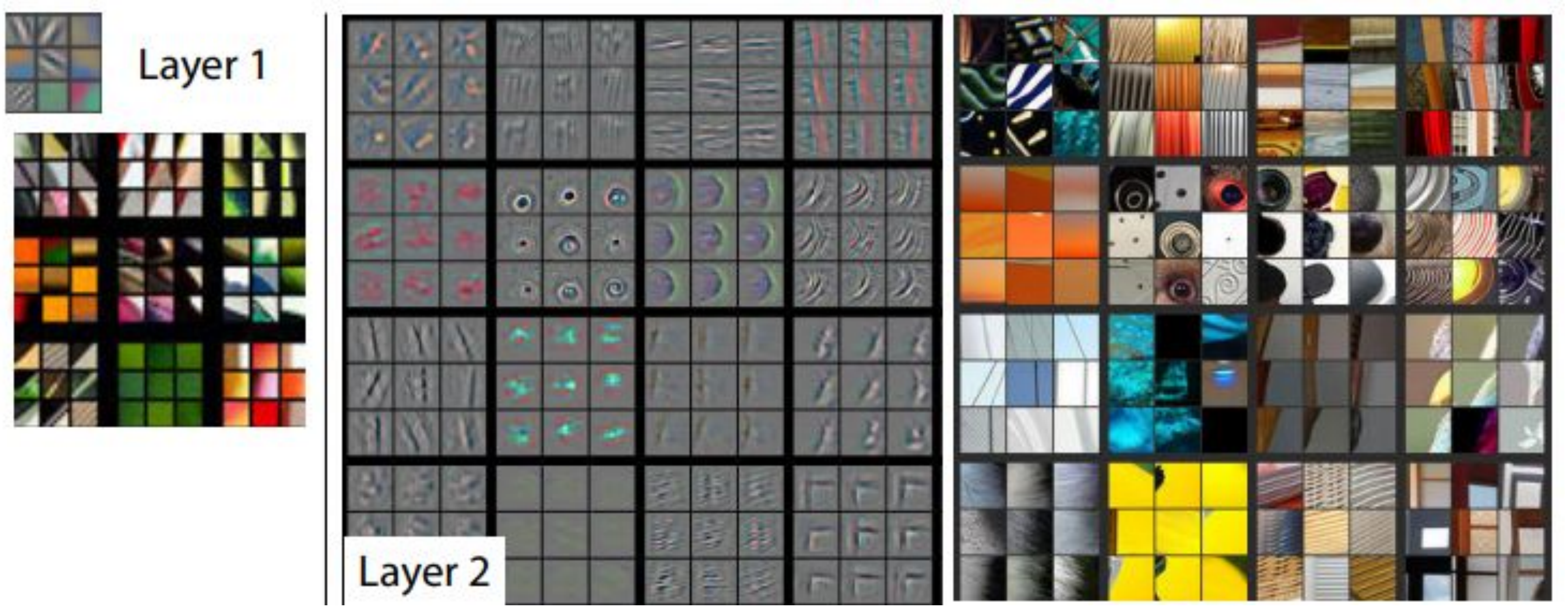
1. Feature visualization
2. Feature Evolution during Training
3. Feature Invariance (optional)
4. Architecture Selection (optional)
5. Correspondence Analysis (optional)

Feature visualization

On the next slides, we will see a visualization of some feature maps of the CNN presented on slide 16.

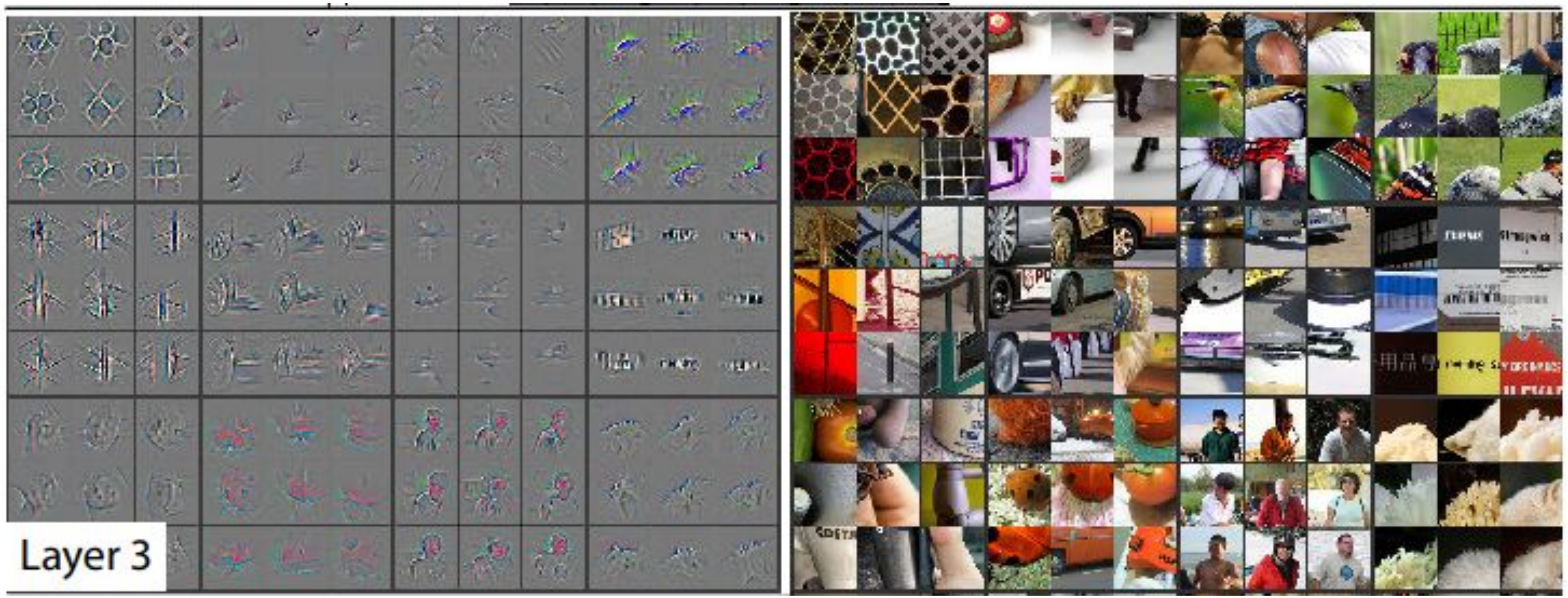
1. For each layer authors took a random subset of feature maps
2. For each feature map, they projected down to the pixel space top 9 activations

Feature visualization layers 1 and 2



Part of Figure 2 in presenting article

Feature visualization, layer 3



Part of Figure 2 in presenting article

Feature visualization, layers 4 and 5

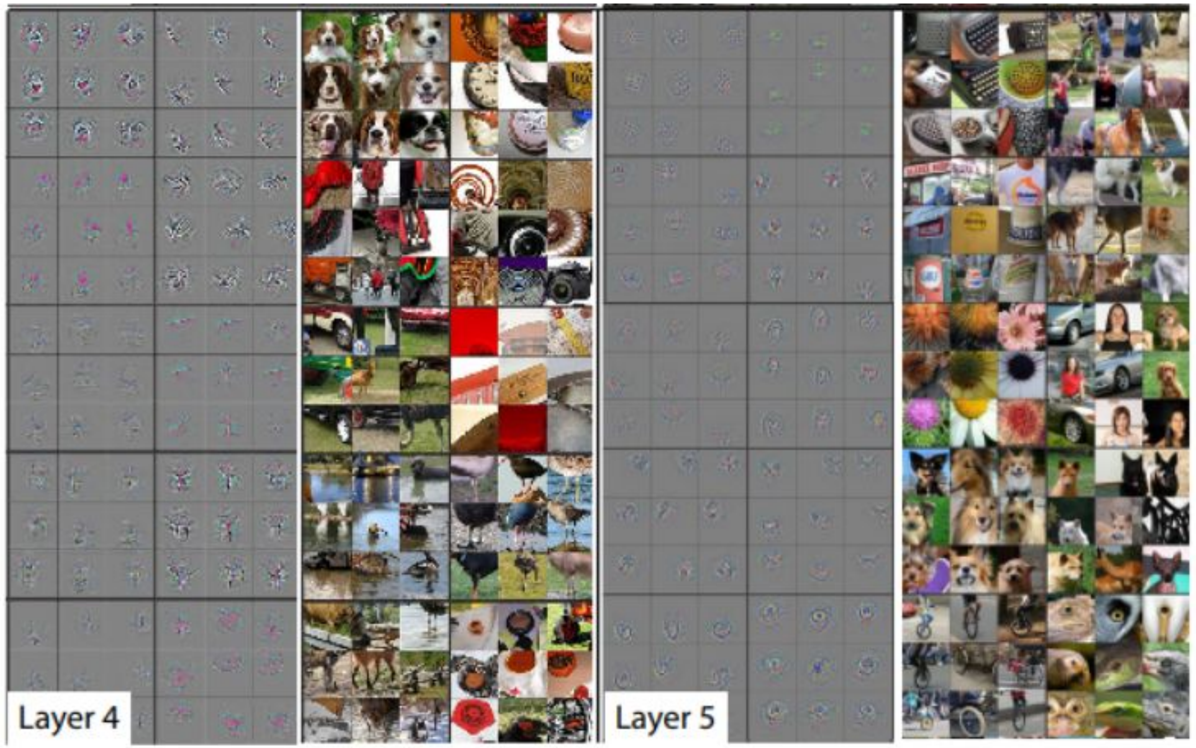


Figure 4 in presenting article

Feature visualization, sum up

- The projections from each layer show the hierarchical nature of the features in the network.
- Deeper layers capture more complex patterns.

Feature Evolution during Training

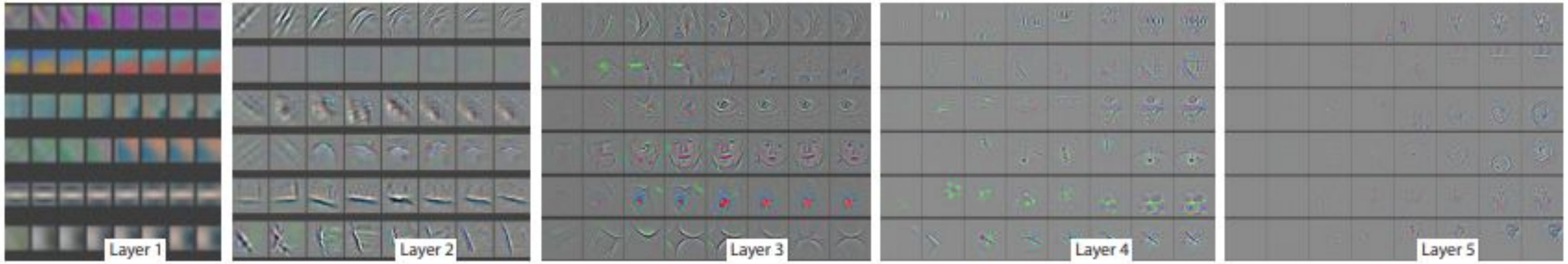


Figure 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to pixel space using our deconvnet approach. Color contrast is artificially enhanced and the figure is best viewed in electronic form.

Figure 4 in presenting article

Feature Invariance

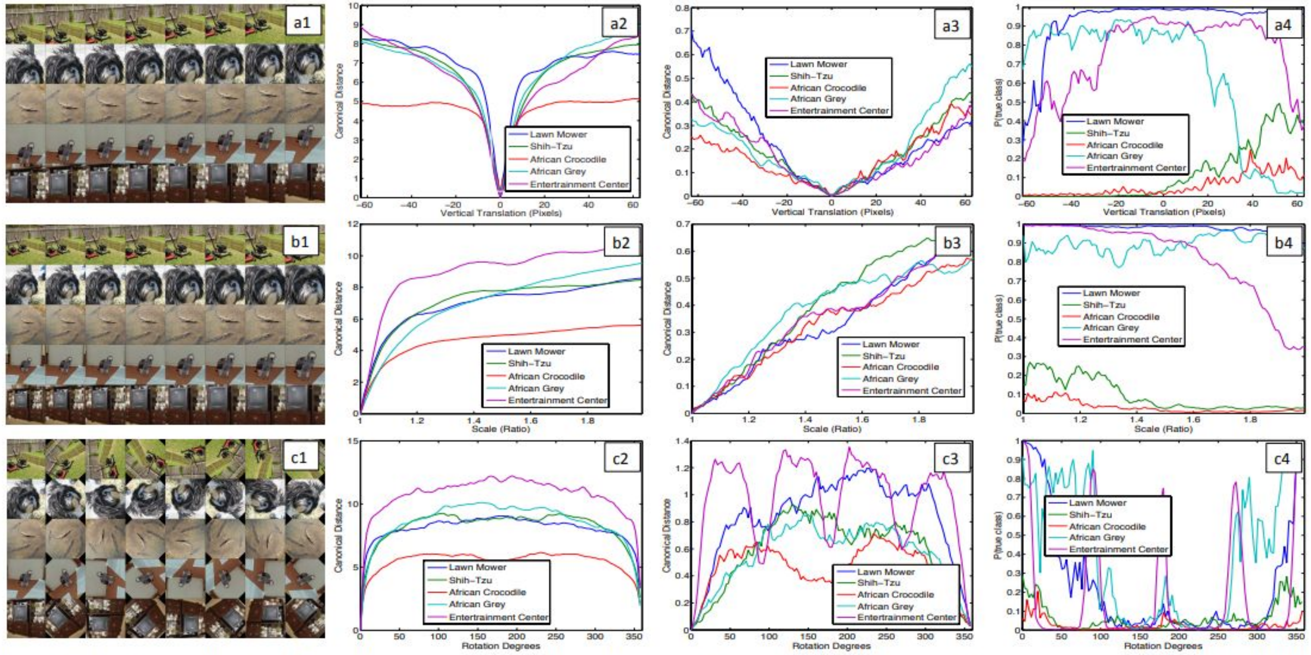


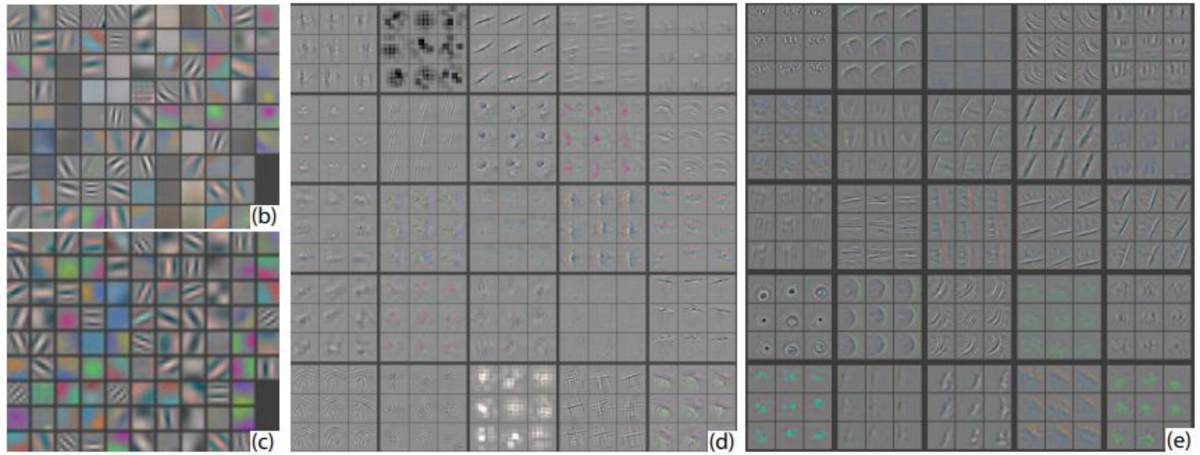
Figure 5. Analysis of vertical translation, scale, and rotation invariance within the model (rows a-c respectively). Col 1: 5 example images undergoing the transformations. Col 2 & 3: Euclidean distance between feature vectors from the original and transformed images in layers 1 and 7 respectively. Col 4: the probability of the true label for each image, as the image is transformed.

Architecture Selection

1. While visualization of a trained model gives insight into its operation, it can also assist with selecting good architectures in the first place.
2. The first layer filters are a mix of extremely high and low frequency information, with little coverage of the mid frequencies
3. The 2nd layer visualization shows aliasing artifacts caused by the large stride 4 used in the 1st layer convolutions.
4. To remedy these problems, we (i) reduced the 1st layer filter size from 11x11 to 7x7 and (ii) made the stride of the convolution 2, rather than 4.

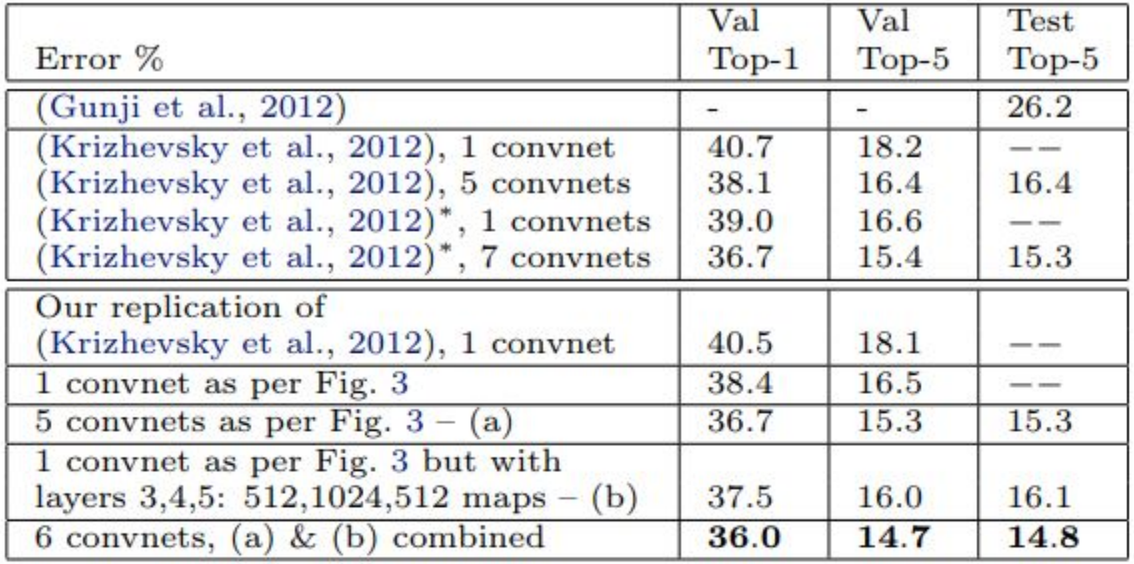
Architecture Selection

- (b) and (d) are the first version of the network, (c) and (e) are upgraded.
- This new architecture retains much more information in the 1st and 2nd layer features, as shown in the picture. More importantly, it also improves the classification performance.



Experiments and results

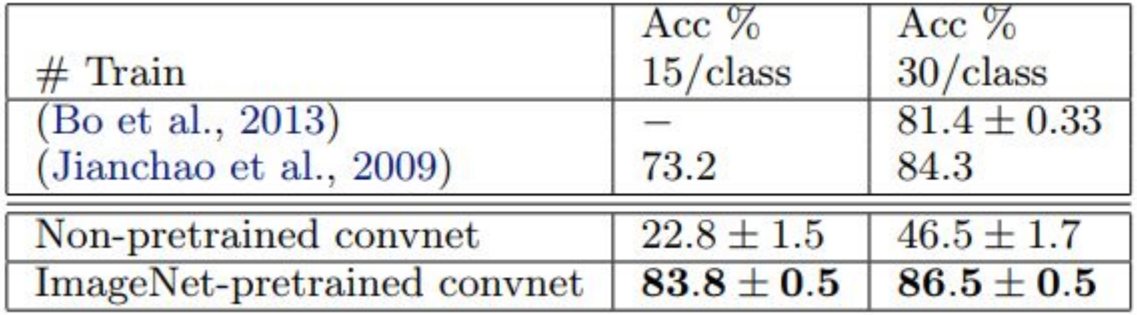
ImageNet results



Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	−−
(Krizhevsky et al., 2012), 5 convnets	38.1	16.4	16.4
(Krizhevsky et al., 2012)*, 1 convnets	39.0	16.6	−−
(Krizhevsky et al., 2012)*, 7 convnets	36.7	15.4	15.3
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	−−
1 convnet as per Fig. 3	38.4	16.5	−−
5 convnets as per Fig. 3 – (a)	36.7	15.3	15.3
1 convnet as per Fig. 3 but with layers 3,4,5: 512,1024,512 maps – (b)	37.5	16.0	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

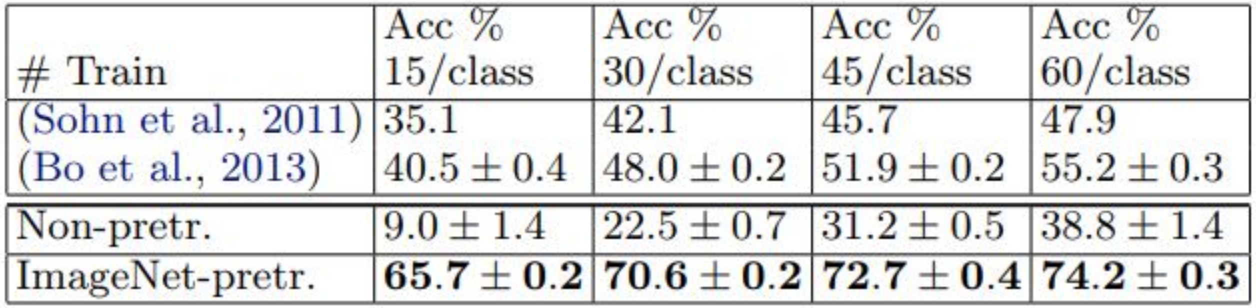
Table 2. ImageNet 2012 classification error rates. The $*$ indicates models that were trained on both ImageNet 2011 and 2012 training sets.

Caltech-101



# Train	Acc % 15/class	Acc % 30/class
(Bo et al., 2013)	–	81.4 ± 0.33
(Jianchao et al., 2009)	73.2	84.3
Non-pretrained convnet	22.8 ± 1.5	46.5 ± 1.7
ImageNet-pretrained convnet	**83.8 ± 0.5**	**86.5 ± 0.5**

Caltech-256



# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
(Sohn et al., 2011)	35.1	42.1	45.7	47.9
(Bo et al., 2013)	40.5 ± 0.4	48.0 ± 0.2	51.9 ± 0.2	55.2 ± 0.3
Non-pretr.	9.0 ± 1.4	22.5 ± 0.7	31.2 ± 0.5	38.8 ± 1.4
ImageNet-pretr.	$\mathbf{65.7 \pm 0.2}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{72.7 \pm 0.4}$	$\mathbf{74.2 \pm 0.3}$

Links

- Matthew D. Zeiler, Rob Fergus *Visualizing and Understanding Convolutional Networks*
- **My notes:** https://drive.google.com/open?id=1CWL3Ui9YBlaNYh72jFx_NTOU6ALiOe8w
- https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d
- https://datascience.stackexchange.com/questions/6107/what-are-deconvolutional-layers
- https://www.quora.com/How-does-a-deconvolutional-neural-network-work#
- 3.3 http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf
- https://stats.stackexchange.com/questions/174295/what-does-the-term-saturating-nonlinearities-mean
- https://prateekvjoshi.com/2016/04/05/what-is-local-response-normalization-in-convolutional-neural-networks/